PENGEMBANGAN BASIS DATA MULTILINGUAL MENGGUNAKAN *HEADLESS* CMS DIRECTUS PADA *WEBSITE* RADYAKARTIYASA

Disusun Oleh:

N a m a : Tegar Bagus Hartadi

NIM : 21523102

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN BASIS DATA MULTILINGUAL MENGGUNAKAN *HEADLESS* CMS DIRECTUS PADA *WEBSITE* RADYAKARTIYASA

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

N a m a	: Tegar Bagus Hartadi
NIM	: 21523102

Yogyakarta, 22 Juli 2025

Pembimbing,

Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs.

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN BASIS DATA MULTILINGUAL MENGGUNAKAN *HEADLESS* CMS DIRECTUS PADA *WEBSITE* RADYAKARTIYASA

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, 30 Juli 2025

Tim Penguji

Nur Wijayaning Rahayu, S.Kom., M.Cs. _____

Anggota 1

Novi Setiani, S.T., M.T. _____

Anggota 2

Hari Setiaji, S.Kom., M.Eng. _____

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	:	Tegar Bagus Hartadi
NIM	:	21523102

Tugas akhir dengan judul:

PENGEMBANGAN BASIS DATA MULTILINGUAL MENGGUNAKAN *HEADLESS* CMS DIRECTUS PADA *WEBSITE* RADYAKARTIYASA

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, 22 Juli 2025



Tegar Bagus Hartadi

HALAMAN PERSEMBAHAN

Karya sederhana ini saya persembahkan untuk kedua orang tua saya tercinta, yang tidak hanya memberikan saya kehidupan, namun juga memberikan keyakinan kepada saya bahwa ilmu dan kerja keras adalah jalan terbaik dalam menjalani kehidupan.

HALAMAN MOTO

Kemampuan bisa dilatih, tapi semangat untuk terus tumbuh merupakan pilihan. Hari ini saya memilih untuk tumbuh.

(Tegar Bagus Hartadi)

Jika tidak hari ini mungkin minggu depan, jika tidak minggu ini mungkin bulan depan, jika tidak bulan ini mungkin tahun depan. Segala harapan kan datang yang kita impikan.

(Kita Usahakan Lagi - Batas Senja)

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat limpahan rahmat, karunia, dan hidayah-Nya, laporan tugas akhir yang berjudul "Pengembangan Basis Data Multilingual Menggunakan *Headless* CMS Directus pada *Website* Radyakartiyasa" ini dapat diselesaikan dengan baik. Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulis menyadari bahwa penyusunan laporan tugas akhir ini tidak dapat terlaksana dengan baik tanpa adanya dukungan, arahan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis diberikan kekuatan, kemampuan, dan semangat yang besar untuk bisa menyelesaikan tugas akhir ini dengan lancar.
2. Kedua orang tua penulis, Bapak Budiyo, S.Pd., dan Ibu Hartati, yang tidak hanya memberikan fasilitas untuk dapat terus melanjutkan pendidikan, tapi juga selalu memberikan semangat serta doa terbaik sehingga diberikan kemudahan dan kelancaran dalam menyusun laporan tugas akhir ini.
3. Bapak Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
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. Ibu Dr. Novi Setiani, S.T., M.T., selaku Dosen Pembimbing Akademik yang telah bersedia membantu serta memberikan informasi terkait penjaluran magang kepada penulis.
7. Ibu Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs., selaku Dosen Pembimbing Tugas Akhir karena telah memberikan arahan, bimbingan, serta semangat kepada penulis untuk bisa menyelesaikan penulisan tugas akhir ini.
8. PT Lumintu Logic, beserta seluruh pimpinan dan staff yang telah memberikan kesempatan kepada penulis untuk melakukan magang dan membantu penulis untuk menyelesaikan tugas akhir ini.

9. Saudara kandung penulis, Ratno Hartadi, Sigit Hartadi, dan Wiwied Sertifiana yang telah memberikan dukungan dan arahan kepada penulis agar menjalankan perkuliahan dengan baik serta menjadi alasan bagi penulis untuk tumbuh menjadi lebih baik.
10. Teman kuliah penulis "Wong Angel, Cumlaude, CDR", yang telah menjadi orang-orang baik dan telah menemani perjalanan kuliah penulis selama menjalani perkuliahan.
11. Teman – teman penulis, Fadli Ananda, Bintang Muhammad Ralfi, Vian Ardiansyah, Kelsi Isha Kirana, Nafa Meilingga Putri, Icha Puspita, Pradya Haza Kurnia yang telah membersamai penulis semenjak pertama kali menginjakkan kaki di Yogyakarta untuk menempuh pendidikan.

Penulis menyadari bahwa dalam penulisan laporan tugas akhir ini masih jauh dari kata sempurna, oleh karena itu diperlukan kritik serta saran yang membangun dari berbagai pihak sehingga nantinya laporan tugas akhir ini menjadi lebih baik. Penulis berharap laporan tugas akhir ini bisa memberikan manfaat untuk menambah wawasan dan pengetahuan bagi pembaca.

Yogyakarta, 22 Juli 2025



Tegar Bagus Hartadi

SARI

Penelitian ini membahas bagaimana proses pengembangan basis data multilingual menggunakan *Headless* CMS Directus pada website Radyakartiyasa yang dimiliki oleh Karaton Yogyakarta. Kegiatan magang ini dilakukan di PT Lumintu Logic, sebuah perusahaan penyedia layanan produk dan jasa IT dari Yogyakarta, dengan penulis berperan sebagai *Back end Developer*. Selama magang, penulis mengikuti dua fase utama, yaitu *training* dan pelaksanaan proyek pengembangan basis data. *Training* difokuskan pada penguasaan teknologi yang digunakan perusahaan, seperti Directus dan Docker, serta penentuan peran penulis sebagai *Back end Developer*.

Manajemen proyek menggunakan metode *Agile Scrum* yang terdiri dari tahapan *product backlog, sprint planning, daily scrum, sprint review*, dan *sprint retrospective*. Dalam proyek ini, penulis bertanggung jawab melakukan instalasi dan konfigurasi Directus menggunakan Docker, perancangan koleksi basis data multilingual, serta pengujian API menggunakan Postman. Basis data dirancang agar mendukung pengelolaan konten multibahasa dengan relasi antar koleksi yang memungkinkan pengaturan konten dalam berbagai bahasa sesuai kebutuhan website pariwisata Karaton Yogyakarta.

Hasil magang menunjukkan bahwa penggunaan Directus sebagai *low-code Headless* CMS mampu mempercepat proses pembuatan dan pengelolaan basis data multilingual tanpa memerlukan penulisan query SQL manual. Selain memberikan pengalaman teknis, magang ini juga memberikan pemahaman pentingnya manajemen proyek yang terstruktur melalui metode *Agile Scrum* dalam pengembangan sistem informasi.

Kata kunci: Basis data multilingual, Headless CMS, Directus, Agile Scrum, Back end Developer, Docker, Postman, Website Radyakartiyasa.

GLOSARIUM

Agile Scrum	Metodologi manajemen proyek yang bersifat iteratif dan inkremental, digunakan untuk mengelola proses pengembangan perangkat lunak dengan pembagian kerja dalam sprint, yang meliputi perencanaan, pelaksanaan, evaluasi, dan perbaikan berkelanjutan.
Back end Developer	Peran dalam pengembangan perangkat lunak yang fokus pada pembuatan dan pengelolaan sisi server, termasuk desain basis data, API, dan logika aplikasi yang mendukung front end.
Basis Data Multilingual	Struktur data yang dirancang untuk mendukung penyimpanan dan pengelolaan konten dalam berbagai bahasa, sehingga memungkinkan website untuk menampilkan informasi dalam beberapa bahasa sesuai kebutuhan pengguna.
Docker Compose	Alat untuk mendefinisikan dan menjalankan aplikasi berbasis container Docker secara multi-kontainer, memudahkan pengaturan lingkungan pengembangan yang konsisten.
Directus	Platform Headless CMS open-source yang digunakan untuk mengelola konten dan struktur basis data secara low-code, memungkinkan pembuatan dan pengelolaan basis data multilingual tanpa perlu menulis query SQL manual.
Deployment	Proses mengaktifkan dan menyebarkan aplikasi atau layanan agar dapat diakses secara online oleh pengguna atau tim pengembang melalui jaringan internet.
Koleksi	Struktur penyimpanan data dalam Directus yang menyerupai tabel pada basis data relasional, digunakan untuk menyimpan berbagai jenis konten atau data aplikasi.
Low-Code Platform	Platform pengembangan yang memungkinkan pembuatan aplikasi dan pengelolaan basis data dengan

	sedikit atau tanpa penulisan kode secara manual, mempercepat proses pengembangan.
Multilingual	Kemampuan suatu sistem untuk mendukung dan menampilkan konten dalam beberapa bahasa, penting untuk menjangkau audiens yang lebih luas dan beragam.
Postman	Tool untuk menguji API yang digunakan untuk memastikan endpoint yang dibuat dari back end berfungsi sesuai dengan spesifikasi dan kebutuhan aplikasi front end.

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 xii
DAFTAR TABEL xiv
DAFTAR GAMBAR xv
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Ruang Lingkup 3
1.3. Batasan aktivitas 3
1.4. Tujuan 4
1.5. Manfaat 4
1.6. Sistematik Penulisan 5
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA 6
2.1. Website 6
2.2. Back end Developer 6
2.3. Basis Data 7
2.4. Headless Content Management System 7
2.5. Directus 8
2.6. Postman 9
2.7. Docker Compose 10
2.8. Agile Scrum 10
2.9. Tinjauan Pustaka 12
BAB III PELAKSANAAN MAGANG 14
3.1. Manajemen Proyek 14
3.2. Hasil Pengembangan Proyek 22
3.2.1. Konfigurasi Directus 22
3.2.2. Kumpulan koleksi basis data 25
3.2.3. Pembuatan Koleksi Language 27
3.2.3. Pembuatan Koleksi Interface 28
3.2.4. Pembuatan Koleksi Site Settings 30
3.2.5. Pembuatan Koleksi Destination 32
3.2.6. Pembuatan Koleksi Event 35
3.2.7. Pembuatan Koleksi Testimonial 37
3.2.8. Pembuatan Koleksi Constant Libraries 39
3.2.9. Pembuatan Koleksi Galleries 41
3.2.10. Pembuatan Koleksi Contact-Us 43
3.2.11. Tampilan *Website* Radyakartiyasa 44
3.2.12. Pengujian Application Programming Interface (API) 45
BAB IV REFLEKSI PELAKSANAAN MAGANG 49
4.1. Relevansi Akademik 49
4.2. Pembelajaran Magang 51
BAB V PENUTUP 54

5.1. Kesimpulan 54
5.2. Saran 54
DAFTAR PUSTAKA 56
LAMPIRAN 58

DAFTAR TABEL

Tabel 2.1 Penelitian Sejenis 12
Tabel 2.2 Perbandingan Penelitian Sebelumnya 13

DAFTAR GAMBAR

Gambar 2.1 Scrum Framework 12
Gambar 3.1 Tampilan Deck Lakon Lumintu 18
Gambar 3.2 Tampilan *Visual Studio Code* 19
Gambar 3.3 Tampilan Docker Desktop Container Group Radyakartiyasa 19
Gambar 3.4 Tampilan Directus 21
Gambar 3.5 Tampilan Postman 22
Gambar 3.6 Kode Program File "`docker-compose.yaml`" 23
Gambar 3.7 Hubungan Antar Komponen Environment 24
Gambar 3.8 Tampilan kumpulan koleksi basis data 25
Gambar 3.9 Tampilan Entity Relationship Diagram (ERD) 26
Gambar 3.10 Tampilan Koleksi *Language* 28
Gambar 3.11 Tampilan koleksi "*interface*" dan sub koleksi "*interface_translations*" 29
Gambar 3.12 Tampilan item pada Koleksi *Interface* 30
Gambar 3.13 Tampilan Koleksi *Site Settings* 31
Gambar 3.14 Tampilan koleksi "*destination*" dan sub koleksi "*destination_translations*" .. 33
Gambar 3.15 Tampilan item pada Koleksi *Destination* 34
Gambar 3.16 Tampilan koleksi "*event*" dan sub koleksi "*event_translations*" 35
Gambar 3.17 Tampilan koleksi "*testimonial*" dan sub koleksi "*testimonial_translations*" .. 38
Gambar 3.18 Tampilan koleksi "*content_testimonial*" dan sub koleksi "*content_testimonial_translations*" 38
Gambar 3.19 Tampilan koleksi "*constant_libraries*" dan isi *content* koleksi "*constant_libraries*" 40
Gambar 3.20 Tampilan koleksi "*galleries*" dan sub koleksi "*galleries_translations*" 42
Gambar 3.21 Tampilan koleksi "*contact_us*" 43
Gambar 3.22 (a) Tampilan Website Radyakartiyasa (b) TampilanWebsite Radyakartiyasa 45
Gambar 3.23 Pengujian API *endpoint* "contact_us" 46
Gambar 3.24 Pengujian API *endpoint* "destination" 46
Gambar 3.25 Data JSON pada Raw Body "contact_us" 47
Gambar 3.26 Data JSON pada Raw Body "destination" 47

BAB I
PENDAHULUAN

1.1. Latar Belakang

Salah satu bentuk proyek nyata yang dikerjakan oleh PT Lumintu Logic adalah kolaborasi dengan pihak Karaton[1] Yogyakarta dalam rangka melakukan digitalisasi budaya melalui pengembangan website resmi yang bernama Radyakartiyasa. Website ini dikembangkan untuk mendukung kegiatan promosi dan pelestarian budaya Kagungan Dalem Karaton Yogyakarta. Latar belakang lahirnya proyek ini didasari oleh kebutuhan Karaton dalam melakukan revitalisasi unit-unit wisata serta meningkatkan daya tarik kunjungan, terutama pada generasi muda dan wisatawan mancanegara. Kawadenan Radyakartiyasa selaku pengelola museum dan pameran Karaton menyusun strategi jangka panjang dengan memanfaatkan media digital agar kekayaan budaya Karaton dapat disampaikan lebih luas dan mudah diakses oleh masyarakat.

PT Lumintu Logic merupakan sebuah perusahaan penyedia layanan produk dan jasa *information technology* (IT) yang berasal dari Yogyakarta, Indonesia. Lumintu Logic sendiri berasal dari bahasa jawa yang memiliki arti layanan berkelanjutan. Perusahaan ini lebih berfokus mengembangkan produk aplikasi untuk lingkangan *website* yang selalu mengikuti perkembangan yang ada pada industri IT dan terus memberikan layanan dengan berinvestasi pada infrastruktur baru yang digunakan perusahaan sebagai rumah bagi SaaS (*Software as a Service).* SaaS merupakan model layanan perangkat lunak berbasis cloud yang memungkinkan pengguna mengakses aplikasi melalui internet tanpa perlu melakukan instalasi atau pemeliharaan sistem secara langsung di perangkat pengguna.

Proyek pengembangan *website* Radyakartiyasa tidak hanya bertujuan sebagai media promosi budaya semata, tetapi juga sebagai portal digital yang informatif dan interaktif, mencakup konten seperti sejarah batik, dokumentasi acara, seni pertunjukan, hingga arsip-arsip penting milik Karaton. Mengingat audiens yang dituju mencakup masyarakat lokal maupun internasional, maka *website* ini harus mampu mendukung penyajian konten dalam dua bahasa atau lebih (bilingual/*multilingual*). Oleh karena itu, kebutuhan akan sistem basis data yang mampu menangani pengelolaan konten

[1] Penamaan ”Karaton” dikarenakan mengarah ke instansi, bukan lokasi.

multibahasa secara efisien menjadi prioritas utama dalam perancangan teknis proyek ini.

Dalam rangka menjawab kebutuhan tersebut, dipilihlah Directus sebagai *Headless CMS* dalam pengembangan basis data *website*. Directus dipilih karena merupakan platform *open-source*, yaitu perangkat lunak yang kode sumbernya terbuka dan dapat digunakan serta dimodifikasi secara bebas. Selain itu, Directus juga bersifat *low-code*, artinya Directus memungkinkan pengelolaan dan pengembangan sistem tanpa harus menulis banyak kode secara manual.

Directus sangat fleksibel dalam mengelola data konten secara struktural melalui antarmuka visual, tanpa perlu menulis *query* SQL secara manual. Kelebihan lainnya adalah kemampuannya dalam membangun struktur *multilingual-ready* yang memungkinkan setiap entri konten memiliki versi berbeda untuk tiap bahasa, serta dukungan API (REST & GraphQL) yang memudahkan tim *front end* mengambil data sesuai preferensi bahasa pengguna. Dengan arsitektur *Headless* ini, konten dapat disajikan dengan cara yang berbeda di setiap *platform* (web, mobile), namun tetap dikelola dari satu sumber basis data utama. Hal ini sejalan dengan kebutuhan fleksibilitas yang diinginkan oleh tim pengembang dalam menghadirkan pengalaman pengguna yang konsisten di berbagai kanal digital.

Aktivitas yang dilakukan penulis selama berada di perusahaan PT Lumintu Logic terbagi menjadi dua fase, fase pertama yaitu melakukan kegiatan *training* dengan tujuan untuk mempelajari teknologi yang digunakan pada pengembangan *website*. Selanjutnya fase kedua, penulis mengikuti kegiatan magang dengan posisi sebagai *Back end Developer* selama tiga bulan hingga kegiatan magang selesai.

Selama melaksanakan magang, penulis berkontribusi dengan mengerjakan pembuatan basis data pada proyek *website* Radyakartiyasa. Selama pembuatan basis data ini, penulis memanfaatkan teknologi Directus sebagai *Headless* CMS untuk pembuatan basis data serta Postman sebagai alat unutk menguji API dari basis data yang sudah penulis buat.

Penulis juga bertanggung jawab untuk melakukan *deployment* Directus agar dapat diakses secara *online* oleh tim pengembang tanpa harus menggunakan *localhost.* Selain itu, penulis juga turut serta mengikuti diskusi terkait tugas tugas yang diberikan untuk pengerjaan proyek ini.

1.2. Ruang Lingkup

Ruang lingkup dalam kegiatan magang ini mencakup proyek pengembangan sistem basis data *multilingual* untuk *website* resmi Kawadenan Radyakartiyasa yang bernama Radyakartiyasa. *Website* ini dibangun bertujuan unutk memperkenalkan kekayaan budaya yang dimiliki oleh Karaton Yogyakarta kepada masyarakat luas melalui media digital, baik itu lokal maupun internasional.

Lingkup proyek serta batasan aktivitas yang dilakukan penulis dalam pengembangan sistem basis data multilingual untuk *website* Radyakartiyasa ini dijabarkan sebagai berikut:

1. **Konfigurasi directus**

   Penulis bertanggung jawab untuk melakukan instalasi dan konfigurasi awal *Headless* CMS Directus menggunakan layanan Docker. Aktivitas ini mencakup pengaturan file `docker-compose.yaml`, konfigurasi variabel lingkungan (*environment variable)*, serta pembuatan akun administrator.

2. **Perancangan koleksi basis data multilingual**

   Penulis membuat struktur koleksi basis data yang dibutuhkan untuk *website multilingual*. Pengembangan dimulai dari membuat koleksi dasar seperti *Language* yang berguna untuk mempresentasikan daftar bahasa yang dapat digunakan untuk basis data *multilingual*, seterusnya penulis membuat koleksi lain seperti *Event, Destination, Testimonial, Supporters*, dan lainnya yang nantinya akan direlasikan dengan koleksi *Language* untuk mendukung sistem multibahasa.

3. **Penerapan metode agile scrum**

   Selama peengembangan sistem basis data ini, penulis juga mengikuti alur kerja proyek menggunakan kerangka kerja *Scrum* yang terdiri dari tahapan: *Product Backlog, Sprint Planning, Daily Scrum, Sprint Review, dan Sprint Retrospective.* Penulis berperan dalam dokumentasi setiap *sprint*, menyusun rencana *sprint*, dan melakukan evaluasi berkala bersama tim pengembang.

1.3. Batasan aktivitas

Batasan aktivitas merupakan lingkup kerja yang dilakukan oleh penulis selama mengikut kegiatan magang. Batasan ini bertujuan untuk memperjelas fokus dan

tanggung jawab penulis dalam proyek serta menjelaskan aktivitas apa saja yang tidak termasuk ke dalam lingkup kerja yang dilakukan. Berikut merupakan batasan aktivitas yang dilakukan oleh penulis:

1. Penulis tidak membuat *query* SQL secara manual menggunakan sistem manajemen basis data konvensional seperti MySQL, karena seluruh pembuatan basis data dilakukan menggunakan antarmuka grafis Directus.
2. Penulis tidak terlibat langsung dalam pengembangan fitur *front end* secara langsung menggunakan Astro.JS, namun hanya menyediakan struktur dan konten data multibahasa yang dapat diakses oleh tim *front end*.

1.4. Tujuan

Tujuan dari penulisan tugas akhir ini adalah untuk mendokumentasikan proses perancangan serta konfigurasi basis data multilingual menggunakan *Headless* CMS Directus dalam pengembangan *website* Radyakartiyasa. Dalam penulisan tugas akhir ini, ada pula tujuan khusus yang meliputi:

1. Membuat struktur basis data yang dapat mendukung pengelolaan konten multibahasa.
2. Menjelaskan penggunaan antarmuka grafis Directus dalam membangun dan memodifikasi basis data tanpa harus menuliskan *query* SQL secara manual.
3. Memastikan struktur basis data yang dibangun telah mampu memenuhi kebutuhan antarmuka *front end*.

1.5. Manfaat

Berikut merupakan manfaat yang bisa diambil dari penulisan tugas akhir ini:

1. Memberikan wawasan praktis tentang bagaimana cara mengelola atau mengembangkan konten basis data *multilingual* menggunakan *Headless* CMS.
2. Memberikan pemahaman baru kepada penulis mengenai pentingnya membuat basis data yang sesuai dengan kebutuhan konten pada bagian *front end* sehingga bisa digunakan dengan baik, karena data dari *back end* akan langsung ditampilkan di halaman *website*, jadi harus dibuat lengkap dan sesuai agar tampilan *website* bisa berjalan dengan benar dan rapi.

3. Memberikan pengalaman langsung kepada penulis dalam penggunaan *low-code tools* untuk merancang basis data guna mendukung percepatan pengembangan sistem informasi.

1.6. Sistematik Penulisan

Penyajian laporan skripsi ini menggunakan sistematika penulisan sebagai berikut:

1. Bagian Utama Skripsi

Bagian utama terbagi atas bab dan sub bab sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang penulisan laporan tugas akhir, ruang lingkup, tujuan, manfaat, dan sistematika penulisan laporan tugas akhir.

2. BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA

Bab ini membahas mengenai landasan teori yang berkaitan dengan pengembangan basis data serta hasil penelitian terdahulu yang membahas pengembangan basis data serupa.

3. BAB III PELAKSANAAN MAGANG

Bab ini membahas tentang aktivitas yang dilakukan selama pelaksanaan magang, penjelasan proyek di PT Lumintu Logic, serta pembahasan metode yang digunakan pada pengembangan.

4. BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini membahas tentang pembelajaran yang didapatkan selama menjalankan magang dan mengerjakan proyek magang, tantangan yang dihadapi, serta keterampilan yang berhasil dikembangkan selama magang.

5. BAB V PENUTUP

Bab ini membahas kesimpulan serta saran yang telah didapatkan dan yang akan dilakukan pada pengembangan selanjutnya.

BAB II
LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1. Website

Website merupakan kumpulan halaman yang saling terhubung dalam suatu domain dan dapat diakses melalui internet. *Website* biasanya menyajikan informasi berupa teks, gambar, video, audio, dan berbagai jenis konten multimedia lainnya yang dapat dibaca dan dilihat oleh pengguna internet melalui sebuah mesin pencari (Costa, 2022).

Sedangkan menurut (Nurlailah & Wardani, 2023), *website* merupakan salah satu media yang dapat dimanfaatkan untuk menyampaikan informasi dan mempromosikan suatu produk sehingga dapat dikenal serta diketahui oleh masyarakat secara luas. *Website* juga dapat dimanfaatkan sebagai alat komunikasi dua arah antara penyedia informasi dengan pengguna, sehingga interaksi dan umpan balik dapat tercipta dengan lebih efektif.

Selain sebagai sarana penyampaian informasi, *website* juga berperan penting dalam membangun citra dan kepercayaan terhadap suatu lembaga, organisasi, maupun perusahaan (Christoper, 2024). Dalam konteks pariwisata budaya, keberadaan *website* dapat menjadi jembatan antar pelestarian nilai-nilai kebudayaan dengan penyampaian infomasi secara global.

Dengan memanfaatkan teknologi digital yang terus berkembang, *website* mampu menjangkau audiens yang lebih luas, menyajikan informasi secara *real-time*, serta memberikan pengalaman interaktif yang dapat meningkatkan daya tarik suatu destinasi wisata.

2.2. Back end Developer

Back end developer merupakan seseorang yang bertanggung jawab dalam membangun dan mengelola sisi server dari sebuah aplikasi atau *website* (Sarmanela et al., 2023). Tugas utama seorang *back end developer* adalah merancang struktur basis data, membuat logika program, dan memastikan data dari server dapat tersaji dengan baik ke sisi pengguna (*front end*) melalui API (*Application Programming Interface*). Dengan kata lain, *back end developer* bekerja di balik layar untuk memastikan sistem

berjalan dengan lancar, data tersimpan dengan aman, serta permintaan dari pengguna dapat diproses dengan benar oleh sistem.

Dalam proyek pengembangan *website* Radyakartiyasa, penulis menjalankan peran sebagai *back end developer* dengan fokus utama pada pembuatan basis data yang mendukung fitur *multilingual*. Pengelolaan basis data dilakukan menggunakan Directus, *Headless* CMS yang memungkinkan pembuatan koleksi basis data secara visual tanpa perlu menulis *query* SQL secara manual. Penulis juga bertanggung jawab untuk mengelola relasi antar koleksi data, melakukan testing API menggunakan Postman, serta membantu proses *deployment* agar sistem dapat diakses oleh tim pengembang lainnya.

2.3. Basis Data

Menurut (Haryanto et al., 2024), basis data merupakan suatu kumpulan data yang saling terhubung atau relasi yang disimpan secara bersama-sama. Pada pengembangan sistem aplikasi *website*, basis data berfungsi sebagai penyimpan utama seluruh informasi yang akan disampaikan melalui antarmuka pengguna. Basis data juga perperan penting dalam menjaga konsistensi, integritas, serta keamanan data yang digunakan dalam proses penyajian konten. Oleh karena itu, pemilihan dan perancangan basis data menjadi aspek yang sangat krusial dalam membangun sistem informasi yang andal dan efisien (Apriza & Sutarbi, 2025).

Basis data *multilingual* adalah sistem basis data yang memungkinkan satu konten memiliki versi dalam beberapa bahasa. Hal ini penting agar informasi yang ditampilkan pada *website* dapat diakses oleh pengguna dari berbagai latar bahasa yang berbeda. Sementara itu, basis data tradisional biasanya dibangun menggunakan *query* SQL secara manual untuk membuat dan mengelola datanya. Dalam pengembangan basis data *multilingual* ini, tidak digunakan cara tradisional tersebut. Sebagai gantinya, digunakan Directus yang memiliki antarmuka visual dan mendukung pengelolaan konten multibahasa tanpa perlu menulis *query* SQL secara langsung, sehingga pengelolaan data menjadi lebih mudah dan efisien.

2.4. Headless Content Management System

Headless Content Management System atau *Headless* CMS adalah jenis sistem manajemen konten yang memisahkan antara pengelolaan konten (*back end*) dan

penyajian konten (*front end*) (Arifiany & Kusuma, 2025). Berbeda dengan CMS tradisional seperti WordPress yang menggabungkan dua bagian tersebut dalam satu sistem, *Headless* CMS hanya berperan sebagai tempat untuk menyimpan, mengelola, dan menyediakan konten, sedangkan tampilan antarmuka diserahkan sepenuhnya pada tim *front end* melalui API (*Application Programming Interface*).

Konsep "*Headless*" berarti sistem pengelola konten tidak langsung menampilkan kontennya ke *website*, melainkan hanya menyimpan dan mengelola datanya saja. Dengan *Headless* CMS, pengembang dapat menggunakan berbagai teknologi *front end* seperti React, Vue, atau AstroJS untuk menampilkan konten sesuai kebutuhan pengembang.

Keunggulan utama *Headless* CMS adalah fleksibilitas, skalabilitas, dan efisiensi dalam mengelola serta menyajikan konten lintas platform (Web, Mobile, IoT) (Berndtsson & Fagerberg, 2025). Selain itu, arsitektur ini juga sangat cocok untuk pengembangan sistem *multilingual*, karena struktur data dapat dirancang agar mendukung konten dalam berbagai bahasa tanpa terbatas pada tampilan bawaan sistem.

2.5. Directus

Directus merupakan *Headless* CMS (*Content Management System*) sekaligus platform manajemen data berbasis *open-source* yang dirancang untuk memberikan fleksibilitas tinggi dalam pengelolaan konten dan struktur basis data. Berbeda dengan CMS tradisional seperti WordPress yang menggabungkan pengelolaan konten dan tampilan (*front end*) dalam satu sistem, Directus hanya berfokus pada pengelolaan konten dan tidak mengatur tampilan akhir kepada pengguna. Selain itu, Directus menyediakan data melalui antarmuka pemrograman aplikasi (*Application Programming Interface*/API) dalam bentuk RESTful API yang menggunakan struktur URL dan metode HTTP seperti *GET , POST, PUT,* dan *DELETE* untuk mengakses data. Misalnya, untuk mengambil data dari koleksi bernama `"destination"`, pengembang cukup menggunakan metode GET dengan URL seperti "`https://backenddirectus.madebybagus.xyz/items/destination`", yang akan mengembalikan data dalam format JSON. Pendekatan ini memudahkan front end dalam mengambil data secara dinamis dari basis data tanpa harus langsung terhubung dengan sistem *back end*.

Sebagai *Headless* CMS, Directus tidak menentukan bagaimana data diampilkan kepada pengguna akhir, melainkan hanya menyediakan data mentah yang dapat diakses dan dimanfaatkan oleh berbagai jenis *front end*, seperti *website*, aplikasi *mobile*, maupun sistem *Internet of Things* (IoT). Hal ini memungkinkan pengembang untuk memiliki kebebasan penuh dalam merancang dan mengembangkan antarmuka pengguna sesuai dengan kebutuhan masing-masing *platform*.

Secara teknis, Directus dibangun menggunakan tumpukan teknologi (*technology stack*) modern yang mendukung fleksibilitas dan kinerja tinggi. Pada lapisan server-side, Directus dibangun menggunakan Node.Js sebagai lingkungan eksekusi *JavaScript* yang bersifat *cross-platform* dan *event-driven*, sehingga mampu menangani proses secara asinkron dengan performa tinggi. Node.js memanfaatkan Express.js sebagai *web server framework* untuk menangani *HTTP request* dan *routing* secara efisien.

Untuk interaksi dengan basis data, Directus menggunakan Knex.js sebagai *SQL query builder* yang bersifat agnostik terhadap jenis sistem manajemen basis data relasional (RDBMS) seperti MySQL, PostgreSQL, dan SQLite. Penggunaan Knex.js memberikan fleksibilitas dalam berpindah antar basis data tanpa memerlukan perubahan signifikan pada kode.

Pada sisi antarmuka (*front end*), Directus menggunakan Vue.js sebagai *JavaScript framework* untuk membangun Directus Data Studio, yaitu antarmuka pengguna interaktif yang memudahkan pengelolaan konten dan konfigurasi sistem secara visual.

Selain itu, Directus memanfaatkan REST API dan GraphQL API untuk menyediakan akses data yang fleksibel bagi aplikasi pihak ketiga, serta menggunakan *WebSocket* untuk mendukung komunikasi *real-time* seperti *live updates* pada konten. Kombinasi teknologi ini menghasilkan sebuah sistem yang efisien, mudah diintegrasikan, dan dapat beradaptasi dengan berbagai kebutuhan pengembangan aplikasi modern.

2.6. Postman

Postman merupakan aplikasi yang dapat digunakan untuk melakukan pengujian pada *Application Programming Interface* (API) dalam sebuah proses pengembangan perangkat lunak (Pratama et al., 2023). Aplikasi ini menyediakan antarmuka grafis yang

dapat digunakan untuk melakukan permintaan (*request*) dan menerima respons (*response*) dari server. Postman memberikan beberapa metode HTTP yang dapat digunakan seperti, *GET, POST, PUT*, dan *DELETE* untuk memastikan apakah API yang dikembangkan dapat bekerja dengan baik sesuai dengan fungsi yang diharapkan.

Dalam pengembangan ini, Postman juga memiliki kaitan dengan pengembangan yang menggunakan *Headless* CMS seperti Directus. Setelah struktur basis data dan konten berhasil dibuat di Directus, API secara otomatis tersedia dan dapat di uji dan divalidasi menggunakan Postman sebelum akhirnya bisa untuk digunakan oleh *front end* maupun layanan lain.

2.7. Docker Compose

Docker Compose merupakan alat bantu yang disediakan oleh Docker untuk mengelola dan menjalankan *container* yang berjumlah lebih dari satu secara bersamaan menggunakan satu konfigurasi yang terstruktur, yaitu "`docker-compose.yaml`" (Isron & Putra, 2021). Ini berguna sekali dalam pengembangan aplikasi dari banyak layanan (*multi-service*), seperti *front end* dan *back end* yang perlu dijalankan secara terintegrasi (Ningsih et al., 2024).

Melalui Docker Compose ini pengembang dapat mendefinisikan seluruh layanan beserta konfigurasinya seperti *image*, *port*, *volume*, dan jaringan dalam satu file. Hal ini mempermudah proses pengaturan, penyebaran, dan replikasi lingkungan aplikasi berbagai sistem. Dengan menggunakan Docker Compose dan memberikan perintah seperti "`docker-compose up`", maka seluruh komponen aplikasi dapat dijalankan secara otomatis, efisien, dan konsisten (Bik & Asmunin, 2017).

2.8. Agile Scrum

Agile merupakan metode yang sering digunakan dalam pengembangan sistem perangkat lunak(Atallah & Mardi, 2024). *Scrum* merupakan kerangka kerja dari metode *Agile* yang biasanya digunakan oleh tim pengembang dalam bekerja untuk mencapai tujuan yang diinginkan (Riana, 2021). Penggunaan *Scrum* dalam pengembangan perangkat lunak dapat dimanfaatkan secapa optimal sesuai dengan kebutuhan yang ada (Amarta & Anugrah, 2021), baik dalam membantu tim pengembang untuk bekerja secara lebih terstruktur dan efisien maupun dalam menyelesaikan berbagai permasalahan yang timbul selama proses pengembangan dengan tujuan untuk

mendapatkan peningkatan kualitas produk akhir dari sistem perangkat lunak yang dikembangkan (Rizky & Sugiarti, 2022).

1. *Product Backlog*

   Tahapan *Product Backlog* ini merupakan daftar yang berisikan seluruh kegiatan yang akan dilakukan pada pengembangan ini.

2. *Sprint Planning*

   Selanjutnya yaitu tahapan *Sprint Planning*, dimana tim menentukan prioritas utama dan membagi pekerjaan kedalam *sprint-sprint* yang memiliki target waktu tertentu.

3. *Daily Scrum*

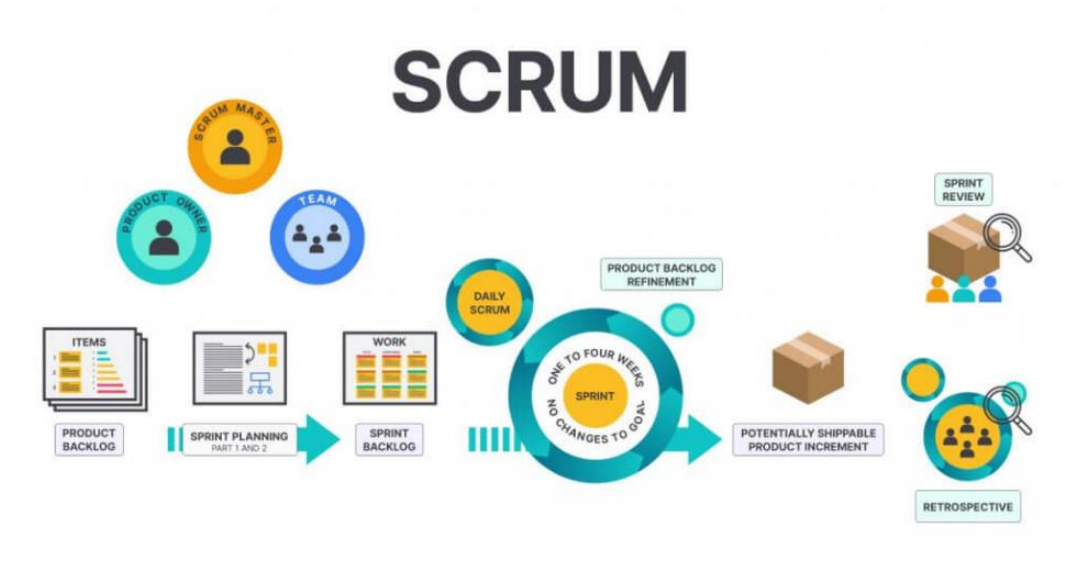
   Selanjutnya yaitu tahapan *Daily Scrum*, dimana tim melakukan kegiatan laporan dengan durasi singkat yang bertujuan mengevaluasi progres harian tim yang sedang melakukan pengembangan.

4. *Sprint Review*

   Tahapan selanjutnya yang dilakukan dalam pengembangan ini yaitu *Sprint Review*, kegiatan ini dilakukan untuk melihat perkembangan pekerjaan yang sudah dilakukan selama satu minggu serta untuk menentukan penyesuaian yang harus dilakukan nantinya di pengembangan selanjutnya.

5. *Sprint Retrospective*

   Setelah satu *sprint*, tim masuk ke tahapan yang terakhir yaitu *Sprint Retrospective*, kegiatan ini dilakukan dengan tujuan mengevaluasi progres kinerja tim, mengidentifikasi setiap tugas yang berjalan dengan baik, menentukan tugas-tugas yang memerlukan perbaikan, serta merangcang strategi selanjutnya guna meningkatkan pencapaian tim pada *sprint* selanjutnya.



Gambar 2.1 Scrum Framework
Sumber: (Alwi, 2024)

2.9. Tinjauan Pustaka

Berikut ini sejumlah penelitian yang telah dilakukan sebelumnya yang membahas topik mengenai pengembangan basis data. Adapun hasilnya sebagai berikut:

Tabel 2.1 Penelitian Sejenis

No	Judul	Tahun	Pembahasan
1	Perancangan Website Sebagai Media Informasi Dan Promosi Oleh-Oleh Khas Kota Pagaralam (Nurlailah & Wardani, 2023)	2023	Penelitian ini membahas tentang perancangan *website* sebagai media informasi dan promosi oleh-oleh khas Kota Pagaralam dengan tujuan untuk memberikan kemudahan kepada wisatawan dalam memperoleh informasi serta membantu pelaku usaha lokal memasarkan produknya secara digital.
2	Website Event Sebagai Daya Tarik Pariwisata Kabupaten Karo Menggunakan Fitur Multilanguage (Sinulingga et al., 2023)	2023	Penelitian ini membahas tentang perancangan *website event* milik Dinas Kebudayaan dan Pariwisata Kabupaten Karo yang mendukung fitur *multilanguage*. *Website* ini ditujukan untuk memudahkan wisatawan, baik lokal maupun mancanegara, dalam mengakses informasi terkait *event-event* yang diselenggarakan.
3	Penerapan Metode Agile Scrum Pada Sistem E-Posyandu Berbasis Web (Christine et al., 2024)	2024	Penelitian ini membahas tentang bagaimana penggunaan metode *Agile Scrum* digunakan dalam merancang sistem informasi E-Posyandu. Dalam jurnal ini dijelaskan bahwa penggunaan metode *Agile Scrum* mempermudah perancangan sistem secara sistematis dan terstruktur, sehingga proses pengembangan menjadi lebih terarah.

Tabel 2.2 Perbandingan Penelitian Sebelumnya

No	Multilingual	Menggunakan Headless CMS	Promosi Pariwisata dan Budaya	Target Wisatawan	Menggunakan Agile Scrum
(Nurlailah & Wardani, 2023)	Tidak	Tidak	Tidak	Ya	Tidak
(Sinulingga et al., 2023)	Ya	Tidak	Ya	Ya	Tidak
(Christine et al., 2024)	Tidak	Tidak	Tidak	Tidak	Ya
Penelitian Ini	Ya	Ya	Ya	Ya	Ya

Berdasarkan tabel perbandingan, dapat disimpulkan bahwa penelitian ini memiliki keunggulan karena menggabungkan beberapa aspek yang belum sepenuhnya dimanfaatkan dalam penelitian sebelumnya. Penelitian ini mendukung fitur multilingual, menggunakan *Headless* CMS, memfokuskan pada promosi pariwisata dan budaya, menargetkan wisatawan, serta menerapkan metode *Agile Scrum*. Sementara itu, penelitian lain hanya menonjol pada sebagian aspek saja dan belum mengintegrasikan semua pendekatan tersebut secara menyeluruh.

BAB III
PELAKSANAAN MAGANG

3.1. Manajemen Proyek

Berikut merupakan tahapan manajemen proyek yang dilakukan dalam pengembangan basis data *multilingual* pada *website* Radyakartiyasa Karaton Yogyakarta:

1. Training

Training merupakan tahapan awal dari kegiatan pengembangan basis data *multilingual* pada *website* Radyakartiyasa. Aktivitas-aktivitas yang dilakukan pada fase *training* ini terdiri dari 45 pertemuan yang dibagi menjadi dua topik:

a. Topik pertama terdiri dari 30 pertemuan yang membahas tentang IT *skill* seperti: *Directus for Back end, AstroJS: The Only framework you need, Deployment in Cloud: Cloudflare, System Architecture & Integration, Database Development, Project Management & Agile, Implementation*
b. Topik kedua terdiri dari 15 pertemuan yang membahasa tentang Non-IT *skill* seperti: *Communication skill, law in IT, English for IT, Career Preparation*

Kedua jenis pertemuan tersebut diselenggarakan secara paralel per pekannya, misalnya satu pekan berisi materi IT,dan pekan berikutnya berisi materi gabungan langsung antara materi IT dan non IT. Pada tahap ini penulis melakukan *training* dengan tujuan menentukan peran penulis dalam proyek sesuai dengan *tools* yang sanggup dikuasai selama kegiatan *training*. Sejumlah peran serta *tools* yang sempat penulis pelajari dalam kegiatan ini yaitu sebagai berikut:

a. *Front End Developer*: mempelajari *tools* serta bahasa pemrograman dasar yang digunakan serta mempelajari *framework* Astro.Js sesuai dengan yang digunakan oleh perusahaan dalam pengembangan proyek ini. Adapun kegiatan yang dilakukan pada masa *training* untuk mempelajari posisi *Front End Developer* seperti melakukan pengenalan dengan *framework* AstroJS dan melakukan instalasi AstroJS untuk pertama kalinya, serta menerapkan langsung *framework* tersebut pada *fake project* yang telah disiapkan sebelumnya sebagai latihan. Adapun *fake project* yang dikerjakan saat masa training adalah sebuah proyek latihan menggunakan template *website* yang

diambil dari internet. Tujuan dari proyek ini adalah untuk belajar cara menggunakan *framework* AstroJS secara langsung. Dalam latihan ini, penulis mencoba membagi bagian-bagian *website* seperti *header* dan *footer* menjadi komponen yang terpisah agar lebih mudah dikelola.

b. *Back End Developer*: mempelajari *tools* Directus yang merupakan *Headless* CMS yang digunakan oleh perusahaan untuk membuat basis data dalam pengembangan proyek ini. Adapun kegiatan yang dilakukan pada masa *training* untuk mempelajari posisi *Back End Developer* seperti melakukan pengenalan dengan Directus dan melakukan instalasi Directus untuk pertama kalinya, serta langsung membuat contoh basis data sesuai dengan yang ada pada dokumentasi resmi Directus sebagai bahan latihan dalam menggunakan Directus.

Berdasarkan kegiatan *training* yang telah dilakukan dan hasil dari penilaian dari Supervisor yang menangani proyek ini, penulis diberikan peran sebagai *Back End Developer* dikarenakan penilaian penguasaan *tools* selama kegiatan *training* dan bertanggung jawab membuat basis data *multilingual* yang dibutuhkan untuk *website* Radyakartiyasa.

2. Pendefinisian Proyek

Setelah tahap *training* selesai dilakukan dan peran penulis telah ditetapkan, selanjutnya masuk ke tahap pendefinisian proyek pengembangan basis data multilingual pada *website* Radyakartiyasa. *Website* Radyakartiyasa merupakan *website* yang dimiliki oleh *Kagungan Dalem* Karaton Yogyakarta dan dikelola oleh Kawadenan Radyakartiyasa. Pengembangan proyek ini dilakukan dengan tujuan untuk memperkenalkan kekayaan yang dimiliki oleh Karaton Yogyakarta seperti batik, seni, dan tradisi yang menjadi warisan Karaton Yogyakarta dan diharapkan dapat menjangkau audiens dengan lebih luas, baik secara lokal maupun internasional.

Dengan tujuan agar nantinya *website* ini dapat dijangkau hingga internasional, dibutuhkan pula basis data *multilingual* yang dapat mendukung pengembangan *website* Radyakartiyasa ini agar dapat menampilkan konten yang mendukung beberapa bahasa. *Website* ini akan dilengkapi dengan dukungan

multibahasa untuk seluruh konten utama, termasuk bahasa Indonesia dan bahasa Inggris, serta kemungkinan penambahan bahasa lain di masa mendatang.

Selain itu, *website* akan memuat halaman informasi budaya yang mencakup penjelasan tentang batik, seni, tradisi, dan sejarah Karaton Yogyakarta, galeri multimedia (foto dan video) dengan deskripsi multilingual, acara budaya , serta kontak dan formulir interaktif yang dapat digunakan oleh pengunjung dari berbagai negara.

Pengembangan proyek ini berlangsung selama tiga bulan, dimulai pada Januari 2025 dan berakhir pada Maret 2025. Tim pengembang terdiri dari Product Owner (perwakilan Kawadenan Radyakartiyasa), Scrum Master (Project Manager dari PT. Lumintu Logic Indonesia), Backend Developer, Frontend Developer, serta UI/UX Designer.

Metode *Agile Scrum* digunakan dalam proyek pengembangan basis data multilingual *website* Radyakartiyasa ini. *Scope* pekerjaan selaku *Back End Developer* yaitu membuat basis data yang diperlukan untuk *website* serta yang mendukung fitur multibahasa. Tahapan pengembangan menggunakan Scrum diawali dengan *Product Backlog*, yaitu penyusunan daftar seluruh kegiatan yang akan dilakukan selama proses pengembangan.

Setelah itu dilanjutkan dengan *Sprint Planning*, di mana tim menentukan prioritas utama dan membagi pekerjaan ke dalam sprint-sprint dengan target waktu tertentu. Setiap harinya dilakukan *Daily Scrum*, yaitu pertemuan singkat untuk melaporkan progres dan mengevaluasi pekerjaan yang telah dicapai. Setelah satu sprint berjalan, dilakukan *Sprint Review* untuk melihat perkembangan pekerjaan yang telah dilakukan serta menentukan penyesuaian yang diperlukan untuk sprint selanjutnya. Dalam pengerjaan proyek ini, penulis melakukan kegiatan *Sprint Review* sebanyak satu kali dalam seminggu dan dilakukan pada setiap hari kamis

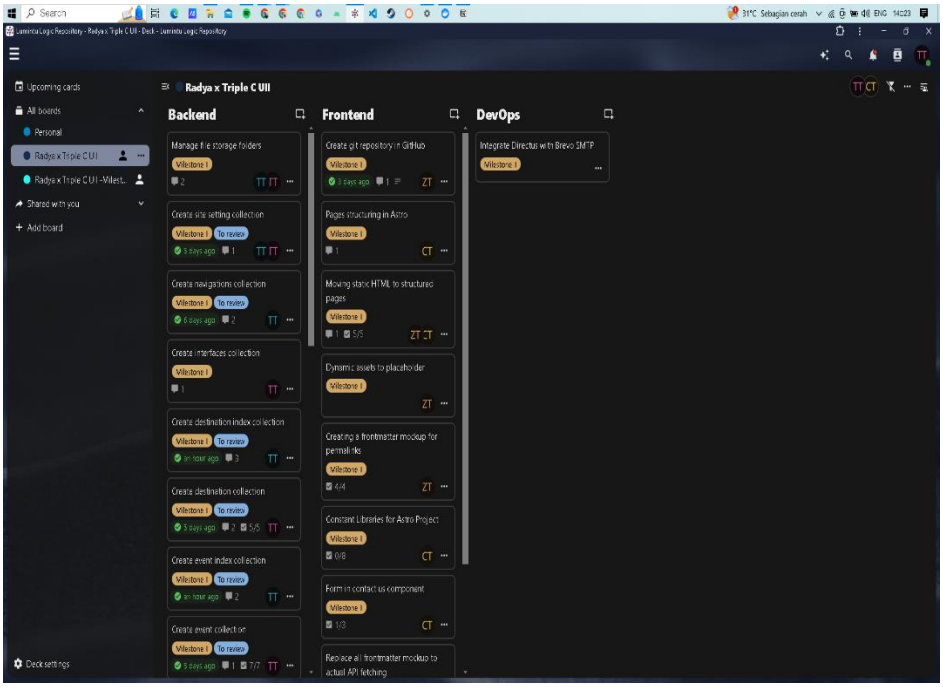
Terakhir, dilaksanakan *Sprint Retrospective* yang bertujuan untuk mengevaluasi kinerja tim, mengidentifikasi hal-hal yang berjalan dengan baik maupun yang perlu diperbaiki, serta merancang strategi untuk meningkatkan pencapaian pada sprint berikutnya. Untuk kegiatan *Sprint Retrospective* ini rutin dilakukan setiap satu bulan sekali di akhir bulan untuk melakukan evaluasi pekerjaan yang telah dilakukan selama sprint berjalan.

3. Pelaksanaan Proyek

Pelaksanaan proyek dilakukan setelah seluruh tahapan sebelumnya selesai dilakukan. Pengerjaan basis data *multilingual* pada *website* Radyakartiyasa dilakukan sesuai dengan metode *Agile Scrum* agar pengembangan dapat terpantau dengan baik dan hasil yang diberikan memiliki kualitas yang baik juga dan sesuai dengan kebutuhan. Pengerjaan basis data multilingual ini dilakukan mulai dari melakukan konfigurasi Directus, membuat koleksi basis data, hingga melakukan *testing* API. Semua kegiatan dilakukan dan dibuat menggunakan *tools-tools* sebagai berikut:

a. Lakon Lumintu

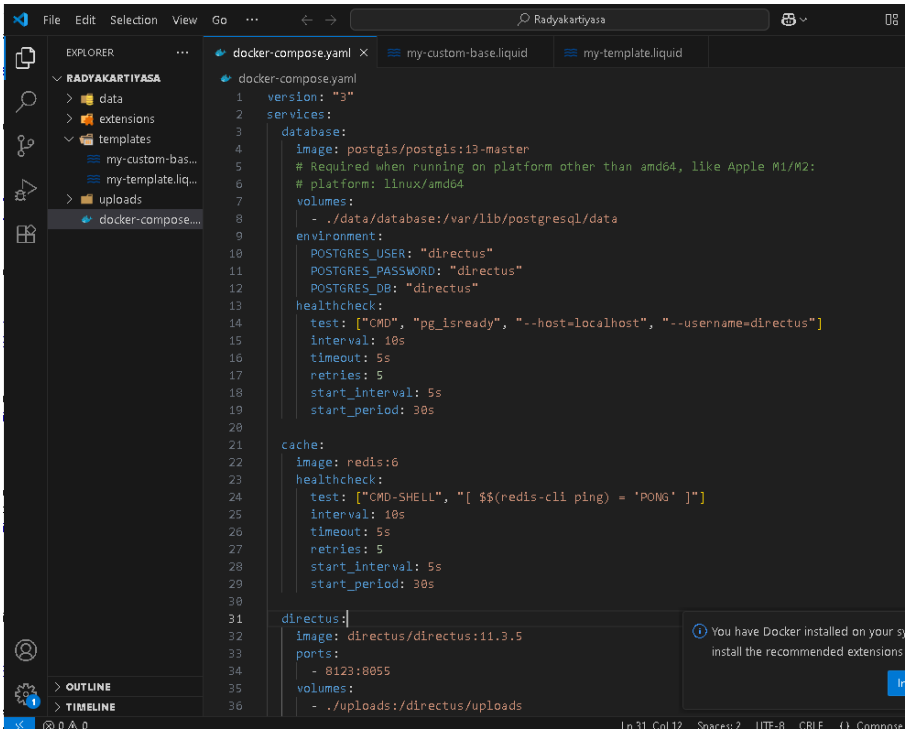
Lakon merupakan *tool* milik PT Lumintu Logic Indonesia yang digunakan oleh seluruh pegawainya dalam melakukan kegiatan pekerjaan dengan dukungan berbagai fitur seperti *Deck* atau daftar pekerjaan, *talk room chat*, *Appointment*, serta banyak lagi fitur yang dapat digunakan. Dalam proyek ini, Lakon digunakan oleh penulis untuk melihat daftar pekerjaan yang akan atau sedang dikerjakan sesuai *Sprint* yang sedang berjalan pada bagian atau fitur *Deck*. Fitur *Deck* ini berfungsi layaknya papan tugas digital, di mana setiap pekerjaan ditampilkan dalam bentuk kartu yang dapat dipindah-pindah sesuai dengan statusnya, seperti *To Do*, *In Progress*, hingga *Done*. Hal ini memudahkan tim untuk memantau progres tugas secara visual dan terorganisir. Pada Gambar 3.1 merupakan tampilan dari Lakon Lumintu yang digunakan oleh seluruh tim yang ada di PT Lumintu Logic.



Gambar 3.1 Tampilan Deck Lakon Lumintu

b. Visual Studio Code

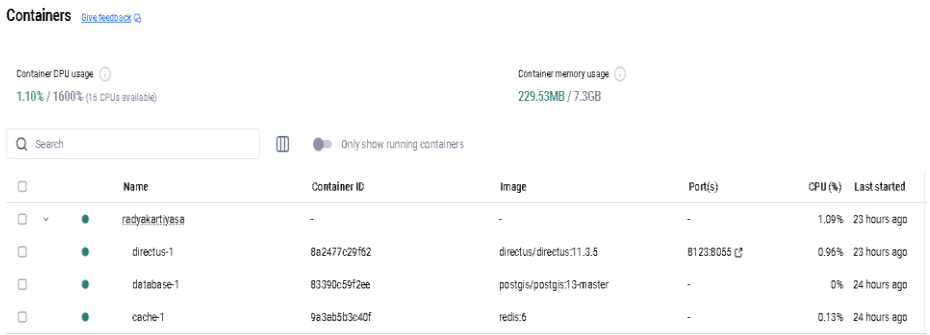
Visual Studio Code merupakan *tools source-code editor* yang mendukung berbagai bahasa pemrograman serta teknologi pengembangan web. Pada proyek ini, *visual studio code* digunakan untuk membuat file konfigurasi Directus yang bernama "`docker-compose.yaml`". Pada Gambar 3.2 merupakan tampilan file "`docker-compose.yaml`" pada aplikasi *Visual Studio Code*.



Gambar 3.2 Tampilan *Visual Studio Code*

c. Docker Compose

Docker Compose merupakan *tool* yang digunakan untuk mendefinisikan dan menjalankan aplikasi multi-kontainer Docker. Dalam proyek ini, Docker mempermudah proses setup lingkungan pengembangan dengan hanya menjalankan satu perintah "`docker compose up`". Dengan perintah tersebut, beberapa container langsung dijalankan sekaligus sesuai dengan file konfigurasi, seperti "`docker-compose.yaml`". Pada Gambar 3.3 merupakan tampilan Docker Desktop pada proyek Radyakartiyasa.



Gambar 3.3 Tampilan Docker Desktop Container Group Radyakartiyasa

Adapun fungsi dari masing masing *image* yang berjalan pada containers Radyakartiyasa sebagai berikut:

1. *Image* `directus/directus:11.3.5`

   Image ini merupakan *image* dari aplikasi Directus versi 11.3.5, yaitu *Headless* CMS yang digunaka untuk mengelola struktur basis data. *Container* ini berperan sebagai *back end* dari proyek Radyakartiyasa.

2. *Image* `postgis/postgis:13-master`

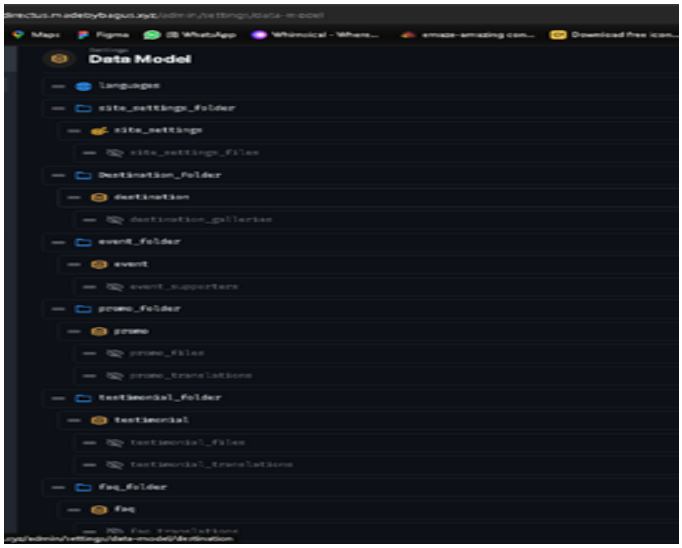
   Image ini merupakan turunan dari PostgreSQL yang sudah dilengkapi dengan ekstensi *PostGIS*, memungkinkan basis data untuk mendukung data spasial (misalnya koordinat atau peta). Container ini berfungsi sebagai sistem manajemen basis data (DBMS) untuk menyimpan semua konten dan struktur data yang diatur oleh Directus.

3. *Image* `redis:6`

   Redis digunakan sebagai *in-memory cache database* yang mempercepat proses pengambilan data. Dalam konteks penggunaan bersama Directus, Redis membantu dalam *caching* data agar akses ke API lebih cepat dan efisien.

d. Directus

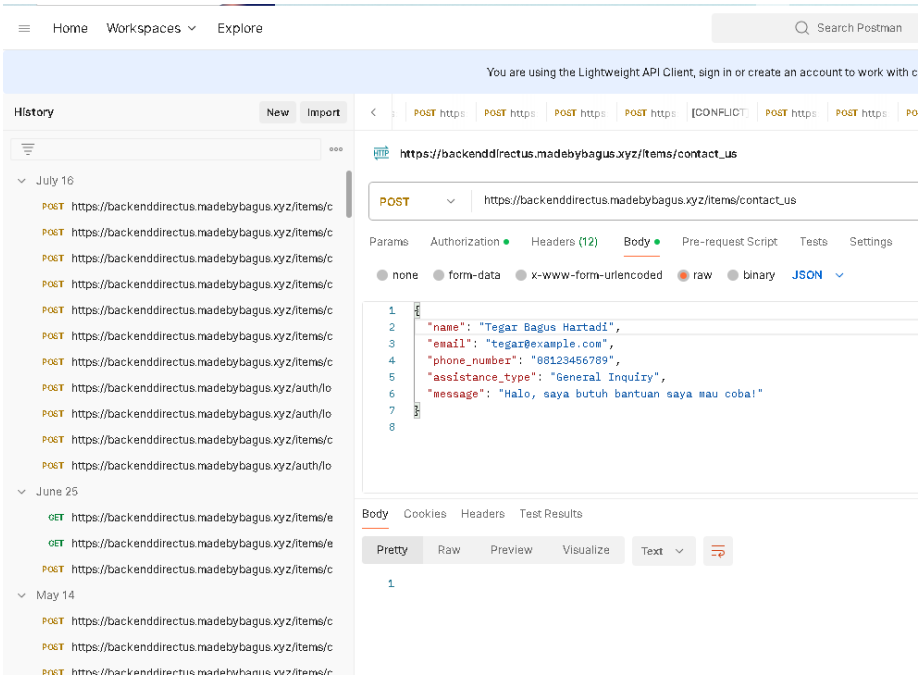
Directus merupakan *Headless content management system* (CMS) yang digunakan untuk melakukan pengembangan basis data *multilingual*. Directus dipilih karena Directus ini sendiri merupakan *Low-Code Platform* yang meminimalkan penulisan kode secara manual dan mengandalkan antarmuka visual untuk membuat basis data sehingga dapat mempercepat proses pengembangan. Pada Gambar 3.4 merupakan tampilan antarmuka Directus yang berisikan daftar basis data yang sudah penulis buat selama mengerjakan proyek ini.



Gambar 3.4 Tampilan Directus

e. Postman

Postman merupakan *tools* yang digunakan untuk menguji *endpoint* API yang dihasilkan dari sistem *back end*. Kaitan penggunaan *tools* ini dengan proyek yang dikerjakan ialah digunakan untuk menguji *collections* yang telah dibuat di Directus, seperti *collections event, destinations*, dan sebagainya. Pada Gambar 3.5 merupakan tampilan dari Postman yan sedang menunjukkan proses mengirim data ke koleksi "`contact_us`" di Directus dengan menggunakan metode *POST* dengan alamat API "`https://backenddirectus.madebybagus.xyz/items/contact_us`"



Gambar 3.5 Tampilan Postman

3.2. Hasil Pengembangan Proyek

Berikut merupakan hasil dari pengembangan proyek yang sudah dilakukan pada PT. Lumintu Logic Indonesia. Pengembangan basis data *multilingual* yang dilakukan oleh penulis mendapatkan beberapa hasil seperti:

3.2.1. Konfigurasi Directus

Proses pengembangan basis data multilingual untuk website Radyakartiyasa ini dimulai dengan tahapan awal berupa konfigurasi Headless CMS Directus yang nantinya akan digunakan sebagai pengelola konten dan struktur basis data.

Langkah awal untuk melakukan konfigurasi pada Directus ini yaitu dengan menyiapkan file "`docker-compose.yaml`" yang disediakan oleh Directus pada dokumentasi resmi Directus. Beberapa bagian penting yang ada pada file "`docker-compose.yaml`" perlu disesuaikan mengikuti kebutuhan proyek, seperti konfigurasi *port* dengan tujuan agar *port* tidak bentrok dengan layanan lain, serta pengaturan akun untuk administrator agar dapat mengakses Directus melalui variabel "`ADMIN_EMAIL`" dan "`ADMIN_PASSWORD`". Pada kode yang penulis lampirkan di Gambar 3.6, "`ADMIN_EMAIL`" dan "`ADMIN_PASSWORD`" harus penulis samarkan karena menyangkut privasi proyek.

```
services:
  database:
```

```
    image: postgis/postgis:13-master
    volumes:
      - ./data/database:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: "directus"
      POSTGRES_PASSWORD: "directus"
      POSTGRES_DB: "directus"

cache:
    image: redis:6

directus:
    image: directus/directus:11.3.5
    ports:
      - 8123:8055
    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: "**************@example.com"
      ADMIN_PASSWORD: "************"
```

Gambar 3.6 Kode Program File "`docker-compose.yaml`"

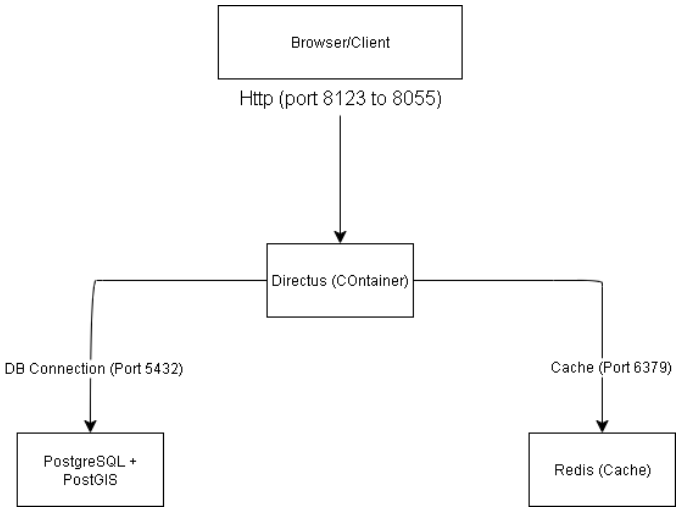
Pada konfigurasi ini terdapat tiga layanan utama, yaitu database, cache, dan Directus sebagai *Headless* CMS. Layanan database menggunakan *image* `postgis/postgis:13-master`, yaitu PostgreSQL versi 13 yang telah dilengkapi ekstensi PostGIS untuk mendukung pengelolaan data geospasial. Data dari layanan ini disimpan secara persisten menggunakan volume `./data/database:/var/lib/postgresql/data` agar tidak hilang meskipun container dihentikan. Pengaturan kredensial database dilakukan melalui variabel lingkungan `POSTGRES_USER`, `POSTGRES_PASSWORD`, dan `POSTGRES_DB` yang masing-masing diisi dengan nilai "`directus`".

Layanan cache menggunakan `image redis:6` yang berfungsi sebagai sistem cache untuk mempercepat pengambilan data dari Directus. Redis ini terintegrasi dengan Directus melalui konfigurasi cache pada *environment* Directus. Layanan Directus itu sendiri dijalankan menggunakan `image directus/directus:11.3.5` dan dihubungkan dengan port lokal 8123 yang

diarahkan ke port default Directus 8055 pada container, sehingga dapat diakses melalui alamat http://localhost:8123.

Konfigurasi koneksi Directus ke database PostgreSQL dilakukan melalui variabel `DB_CLIENT`, `DB_HOST`, `DB_PORT`, `DB_DATABASE`, `DB_USER`, dan `DB_PASSWORD`. Untuk sistem cache, Directus diatur agar aktif dengan `CACHE_ENABLED: "true"`, pembersihan otomatis cache diaktifkan melalui `CACHE_AUTO_PURGE: "true"`, penyimpanan cache menggunakan Redis (`CACHE_STORE: "redis"`), dan alamat koneksi Redis ditentukan pada variabel `REDIS: "redis://cache:6379"`. Selain itu, disediakan pula variabel SECRET untuk keamanan internal Directus.

Pada fungsi "ports: 8123:8055" menjelaskan bahwa nilai 8123 menjadi port di komputer yang dapat diakses di browser dan nilai 8055 menjadi port *default* yang digunakan Directus pada *container* docker. Untuk fungsi `"ADMIN_EMAIL: "**********@example.com""` menjelaskan bahwa kode tersebut digunakan untuk mendefinisikan email yang digunakan oleh administrator (Super Admin) dan untuk fungsi `"ADMIN_PASSWORD: "**********"` yaitu mengatur *password* yang nantinya akan digunakan untuk masuk ke Directus menggunakan `"ADMIN_EMAIL"` yang sebelumnya sudah didefinisikan.



Gambar 3.7 Hubungan Antar Komponen Environment

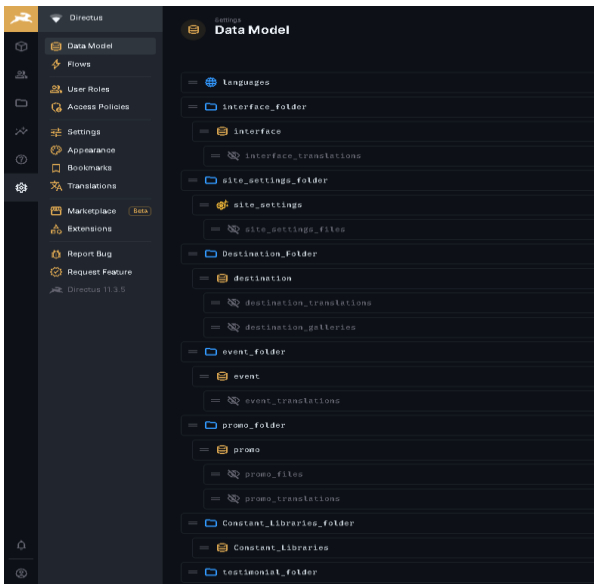
Pada Gambar 3.7 diatas menunjukkan alur kerja dan hubungan antar komponen dalam *environment* yang digunakan untuk menjalankan Directus. Komponen pertama yaitu pengguna yang mengakses aplikasi melalui browser

menggunakan alamat HTTP pada port 8123, lalu port ini diteruskan ke port 8055 didalam container Directus, yang merupakan port *default* dari Directus.

Directus, sebagai inti sistem *Headless CMS*, di-*deploy* dalam container Docker untuk mengelola konten dan menyediakan API. Sistem ini terhubung ke basis data PostgreSQL yang dilengkapi PostGIS melalui port 5432 untuk menyimpan seluruh data utama, termasuk konten multilingual dan data geospasial. Selain itu, Directus juga terkoneksi ke Redis melalui port 6379 sebagai sistem cache, yang berfungsi mempercepat pengambilan data dan mengurangi beban kerja pada database. Arsitektur ini menunjukkan alur interaksi dari browser ke Directus, lalu ke database dan cache, sesuai kebutuhan proyek.

3.2.2. Kumpulan koleksi basis data

Pada proses pengembangan basis data untuk *website* Radyakartiyasa ini, penulis telah membuat sejumlah koleksi data (*collections*) yang disusun dan dikelola menggunakan Directus. Pada Gambar 3.8 (a) merupakan tampilan halaman koleksi di Directus, yang menunjukkan keseluruhan basis data yang telah dibuat pada Directus. Koleksi-koleksi tersebut disusun di dalam folder agar lebih terorganisir dan mudah dikelola.

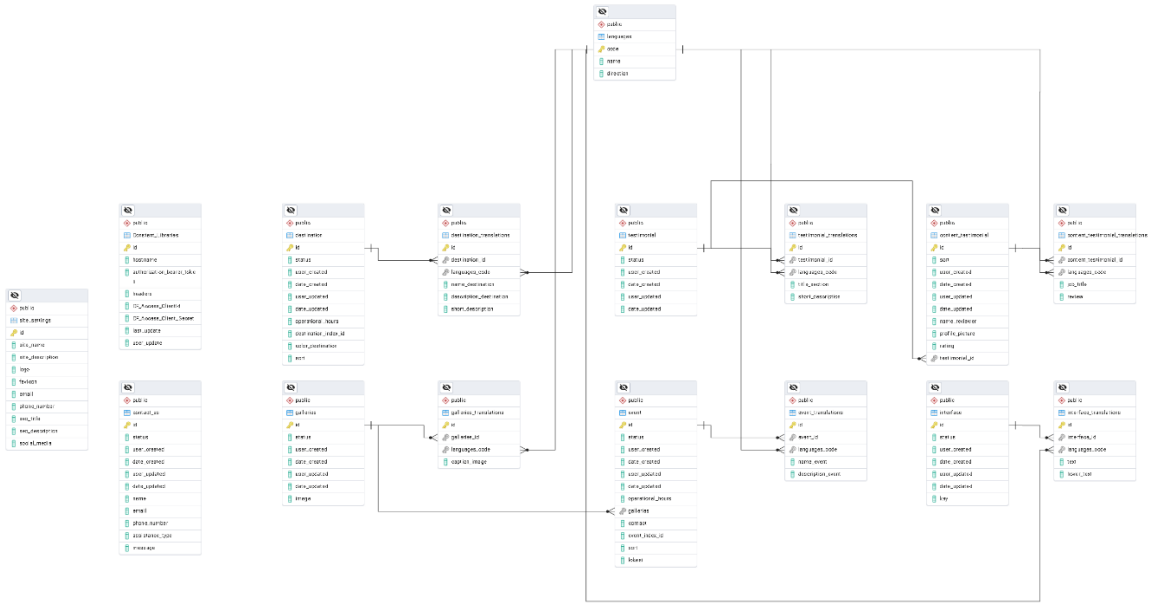


(a)

Gambar 3.8 Tampilan kumpulan koleksi basis data

Selain itu, untuk mendukung fitur *multilingual*, sebagai koleksi juga memiliki sub-koleksi bernama "`NamaKoleksi_translations`" yang berfungsi untuk menyimpan data hasil terjemahan dalam berbagai bahasa yang dipilih. Hal ini memungkinkan konten yang ada pada *website* dapat ditampilkan sesuai dengan bahasa yang dipilih oleh pengguna *website*.

Struktur koleksi ini secara keseluruhan mencerminkan kebutuhan data *website* Radyakartiyasa, serta memberikan fleksibilitas dalam pengelolaan konten multibahasa yang terstruktur dan mudah dikembangkan lebih lanjut.



Gambar 3.9 Tampilan Entity Relationship Diagram (ERD)

Pada Gambar 3.9 ditampilkan *Entity Relationship Diagram* (ERD) yang menggambarkan hubungan antar tabel pada basis data yang digunakan dalam pengembangan website Radyakartiyasa. ERD ini dibuat berdasarkan keseluruhan koleksi data yang telah disusun pada Directus oleh penulis, termasuk tabel utama dan tabel *translations* yang digunakan untuk mendukung fitur multibahasa. Setiap entitas pada diagram dihubungkan melalui relasi *One-to-Many* (O2M) atau *Many-to-One* sesuai dengan struktur data yang dibutuhkan, sehingga memudahkan pengelolaan dan konsistensi data di seluruh sistem.

Dari ERD tersebut dapat terlihat bahwa tabel utama, seperti *language, destination, event, testimonial, content_testimonial, interface*, dan *galleries*, memiliki hubungan langsung dengan tabel *translations* masing-masing.

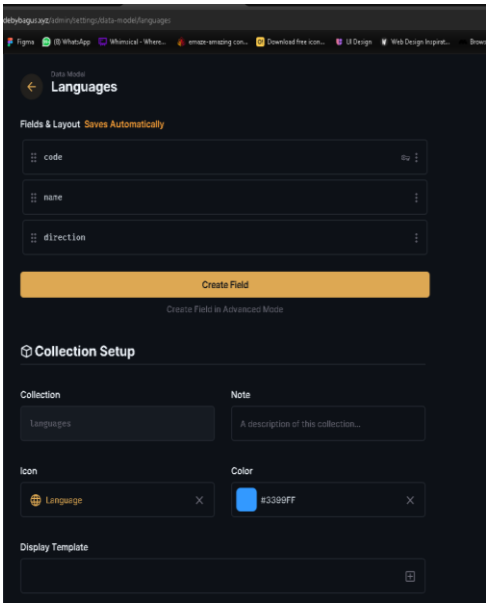
Hubungan ini memungkinkan setiap entitas utama untuk memiliki banyak data terjemahan dalam berbagai bahasa, sementara setiap data terjemahan hanya terhubung ke satu entitas utama.

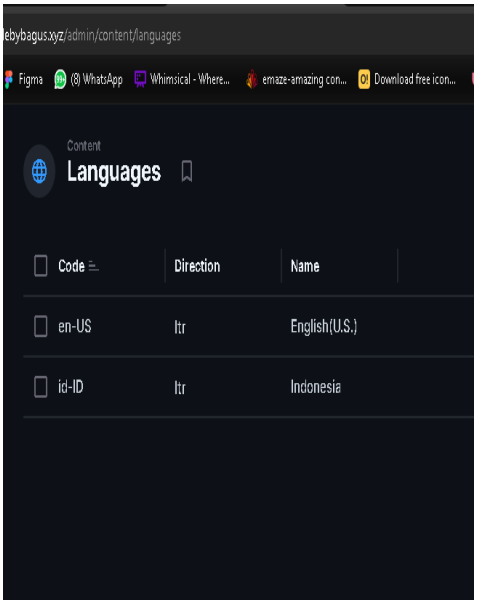
3.2.3. Pembuatan Koleksi Language

Koleksi *language* ini berfungsi sebagai referensi dalam bahasa yang didukung dalam pembuatan basis data *multilingual*, yang kemudian akan direlasikan dengan koleksi lain sehingga konten pada koleksi selanjutnya dapat ditampilkan sesuai dengan bahasa yang terdaftar pada koleksi *Language*. Koleksi *language* merupakan koleksi yang harus dibuat karena akan menjadi koleksi dasar dalam membangun sistem basis data *multilingual*.

Koleksi *Language* ini berisi tiga *field* utama, yaitu:

- *code*: *Field* ini berfungsi untuk menyimpan kode bahasa dan wilayah secara unik mengikuti format internasional seperti `"en-US"` untuk Bahasa Inggris dan `"id-ID"` untuk Bahasa Indonesia.
- *name*: *Field* name ini berfungsi untuk menyimpan nama dari bahasa dan wilayah yang bersangkutan agar nantinya dapat mendeskripsikan *code* yang sebelumnya dibuat kepada pengguna.
- *direction*: *Field* terakhir yang dibuat pada koleksi *Language* ini berfungsi untuk memberikan penjelasan arah penulisan teks seperti, "ltr (*left to right*)" untuk bahasa-bahasa seperti Indonesia dan Inggris, serta "rtl (*right to left*)" untuk bahasa seperti Arab.





(a) (b)

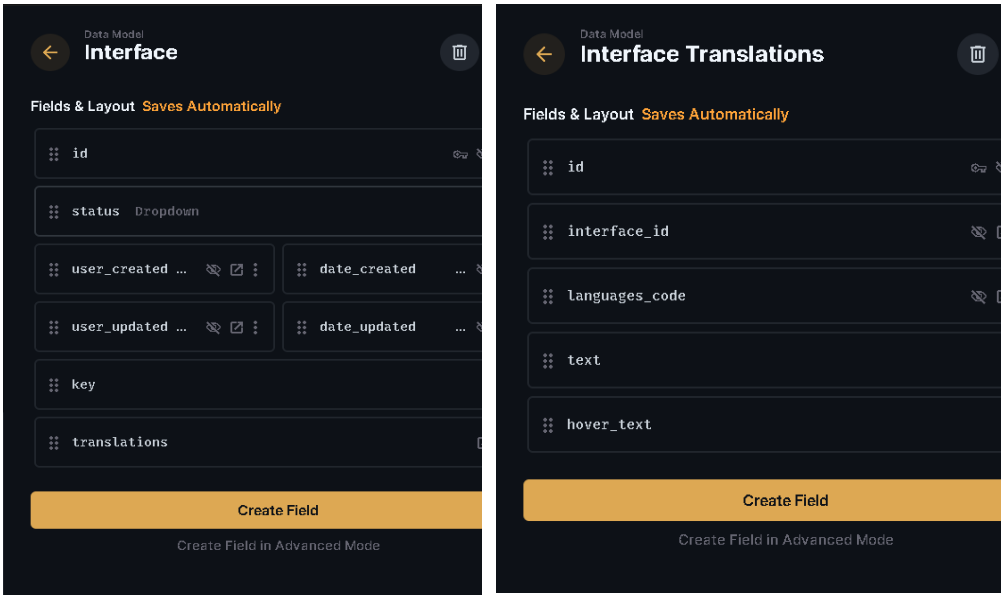
Gambar 3.10 Tampilan Koleksi *Language*

Gambar 3.10 (a) merupakan tampilan *field* pada koleksi *Language* yang sudah dibuat dengan *field* yang tertera seperti *code*, *name*, dan *direction*. Sedangkan Gambar 3.10 (b) menjelaskan tampilan item yan sudah ada pada koleksi *Language*.

3.2.3. Pembuatan Koleksi Interface

Koleksi "interface" merupakan koleksi yang berfungsi untuk membuat item-item yang digunakan di berbagai tampilan *website*, seperti teks pada tombol umum yang sering muncul, misalnya "Lihat Selengkapnya", "Booking", dan "Kunjungi Sekarang". Koleksi ini tentunya sudah direlasikan dengan koleksi *Language* guna mendukung fitur *multilingual*. Pada koleksi "*interface*" terdapat dua *field* utama yaitu "*key*" dan "*translations*" seperti pada Gambar 3.11 (a). Kedua koleksi pada Gambar 3.11 (a) dan Gambar 3.11 (b) saling terhubung, yaitu koleksi "*interface_translations*" otomatis dibuat oleh sistem setelah *field* "*translations*" ditambahkan ke dalam koleksi "*interface*" untuk mendukung konten multibahasa.

Dengan adanya relasi antara koleksi ini dengan koleksi *Language*, maka dari Directus secara otomatis membuat koleksi baru sehingga terdapat dua koleksi untuk koleksi *Interface* ini, yaitu koleksi "*interface*" dan sub koleksi "*interface_translations*". Sedangkan pada sub koleksi "*interface_translations*", terdapat beberapa *field* utama yang akan menggunakan fitur *multilingual*, seperti "*text*" dan "*hover_text*" sesuai dengan yang ada pada Gambar 3.10 (b).



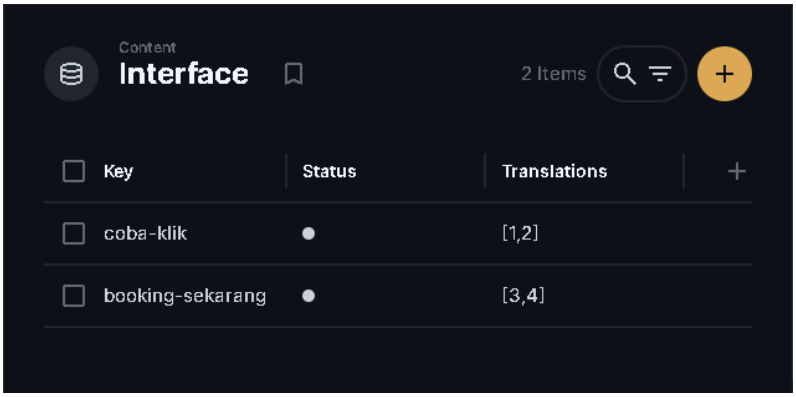
(a) (b)

Gambar 3.11 Tampilan koleksi "*interface*" dan sub koleksi "*interface_translations*"

Berikut merupakan fungsi dari masing-masing *field* utama yang telah dibuat yang ada pada kedua koleksi diatas:

- *key*: *field* ini berfungsi sebagai *identifier* unik untuk setiap item yang dibuat pada koleksi *interface* sehingga nantinya dapat dipanggil oleh *front end* sebagai pananda unik.
- *translations*: *field* ini berfungsi untuk membuat koleksi *multilingual* yang akan direlasikan dengan koleksi *Language*. *Field* ini menunjukkan semua bahasa aktif (dalam contoh ini: Indonesia dan English (U.S.)). Masing-masing berisi *field* terjemahan. Setelah *field* ini dibuat, nanti akan secara otomatis membuat koleksi baru yang bernama *interface_translations*.
- *text*: *field* ini berfungsi untuk memberikan nama item yang ingin dibuat seperti nama tombol. *Field* ini dibuat di koleksi *interface_translations* karena akan ditampilkan secara *multilingual*.
- *hover_text*: *field* ini berfungsi untuk menambahkan efek tambahan pada teks jika diperlukan. *Field* ini dibuat di koleksi *interface_translations* karena akan ditampilkan secara *multilingual*.

Pada Gambar 3.12 (c) dibawah ini merupakan tampilan dari koleksi *Interface* yang sudah berhasil dibuat beserta dengan item yang telah dibuat selama kegiatan magang:



(c)

Gambar 3.12 Tampilan item pada Koleksi *Interface*

3.2.4. Pembuatan Koleksi Site Settings

Koleksi "*site settings*" ini berfungsi sebagai sumber pengaturan untuk *website* Radyakartiyasa yang dapat dengan mudah diatur melalui *back end* Directus tanpa perlu mengubah kode *front end*. Koleksi ini biasanya menyimpan berbagai informasi penting yang bersifat umum, konstan, dan digunakan secara menyeluruh di berbagai bagian *website*. Untuk koleksi ini bersifat *singleton* karena hanya perlu berisikan satu item saja dengan beberapa *field* didalamnya. Pada Gambar 3.13 (a) ditampilkan *field* yang ada pada koleksi Site Settings.



(a)

Gambar 3.13 Tampilan Koleksi *Site Settings*

Berikut merupakan fungsi dari masing masing *field* utama yang telah dibuat yang ada pada koleksi *Site Settings* seperti pada gambar diatas:

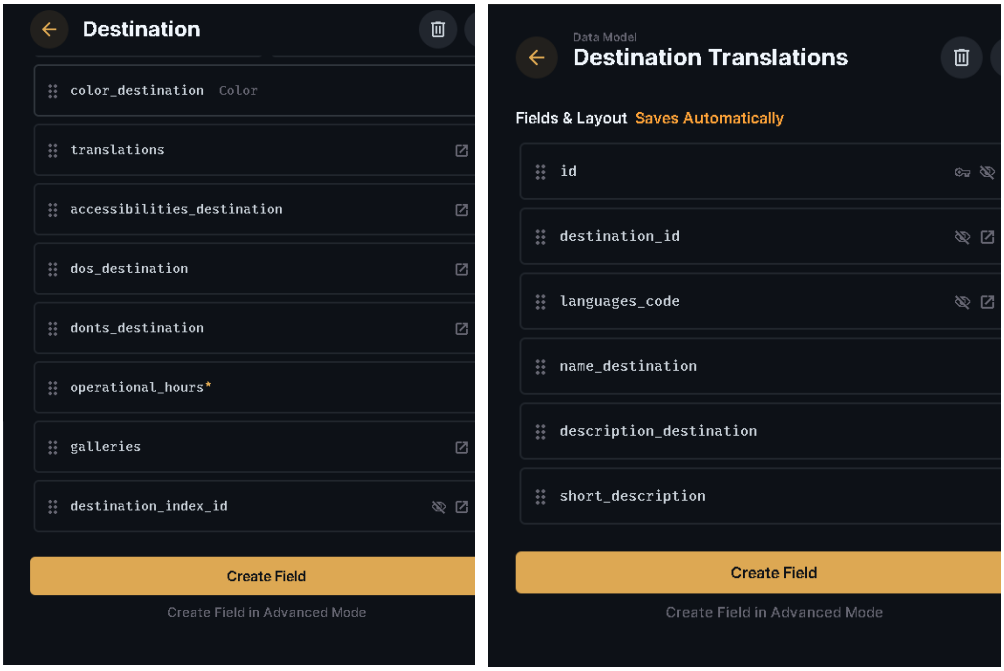
- *site_name*: *field* ini berfungsi untuk menyimpan nama *website* yang dikembangkan.
- *site_description*: *field* ini berfungsi untuk memberikan deskripsi singkat terkait *website* Radyakartiyasa dan berguna untuk *Search Engine Optimization* (SEO) dan tampilan ringkasan saat dibagikan di media sosial.
- logo: *field* ini berfungsi untuk menyimpan file gambar logo utama *website*. Pada pengembangan ini, logo yang digunakan berupa lambang dari Karaton Ngayogyakarta Hadiningrat.
- favicon: *field* ini berfungsi untuk menyimpan gambar favicon yang biasanya menggunakan format .ico atau .png dan biasanya menjadi ikon kecil di tab browser. Pada penelitian ini, gambar favicon yang digunakan sama seperti gambar pada logo.

- *email*: *field* ini berfungsi untuk menyimpan alamat *email* resmi untuk kontak atau komunikasi publik.
- *phone_number*: *field* ini berfungsi untuk menyimpan nomor telepon resmi yang bisa dihubungi oleh pengunjung.
- *seo_title*: *field* ini berfungsi untuk memberikan judul yang ditampilkan di mesin pencari
- *seo_description*: *field* ini berfungsi untuk memberikan deskripsi yang akan muncul dibawah SEO *title* di Google.
- social_media: field ini berfungsi untuk membuat daftar akun media sosial resmi.

3.2.5. Pembuatan Koleksi Destination

Koleksi "*destination*" merupakan koleksi yang berfungsi untuk membuat daftar destinasi wisata Karaton Yogyakarta. Koleksi ini tentunya sudah direlasikan dengan koleksi *Language* guna mendukung fitur *multilingual*. Pada koleksi "*destination*" terdapat beberapa *field* utama yaitu "*color_destination", "translations", "operational_hours*", dan lainnya seperti pada Gambar 3.14 (a). Kedua koleksi pada Gambar 3.14 (a) dan Gambar 3.14 (b) saling terhubung, di mana koleksi "*destination_translations*" otomatis dibuat oleh sistem setelah *field* "*translations*" ditambahkan ke dalam koleksi "*destination*" untuk mendukung konten multibahasa.

Dengan adanya relasi antara koleksi ini dengan koleksi *Language*, maka dari Directus secara otomatis membuat koleksi baru sehingga terdapat dua koleksi untuk koleksi *Destination* ini, yaitu koleksi "*destination*" dan sub koleksi "*destination_translations*". Sedangkan pada sub koleksi "*destination_translations*", terdapat beberapa *field* utama yang akan menggunakan fitur *multilingual*, seperti "*name_destinantion*", "*description_destination*", dan "*short_description*" sesuai dengan yang ada pada Gambar 3.14 (b).



(a) (b)

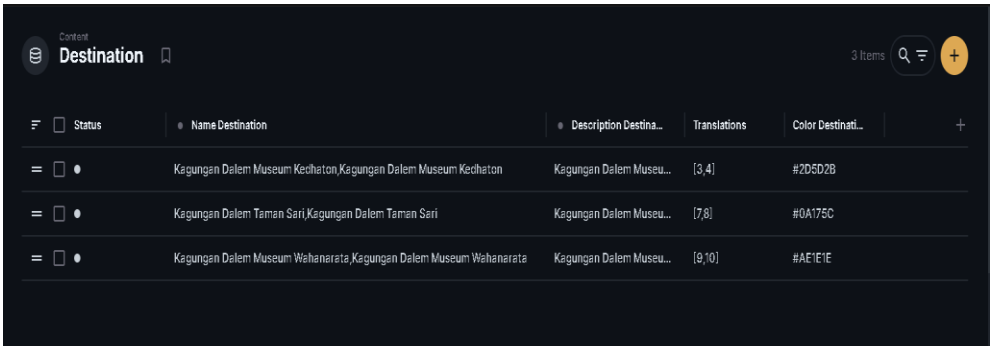
Gambar 3.14 Tampilan koleksi "*destination*" dan sub koleksi "*destination_translations*"

Berikut merupakan fungsi dari masing-masing *field* utama yang telah dibuat yang ada pada kedua koleksi diatas:

- *color_destination*: *field* ini berfungsi untuk menentukan warna utama (*theme color*) yang digunakan untuk tampilan visual dari destinasi tertentu.
- *translations*: *field* ini berfungsi untuk membuat koleksi *multilingual* yang akan direlasikan dengan koleksi *Language*. *Field* ini menunjukkan semua bahasa aktif (dalam contoh ini: Indonesia dan English (U.S.)). Masing-masing berisi *field* terjemahan. Setelah *field* ini dibuat, nanti akan secara otomatis membuat koleksi baru yang bernama *destination_translations*.
- *accessibilities_destination*: *field* ini berfungsi untuk membuat daftar fitur aksesibilitas yang tersedia di destinasi tersebut (misalnya: ramah tunanetra, kursi roda, dan lainnya).
- *dos_destination*: *field* ini berfungsi untuk membuat daftar hal yang disarankan untuk dilakukan oleh pengunjung di destinasi tersebut.

- *donts_destination*: *field* ini berfungsi untuk membuat daftar hal yang dilarang atau tidak boleh dilakukan oleh pengunjung selama berada di destinasi tersebut.
- *operational_hours*: *field* ini berfungsi untuk mengatur waktu operasional atau waktu kunjungan pengunjung untuk destinasi tersebut.
- *galleries*: *field* ini berfungsi untuk menyimpa gambar dari setiap destinasi.
- *destinations_index_id*: *field* ini berfungsi sebagai referensi urutan, penanda khusus, atau koneksi ke koleksi *destination_index* yang nantinya digunakan untuk menampilkan beberapa destinasi pilihan serta mengurutkannya untuk ditampilkan di halaman utama *website*.
- *name_destination*: *field* ini berfungsi untuk membuat nama destinasi.
- *description_destination*: *field* ini berfungsi untuk memberikan deskripsi atau penjelasan panjang mengenai destinasi tersebut.
- *short_description*: *field* ini berfungsi untuk membuat deskripsi atau penjelasan singkat dari destinasi dan akan ditampilkan di daftar destinasi atau *preview cards*.

Pada Gambar 3.15 (c) dibawah ini merupakan tampilan dari koleksi *Destination* yang sudah berhasil dibuat beserta dengan item yang telah dibuat selama kegiatan magang:



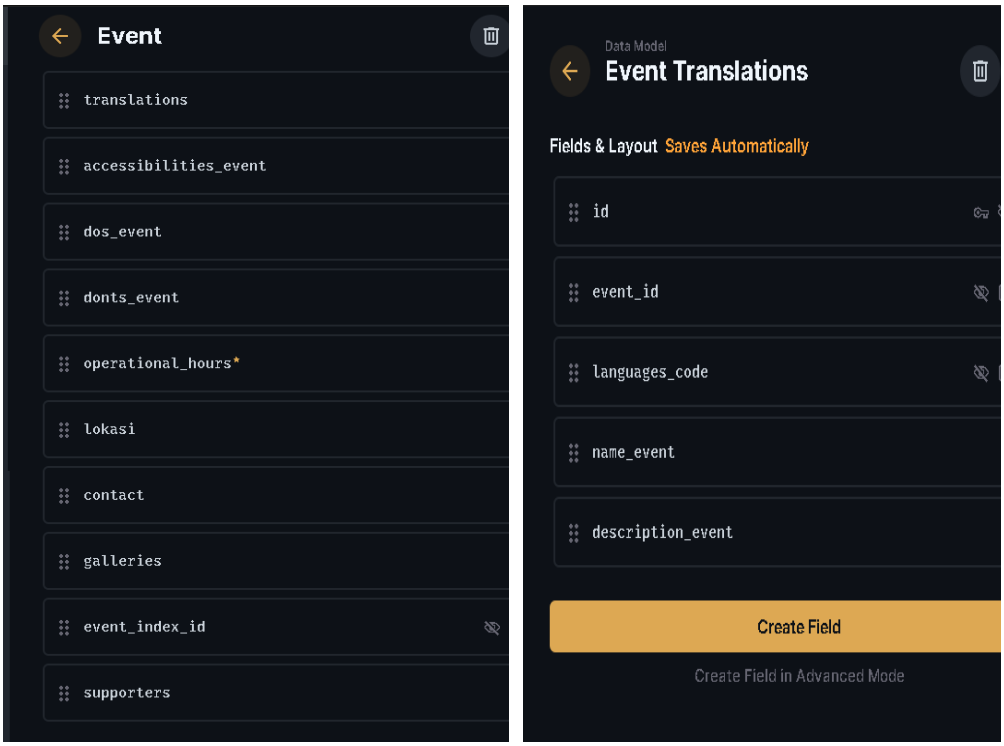
(c)

Gambar 3.15 Tampilan item pada Koleksi *Destination*

3.2.6. Pembuatan Koleksi Event

Koleksi "*event*" merupakan koleksi yang berfungsi untuk membuat daftar kegiatan atau acara akan yang dilaksanakan oleh Karaton Yogyakarta. Koleksi ini tentunya sudah direlasikan dengan koleksi *Language* guna mendukung fitur *multilingual*. Pada koleksi "*event*" terdapat beberapa *field* utama yaitu "lokasi", "*translations*", "*operational_hours*", "*supporters*", dan lainnya seperti pada Gambar 3.16 (a). Kedua koleksi pada Gambar 3.16 (a) dan Gambar 3.16 (b) saling terhubung, di mana koleksi "*event_translations*" otomatis dibuat oleh sistem setelah *field* "*translations*" ditambahkan ke dalam koleksi "*event*" untuk mendukung konten multibahasa.

Dengan adanya relasi antara koleksi ini dengan koleksi *Language*, maka dari Directus secara otomatis membuat koleksi baru sehingga terdapat dua koleksi untuk koleksi *event* ini, yaitu koleksi "*event*" dan sub koleksi "*event_translations*". Sedangkan pada sub koleksi "*event_translations*", terdapat beberapa *field* utama yang pastinya akan menggunakan fitur multilingual, seperti "*name_event*" dan "*description_event*" sesuai dengan yang ada pada Gambar 3.16 (b).



(a) (b)

Gambar 3.16 Tampilan koleksi "*event*" dan sub koleksi "*event_translations*"

Berikut merupakan fungsi dari masing masing *field* utama yang telah dibuat yang ada pada koleksi *event* seperti pada gambar:

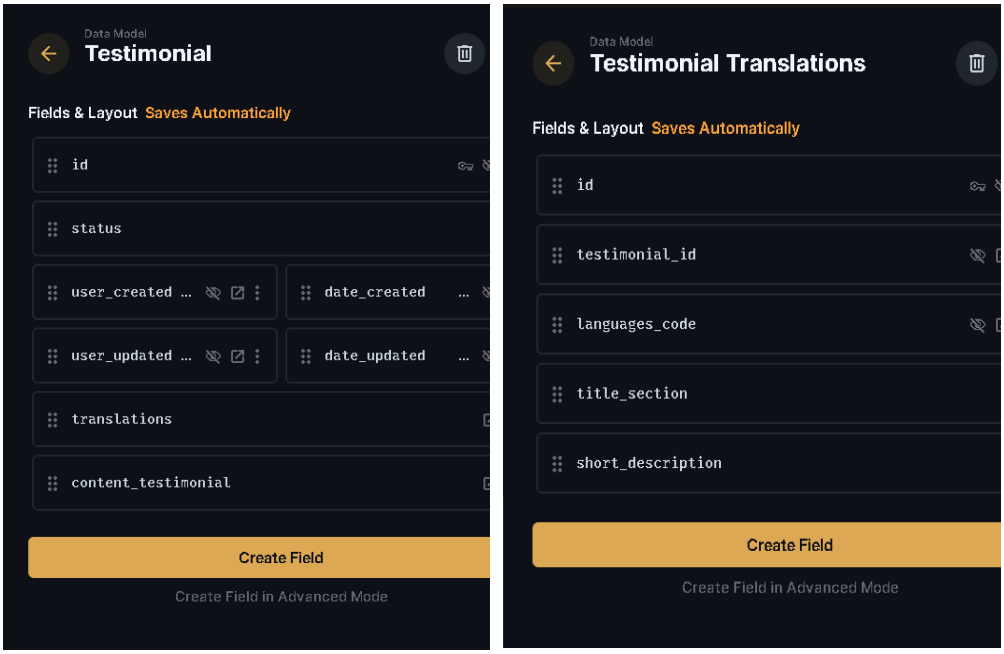
- *translations*: *field* ini berfungsi untuk membuat koleksi *multilingual* yang akan direlasikan dengan koleksi *Language*. *Field* ini menunjukkan semua bahasa aktif (dalam contoh ini: Indonesia dan English (U.S.)). Masing-masing berisi *field* terjemahan. Setelah *field* ini dibuat, nanti akan secara otomatis membuat koleksi baru yang bernama *event_translations*.
- *accessibilities_event*: *field* ini berfungsi untuk membuat daftar fitur aksesibilitas yang tersedia di *event* tersebut (misalnya: ramah tunanetra, akses kursi roda, dan lainnya).
- *dos_event*: *field* ini berfungsi untuk membuat daftar hal yang disarankan untuk dilakukan oleh pengunjung di *event* tersebut.
- *donts_event*: *field* ini berfungsi untuk membuat daftar hal yang dilarang atau tidak boleh dilakukan oleh pengunjung selama berada di *event* tersebut.
- *operational_hours*: *field* ini berfungsi untuk mengatur waktu operasional atau waktu kunjungan pengunjung untuk *event* tersebut.
- lokasi: *field* ini berfungsi untuk menambahkan informasi lokasi pelaksanaan *event* yang diadakan.
- *galleries*: *field* ini berfungsi untuk menyimpan gambar dokumentasi *event* yang diadakan.
- *event_index_id*: *field* ini berfungsi sebagai referensi urutan, penanda khusus, atau koneksi ke koleksi *event_index* yang nantinya digunakan untuk menampilkan beberapa event pilihan serta mengurutkannya untuk ditampilkan di halaman utama *website*.
- *supporters*: *field* ini berfungsi untuk membuat daftar sponsor atau pendukung acara.
- *name_event*: *field* ini berfungsi untuk memberikan nama pada *event* yang akan diadakan.
- *description_event*: *field* ini berfungsi untuk memberikan deskripsi atau penjelasan terkait *event* yang akan diadakan.

3.2.7. Pembuatan Koleksi Testimonial

Pada pembuatan koleksi "*testimonial*" ini, terdapat langkah yang sedikit lebih panjang karena harus membuat dua koleksi yang berbeda untuk nantinya direlasikan. Dua koleksi tersebut yaitu "*testimonial*" dan "*content_testimonial*". Koleksi ini sudah direlasikan dengan koleksi *Language* guna mendukung fitur *multilingual*. Kedua koleksi pada Gambar 3.17 (a) dan Gambar 3.17 (b) saling terhubung, di mana koleksi "*testimonial_translations*" otomatis dibuat oleh sistem setelah *field* "*translations*" ditambahkan ke dalam koleksi "*testimonial*" untuk mendukung konten multibahasa.

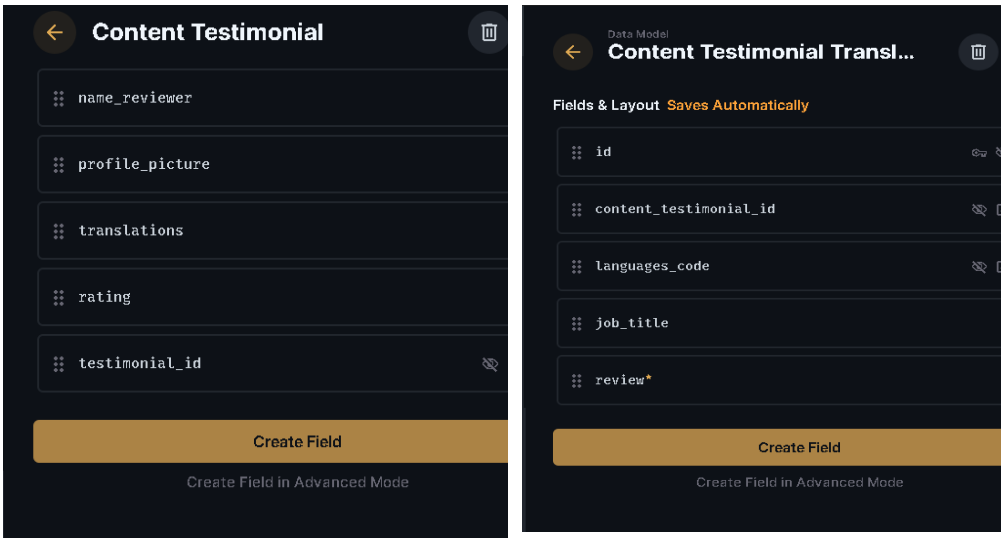
Dengan adanya relasi antara koleksi ini dengan koleksi *Language*, maka dari Directus secara otomatis membuat koleksi baru sehingga terdapat empat koleksi untuk koleksi *testimonial* ini, yaitu koleksi "*testimonial*" dengan sub koleksinya berupa "*testimonial_translations*" dan koleksi "*content_testimonial*" dengan sub koleksinya "*content_testimonial_translations*". Gambar 3.17 (a) merupakan koleksi "*testimonial*" dan Gambar 3.17 (b) merupakan "*testimonial_translations*".

Sedangkan Gambar 3.18 (c) merupakan "*content_testimonial*" dan Gambar 3.18 (d) merupakan "*content_testimonial_translations*". Kedua koleksi pada Gambar 3.18 (c) dan Gambar 3.18 (d) saling terhubung, di mana koleksi "*content_testimonial_translations*" otomatis dibuat oleh sistem setelah *field* "*translations*" ditambahkan ke dalam koleksi "*content_testimonial*" untuk mendukung konten multibahasa.



(a) (b)

Gambar 3.17 Tampilan koleksi "*testimonial*" dan sub koleksi "*testimonial_translations*"



(c) (d)

Gambar 3.18 Tampilan koleksi "*content_testimonial*" dan sub koleksi "*content_testimonial_translations*"

Berikut merupakan fungsi dari masing masing *field* utama yang telah dibuat yang ada pada koleksi *testimonial* seperti pada gambar diatas:

- *translations*: *field* ini berfungsi untuk membuat koleksi *multilingual* yang akan direlasikan dengan koleksi *Language*. *Field* ini menunjukkan semua bahasa aktif (dalam contoh ini: Indonesia dan English (U.S.)). Masing-masing berisi *field* terjemahan. Setelah *field*

ini dibuat, nanti akan secara otomatis membuat koleksi baru yang bernama *testimonial_translations*.

- *content_testimonial*: *field* ini merupakan *field* yang direlasikan *One to Many* kepada koleksi "*content_testimonial*" yang nantinya berisi daftar ulasan dari *reviewer*.
- *title_section*: *field* ini berfungsi untuk memberikan judul langsung pada bagian *testimonial* yang ada di tampilan *front end*, seperti "Ulasan Masyarakat".
- *short_description*: *field* ini berfungsi untuk membuat deskripsi singkat untuk memperkenalkan bagian *testimonial*.
- *name_reviewer*: *field* ini berfungsi untuk mengisi nama orang yang memberikan *testimonial*.
- *profile_picture*: *field* ini berfungsi untuk mengisi gambar profil orang yang memberikan *testimonial*.
- *rating*: *field* ini berfungsi untuk membuat penilaian pada bagian *testimonial* dengan nilai terendah yaitu bintang 1 hingga yang tertinggi yaitu bintang 5.
- *testimonial_id*: *field* ini merupakan *field* hasil relasi ke koleksi "*testimonial*" sebagai induk
- *job_title*: *field* ini berfungsi untuk mengisi jabatan atau peran seorang *reviewer*, misalnya mahasiswa atau wisatawan.
- *review*: *field* ini merupakan *field* utama yang berfungsi untuk mengisi *testimonial* atau ulasan dari *reviewer* yang memberikan testimonial.

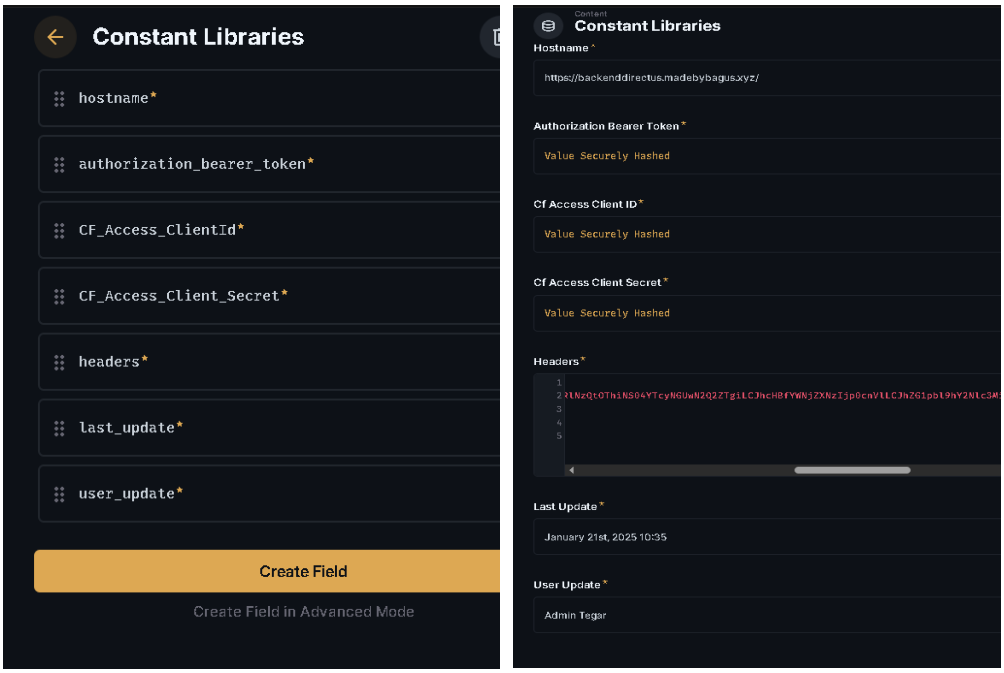
3.2.8. Pembuatan Koleksi Constant Libraries

Koleksi "*constant_libraries*" merupakan sebuah *singleton collection* yang dirancang untuk menyimpan data konfigurasi internal sistem yang bersifat konstan dan tidak berubah-ubah dalam proses pengembangan. Koleksi ini berfungsi sebagai pusat penyimpanan informasi teknis yang dibutuhkan oleh tim pengembang, seperti *hostname* proyek, token akses layanan pihak ketiga (seperti Cloudflare), serta data-data pembaruan seperti waktu dan pengguna terakhir yang melakukan modifikasi data. Cloudflare digunakan dalam proyek

ini sebagai layanan untuk menampilkan *website* ke internet dan membantu agar *website* tetap aman dari serangan yang tidak diinginkan.

Penggunaan tipe *singleton* pada koleksi ini dikarenakan koleksi ini hanya membutuhkan satu entri data saja, tidak memerlukan banyak item seperti koleksi lainnya. Koleksi ini tidak memerlukan fitur *multilingual* karena hanya digunakan oleh pengembang saja sehingga tidak diakses oleh pengguna *website*. Selain itu, tujuan dibuatnya koleksi ini agar semua informasi teknis bisa disimpan di satu tempat, lebih rapi, mudah diatur, dan tidak tercampur dengan data lain seperti konten destinasi ataupun testimoni.

Gambar 3.19 (a) merupakan tampilan dari koleksi "*constant_libraries*" yang berisikan berbagai *field* utama yang akan digunakan untuk menyimpan informasi pengembangan. Sedangkan Gambar 3.19 (b) merupakan tampilan dari *content* yang ada pada koleksi "*constant_libraries*" dengan data pengembangan yang sudah diisi sesuai sesuai dengan kebutuhan masing-masing *field*.



(a) (b)

Gambar 3.19 Tampilan koleksi "*constant_libraries*" dan isi *content* koleksi "*constant_libraries*"

Berikut merupakan fungsi dari masing masing *field* utama yang telah dibuat yang ada pada koleksi *constat_libraries* seperti pada gambar diatas:

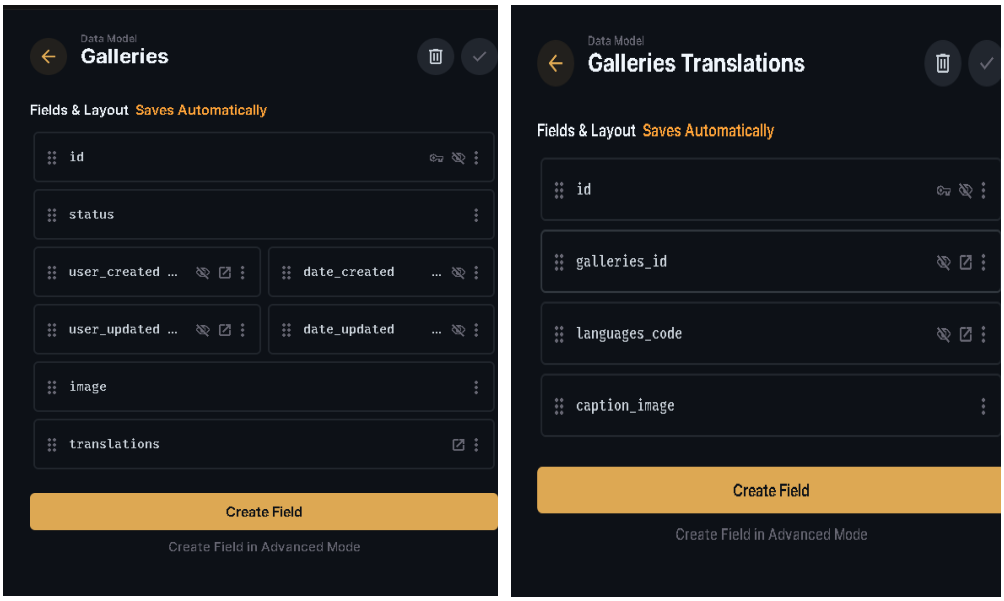
- *hostname*: *field* ini berfungsi untuk menyimpan *hostname* atau *Uniform Resource Locator* (URL) utama *back end* seperti domain Directus yang digunakan untuk pengembangan.
- *authorization_bearer_token*: *field* ini berfungsi untuk menyimpan token otorisasi statis yang digunakan untuk mengakses *endpoint* tertentu (seperti API Directus secara aman).
- *CF_Access_ClientId*: *field* ini berfungsi untuk menyimpan ID klien *Cloudflare Access* yang digunakan untuk otentikasi akses melalui *Cloudflare Zero Trust* dan biasanya dipasangkan dengan *Client Secret* untuk validasi akses ke layanan yang dilindungi Cloudflare seperti Directus ini.
- *CF_Access_Client_Secret*: *field* ini berfungsi sebagai *secret key Cloudflare Access* yang digunakan sebagai pasangan *Client* ID untuk otorisasi akses layanan.
- *headers*: *field* ini berfungsi untuk menyimpan data *header Hypertext Transfer Protocol* (HTTP) tambahan yang diperlukan saat melakukan *request* ke layanan eksternal, contohnya Cloudflare seperti yang digunakan dalam pengembangan.
- *last_update*: *field* ini berfungsi untuk menunjukkan waktu terakhir kali data dalam koleksi ini diperbarui.
- *user_update*: *field* ini berfungsi untuk menyimpan data *user* yang terakhir kali melakukan *update* data pada koleksi ini.

3.2.9. Pembuatan Koleksi Galleries

Koleksi "*galleries*" merupakan koleksi yang berfungsi untuk menyimpan data berupa gambar yang akan ditampilkan di berbagai bagian *website* Radyakartiyasa, seperti dokumentasi acara, destinasi wisata, dan galeri budaya. Koleksi ini tentunya sudah direlasikan dengan koleksi *Language* guna mendukung fitur *multilingual* agar nantinya setiap gambar dapat memiliki *caption* yang disesuaikan dengan bahasa pengguna. Pada koleksi "*galleries*" terdapat beberapa *field* utama yaitu "*image*" dan "*translations*" seperti pada Gambar 3.20 (a). Kedua koleksi pada Gambar 3.20 (a) dan Gambar 3.20 (b) saling terhubung, di mana koleksi "*galleries_translations*" otomatis dibuat oleh

sistem setelah *field* "*translations*" ditambahkan ke dalam koleksi "*galleries*" untuk mendukung konten multibahasa.

Dengan adanya relasi antara koleksi ini dengan koleksi *Language*, maka dari Directus secara otomatis membuat koleksi baru sehingga terdapat dua koleksi untuk koleksi *galleries* ini, yaitu koleksi "*galleries*" dan sub koleksi "*galleries_translations*". Sedangkan pada sub koleksi "*galleries_translations*", terdapat beberapa *field* utama yang pastinya akan menggunakan fitur *multilingual*, seperti "*caption_image*" sesuai dengan yang ada pada Gambar 3.20 (b).



(a) (b)

Gambar 3.20 Tampilan koleksi "*galleries*" dan sub koleksi "*galleries_translations*"

Berikut merupakan fungsi dari masing masing *field* utama yang telah dibuat yang ada pada koleksi *galleries* seperti pada gambar diatas:

- *image*: *field* ini berfungsi untuk menyimpan file gambar yang digunakan pada *website* Radyakartiyasa.
- *translations*: *field* ini berfungsi untuk membuat koleksi *multilingual* yang akan direlasikan dengan koleksi *Language*. *Field* ini menunjukkan semua bahasa aktif (dalam contoh ini: Indonesia dan English (U.S.)). Masing-masing berisi *field* terjemahan. Setelah *field* ini dibuat, nanti akan secara otomatis membuat koleksi baru yang bernama *galleries_translations*.

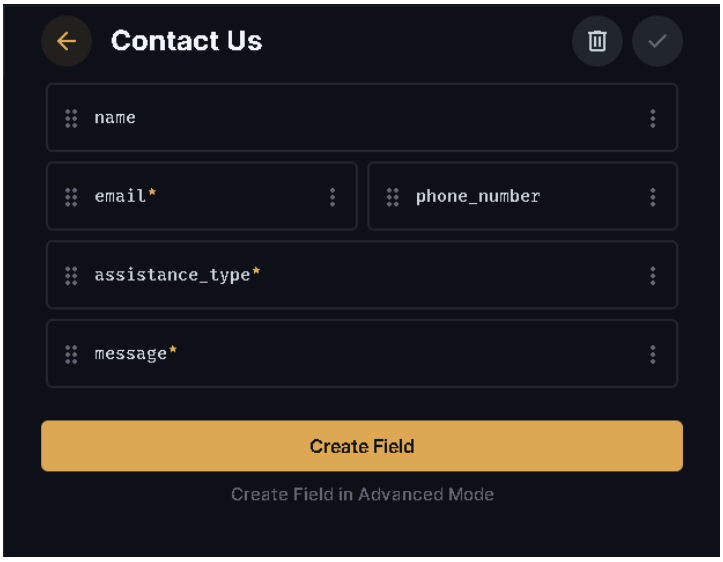
- *caption_image*: *field* ini berfungsi untuk memberikan *caption* atau keterangan pada gambar sesuai dengan bahasa yang dipilih. *Caption* ini nantinya akan ditampilkan di *front end* sesuai dengan bahasa pengguna.

3.2.10. Pembuatan Koleksi Contact-Us

Koleksi "*contact_us*" ini berfungsi untuk menerima serta menyimpan data atau pesan yang dikirim oleh pengguna melalui formulir kontak yang ada pada *website* Radyakartiyasa. Koleksi ini menampung informasi penting seperti nama pengirim, alamat *email*, nomor telepon, jenis bantuan yang dibutuhkan, serta pesan yang ingin disampaikan.

Selain itu tujuan dari adanya koleksi ini yaitu untuk menyediakan kanal komunikasi antara pengunjung dengan pengelola *website*, sehingga pertanyaan, permintaan bantuan, maupun masukan dari pengunjung dapat didokumentasikan dan ditindaklanjuti secara baik melalui sistem *back end* Directus.

Gambar 3.21 (a) merupakan tampilan dari koleksi "*contact_us*" yang berisikan berbagai *field* utama yang akan digunakan untuk menyimpan informasi kebutuhan dari pengunjung yang mengirimkan pesan kepada pengelola.



(a)

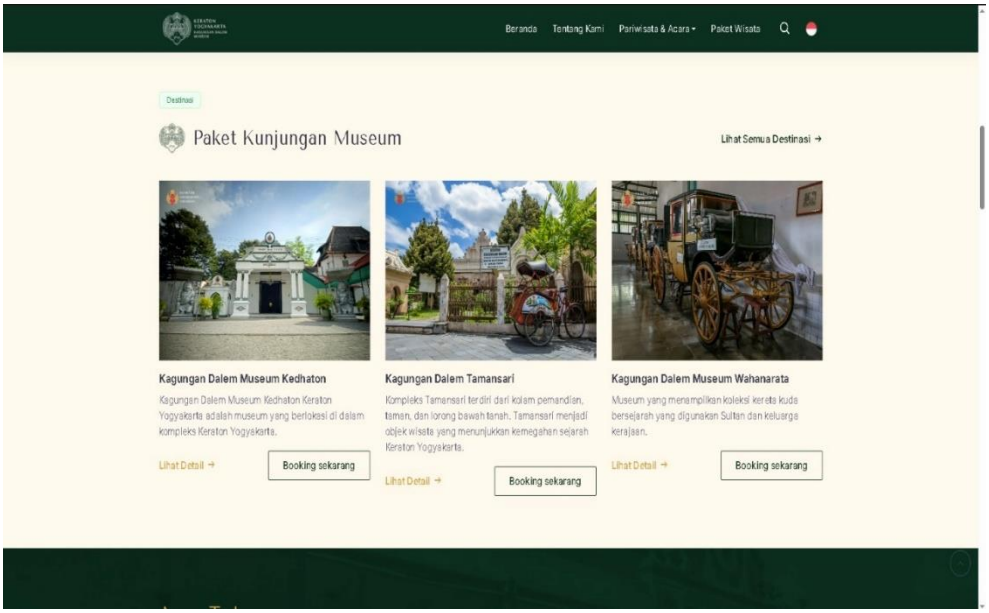
Gambar 3.21 Tampilan koleksi "*contact_us*"

Berikut merupakan fungsi dari masing masing *field* utama yang telah dibuat yang ada pada koleksi *contact_us* seperti pada gambar diatas:

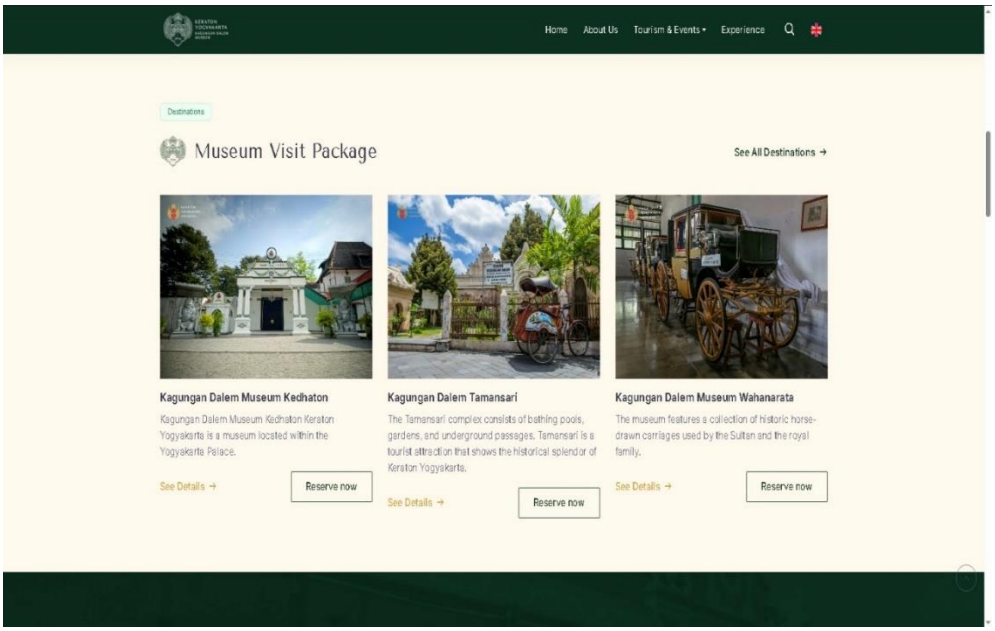
- nama: field ini berfungsi untuk menyimpan nama pengguna yang mengirimkan pesan melalui *form Contact Us*, bisa berupa nama lengkap maupun nama panggilan. *Field* ini membantu pengembang untuk mengetahui siapa yang mengirim pesan.
- *email*: *field* ini berfungsi untuk menyimpan *email* dari pengguna yang mengirimkan pesan melalui *form Contact Us*. *Field* ini wajib diisi, karena nanti akan digunakan untuk membalas pesan yang sebelumnya pengguna kirimkan.
- *phone_number*: *field* ini berfungsi untuk menyimpan nomor telepon pengguna yang mengirimkan pesan. *Field* ini tidak wajib diisi, namun bisa berguna sebagai alternatif kontak selain *email*.
- *assistance_type*: *field* ini berfungsi untuk membuat daftar jenis bantuan yang diminta, seperti: “pertanyaan umum”, “pengaduan”, maupun “saran”. *Field* ini penting untuk mempermudah klasifikasi dan penanganan pesan.
- *message*: *field* ini berfungsi untuk menyimpan isi pesan yang ditulis dan dikirimkan oleh pengguna. *Field* ini juga wajib diisi karena merupakan inti dari komunikasi yang ingin disampaikan oleh pengguna.

3.2.11. Tampilan *Website* Radyakartiyasa

Berikut merupakan tampilan dari *website* yang sudah menggunakan basis data *multilingual*. Pada Gambar 3.22 (a) merupakan tampilan *website* yang menggunakan Bahasa Indonesia. Sedangkan pada Gambar 3.22 (b) merupakan tampilan *website* yang sudah menerapkan fitur *multilingual* dengan bahasa pilihan berupa Bahasa Inggris.



(a)



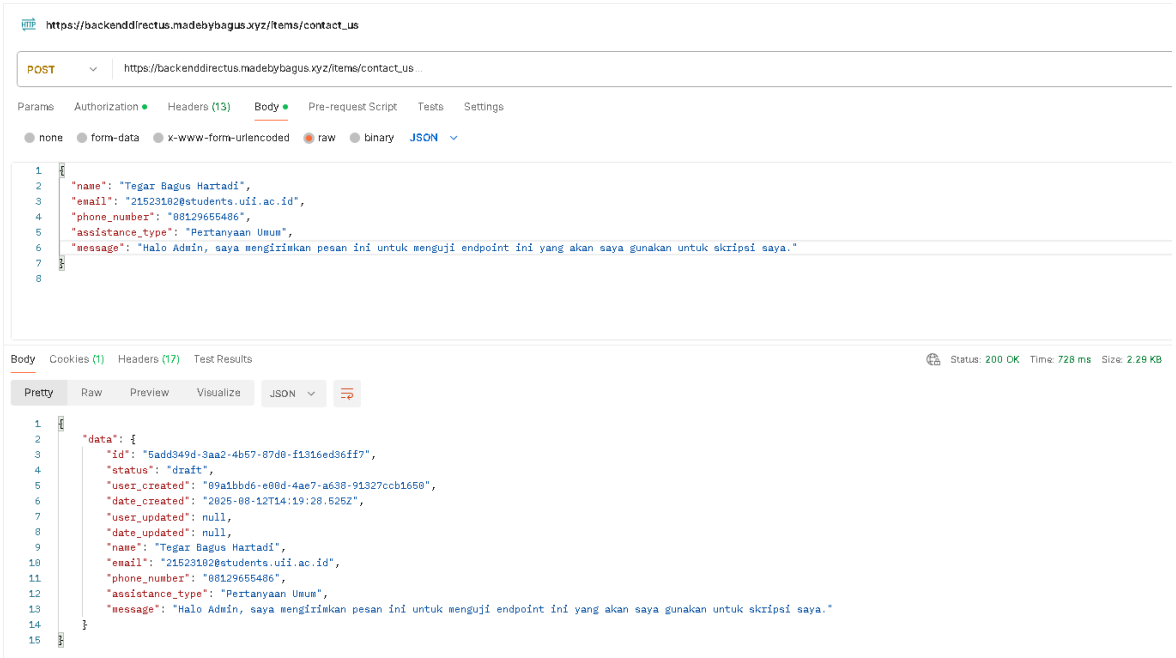
(b)

Gambar 3.22 (a) Tampilan Website Radyakartiyasa (b) Tampilan Website Radyakartiyasa

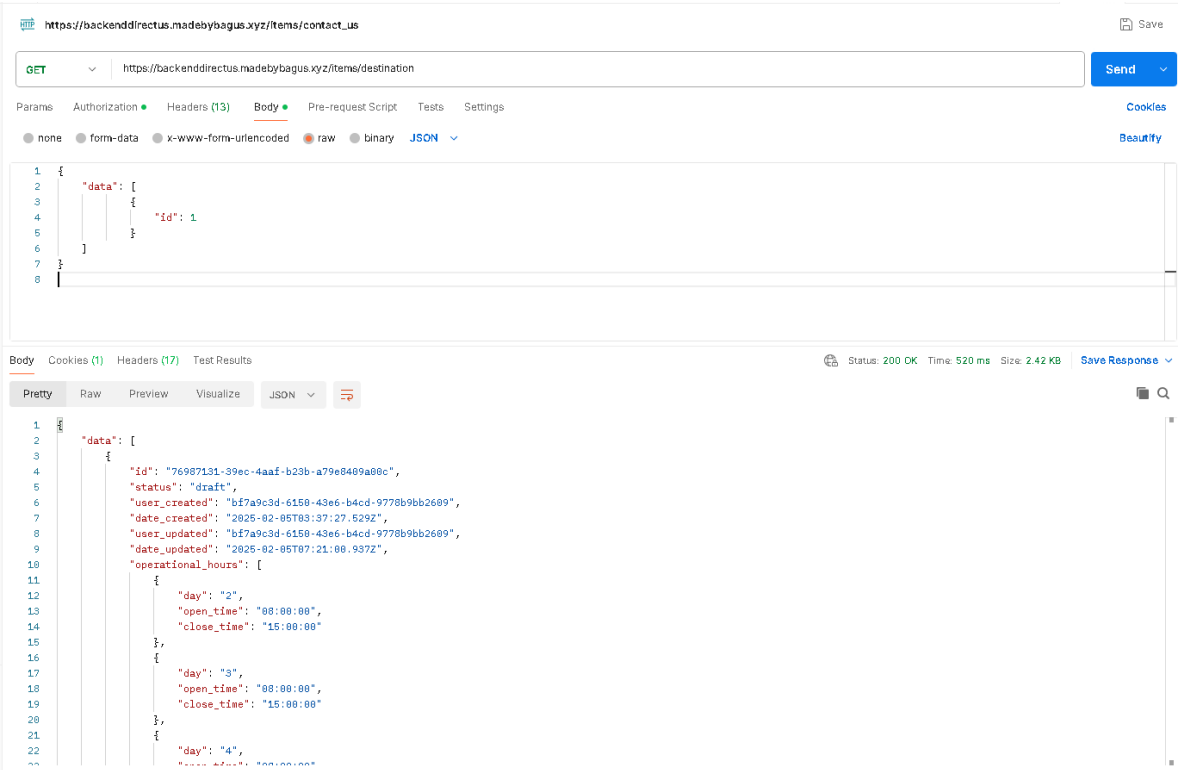
3.2.12. Pengujian Application Programming Interface (API)

Pengujian API dilakukan untuk memastikan bahwa setiap endpoint yang dihasilkan oleh sistem *back end* Directus dapat berfungsi sesuai dengan kebutuhan proyek. Proses pengujian ini menggunaka Postman, yaitu *tools* yang

umum digunakan untuk menguji dan memverifikasi respon API, baik pada metode `GET`, `POST`, `PATCH`, maupun `DELETE`.



Gambar 3.23 Pengujian API *endpoint* “contact_us”



Gambar 3.24 Pengujian API *endpoint* “destination”

Pada gambar 3.23, 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* berformat JSON yang berisi informasi nama, email, nomor telepon, jenis bantuan, dan pesan. Contoh data yang dikirim adalah:

```
{

  "name": "Tegar Bagus Hartadi",
  "email": "21523102@student.uii.ac.id",
  "phone_number": "081295654687",
  "assistance_type": "Pertanyaan Umum",
  "message": "Halo Admin, saya mengirimkan pesan ini untuk  menguji endpoint ini yang akan saya gunakan untuk skripsi saya."

}
```

Gambar 3.25 Data JSON pada Raw Body “contact_us”

Hasil pengujian pada Gambar 3.23 menunjukkan bahwa sistem mengembalikan *status code* **200 OK** dan menampilkan data yang sama seperti yang dikirimkan, yang menandakan proses penyimpanan ke basis data berhasil dilakukan.

Sementara itu, pada Gambar 3.24 pengujian API pada koleksi “destination” dilakukan dengan metode GET menggunakan *endpoint* “https://backenddirectus.madebybagus.xyz/items/destination”. Pada pengujian ini, *raw body* JSON yang dikirim adalah:

```
{

  "data": [
    {
      "id": 1
    }
  ]

}
```

Gambar 3.26 Data JSON pada Raw Body “destination”

Hasil pengujian Gambar 3.24 menunjukkan bahwa sistem mengembalikan *status code* **200 OK** dengan respon berisi detail data destinasi seperti id, status, operational_hours, date_created, dan atribut lain. Hal ini membuktikan bahwa data dapat diambil dengan benar sesuai ID yang diminta pada *request body*.

Dengan adanya pengujian pada kedua *endpoint* tersebut, dapat disimpulkan bahwa API yang dibangun melalui Directus mampu berfungsi dengan baik, baik dalam menambahkan data baru (POST) maupun mengambil data yang sudah ada (GET). Hasil ini menunjukkan bahwa sistem API berjalan

dengan lancar serta siap digunakan oleh tim *front end* untuk menampilkan konten sesuai dengan kebutuhan pengguna.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1. Relevansi Akademik

Berdasarkan pengalaman yang telah didapatkan selama melakukan pengembangan aplikasi secara bersama-sama pada masa perkuliahan, terdapat perbedaan yang penulis rasakan dengan proses pengembangan pada *website* Radyakartiyasa. Perbedaan yang dirasakan terdapat pada proses pengembangan *website* selama perkuliahan terdapat pada pola kerja dan manajemen proyek. Berikut merupakan poin perbedaan antara pengalaman selama perkuliahan dan pengalaman yang penulis dapatkan selama kegiatan magang:

1. **Metode kerja dan manajemen proyek**
    a. Selama kuliah, proses pengembangan proyek dilakukan secara tidak terstruktur tanpa adanya pembagian tahapan pengembangan yang jelas, tanpa jadwal kerja, dan tanpa koordinasi yang rutin antar anggota tim yang terlibat.
    b. Di sisi lain, selama mengikuti kegiatan magang, pengembangan yang dilakukan oleh penulis dan tim dilakukan dengan cara yang profesional menggunakan metode *Agile Scrum* dengan pembagian tahapan kerja per *sprint*, evaluasi rutin (*daily scrum, sprint review, retrospective*), serta dokumentasi *backlog* yang membuat proses lebih terarah dan terukur sesuai kebutuhan. Dalam perancangan koleksi, tahapan yang dilakukan tidak hanya sebatas menerjemahkan instruksi (translator) atau mencatat perintah dari supervisor. Penulis juga terlibat dalam menganalisis kebutuhan data, merancang struktur tabel dan relasi antar koleksi, menyesuaikan rancangan agar mendukung fitur multilingual, serta melakukan validasi bersama tim untuk memastikan rancangan konsisten dan sesuai dengan tujuan proyek.
2. **Penggunaan sistem operasi dan lingkungan pengembangan**
    a. Pada awal perkuliahan, penulis sempat dikenalkan langsung dengan sistem operasi *Linux* dalam bentuk materi pengantar dan latihan dasar. Namun, dalam praktik pengembangan proyek-proyek perkuliahan, sistem operasi yang digunakan tetap berbasis *Windows*, sehingga pemahaman terhadap *Linux* tidak berkembang secara mendalam.

b. Sebaliknya, selama kegiatan magang, penulis mendapatkan kesempatan langsung untuk menggunakan *Linux* dalam proses *instalasi*, konfigurasi, dan *deployment* apliksi menggunakan Docker. Walaupun pada akhirnya dalam proses pengembangan sehari-hari, sistem operasi yang digunakan tetap berbasis *Windows* dan tidak menerapkan *Linux* sepenuhnya. Hal ini disebabkan karenam mempertimbangkan kemudahan akses bagi seluruh anggota tim khususnya anggota tim magang, yang sebagian besar telah terbiasa bekerja di sistem operasi tersebut sehingga dapat mempercepat proses adaptasi dan koordinasi selama pengembangan berlangsung. Pengetahuan dasar mengenai *Linux* yang diperoleh dapat menjadi bekal penting yang berguna untuk pemahanan infrastruktur server dimasa mendatang.

3. Teknologi basis data

a. Selama perkuliahan, penulis hanya dikenalkan dengan pengelolaan basis data secara konvensional menggunakan MySQL sebagai sistem manajemen basis data. Umumnya, pembuatan dan pengelolaan basis data MySQL dilakukan melalui phpMyAdmin yang diakses menggunakan paket server lokal seperti XAMPP.

b. Namun, saat mengikuti kegiatan magang, penulis diperkenalkan dengan pendekatan yang lebih modern dalam pengelolaan basis data menggunakan Directus, sebuah *Headless* CMS berbasis *low-code*. Directus memungkinkan pengguna untuk membuat struktur basis data dan mengelola kontennya secara visual, melalui antarmuka yang intuitif tanpa harus menulis *query* secara manual.

4. Peran dan fokus pekerjaan

a. Selama magang, penulis lebih banyak berperan sebagai *UI/UX Designer* yang hanya berfokus pada desain antarmuka dan pengalaman pengguna.

b. Dalam magang, penulis mendapatkan peran baru yang sama sekali belum pernah penulis perankan, yaitu menjadi *Back end Developer* yang bertanggung jawab kepada pembuatan basis data, pengujian API, serta melakukan *deployment*. Dari peran baru yang penulis dapatkan tersebut memberikan manfaat kepada penulis untuk memperluas kemampuan dari sisi teknis dan pengelolaan sistem informasi.

Secara keseluruhan, pengalaman magang yang penulis jalani telah memberikan pemahaman dan keterampilan baru yang belum pernah diperoleh selama masa perkuliahan. Perbedaan metode kerja, lingkungan pengembangan, tools yang digunakan, serta peran yang dijalani memberikan pembelajaran yang sangat berharga dan memperkaya perspektif penulis dalam menghadapi dunia kerja yang lebih profesional. Hal ini menjadi bekal penting bagi penulis untuk terus berkembang sebagai pengembang sistem yang tidak hanya memahami teori, tetapi juga mampu menerapkannya dalam lingkungan kerja nyata yang dinamis dan menantang.

4.2. Pembelajaran Magang

1. Manfaat

Manfaat yang dapat dirasakan oleh penulis selama melaksanakan kegiatan magang yaitu:

a. Manfaat penerapan *Agile Scrum* dalam proses pengembangan *website* khususnya pada pengembangan basis data multilingual untuk *website* Radyakartiyasa ini. Proses-proses *scrum* yang dilalui memberikan pengalaman baru kepada penulis tentang bagaimana melakukan pengembangan yang lebih terstruktur dan efektif.

   Proses-proses yang mengajarkan penulis untuk melakukan pengembangan dengan lebih terstruktur dimulai dari *sprint planning* dimana penulis melakukan penyusunan rencana tugas pengembangan basis data yang akan dilakukan selama periode *sprint* berlangsung. Selanjutnya penulis melalui tahapan *sprint* yang bertujuan untuk melakukan proses pengembangan sesuai dengan tugas-tugas yang sebelumnya sudah disusun. Selanjutnya yaitu tahapan *sprint review* dimana hasil dari tugas-tugas yang telah penulis kerjakan dilakukan pemeriksaan atau *review* oleh *supervisor* apakah tugas yang telah dilakukan itu sudah sesuai dengan yang diperlukan.

   Penulis juga melalui tahapan *sprint retrospective* yang rutin dilakukan disetiap masa akhir sprint untuk melihat kinerja tim, menidentifikasi setiap tugas yang berjalan dengan baik, menentukan tugas-tugas yang memerlukan perbaikan, serta merancang strategi selanjutnya guna meningkatkan pencapaian tim pada sprint pada masa mendatang. Tugas-

tugas yang telah penulis kerjakan dan memerlukan perbaikan yang sebelumnya ditemukan pada tahapan *sprint retrospective* maka akan masuk ke tahapan *product backlog* untuk dilakukan perbaikan serta penyempurnaan agar mendapatkan hasil yang lebih maksimal.

Dengan penerapan *Scrum* dalam pengembangan, hal ini memberikan penulis pembelajaran baru tentang manajemen waktu yang baik selama proses magang. Penulis belajar untuk membagi waktu antara pengerjaan tugas, koordinasi tim, dan pelaporan kemajuan secara rutin.

b. Selain manfaat dari penggunaan Scrum, penulis juga mendapatkan manfaat dari sisi teknis, memperoleh pemahaman yang lebih mendalam mengenai pembuatan dan pengelolaan struktur basis data modern menggunakan Directus, serta keterampilan teknis dalam merancang database *multilingual* yang mendukung kebutuhan konten multibahasa pada proyek *website* ini.

Selain itu, penulis juga mendapatkan pengalaman baru tentang bagaimana melakukan instalasi sistem operasi seperti Linux. Pemagang dapat memahami bagaimana menggunakan *Linux* dalam proses *instalasi*, konfigurasi, dan *deployment* Directus.

2. Kendala, Hambatan, Tantangan

Kendala, hambatan, dan tantangan yang penulis rasakan selama magang yaitu posisi yang berbeda antara *UI/UX Designer* dengan *Back end Developer.* Hal itu menjadi kendala bagi penulis karena selama berkuliah, penulis sering terlibat dalam proyek dengan posisi sebagai *UI/UX Designer* sehingga kurang mengetahui banyak ilmu pada bidang *Back end Developer*. Namun kenala, hambatan, dan tantangan ini dapat penulis atasi dengan mempelajari ilmu-ilmu yang digunakan di bidang *Back end Developer*.

Kendala, hambatan, dan tantangan selanjutnya yang penulis rasakan adalah penggunaan teknologi baru yang sebelumnya sama sekali belum pernah penulis pelajari di perkuliahan seperti Directus. Hal ini menjadi kendala bagi penulis karena sumber-sumber media belajar yang tersedia masih sangat terbatas. Kendala ini dapat diatasi dengan mempelajari dan mempraktikkan secara baik materi-materi yang disampaikan pada saat *training* dan mempelajari

sendiri materi yang sudah diberikan sewaktu *training* diluar kegiatan *training* serta membaca-baca dokumentasi resmi Directus.

Selanjutnya yang menjadi tantangan terakhir bagi penulis adalah tugas untuk membuat koleksi basis data yang banyak. Hal ini menjadi tantangan bagi penulis karena ada beberapa *milestone* yang didalamnya berisikan beberapa tugas yang harus diselesaikan dengan tenggat waktu yang sudah ditentukan yaitu satu *milestone* harus diselesaikan dalam waktu satu bulan. Tantangan ini dapat penulis hadapi dengan cara mengerjakan tugas dengan baik dan tepat waktu tanpa menunda pengerjaan.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan pelaksanaan kegiatan magang yang telah dilakukan oleh penulis di PT Lumintu Logic, serta pembahasan yang telah diuraikan pada bagian Bab sebelumnya, maka dapat disimpulkan bahwa tujuan dari penulisan tugas akhir ini telah tercapai. Penulis berhasil mengembangkan basis data multilingual pada *website* Radyakartiyasa Karaton Yogyakarta dengan menggunakan Directus sebagai *Headless* CMS dan mengikuti metode pengembangan *Agile Scrum*.

Pengembangan basis data dilakukan melalui beberapa tahapan penting, mulai dari *training*, pendefinisian proyek, hingga pelaksanaan proyek. Pada tahapan *training*, penulis mengikuti pelatihan untuk mempelajari peran sebagai *Back end Developer* dan mempelajari berbagai *tools* yang akan digunakan, khususnya Directus. Setelah melalui proses penilaian, penulis ditugaskan sebagai *Back end Developer* dan bertanggung jawab atas pengembangan basis data multilingual.

Selama pelaksanaan proyek, penulis berhasil menyelesaikan konfigurasi awal Directus, menyusun struktur file konfigurasi "docker-compose.yaml", serta membuat dan merelasikan berbagai koleksi data seperti *Language, Event, Destination, Supporters*, agar mendukung fitur multibahasa. Proses pengujian API juga dilakukan menggunakan Postman guna memastikan *endpoint* berfungsi dengan baik.

Secara keseluruhan, kegiatan magang ini memberikan pengalaman profesional dan kontribusi signifikan terhadap pengembangan sistem informasi pariwisata berbasis web yang bersifat *multilingual*, serta memberikan *insight* bagi penulis dalam menghadapi tantangan teknologi dan dunia kerja yang sesungguhnya.

5.2. Saran

Meskipun pengembangan basis data multilingual pada *website* Radyakartiyasa telah berhasil diselesaikan dengan baik dan sesuai dengan peran serta tujuan awal, terdapat beberapa aspek yang masih dapat ditingkatkan agar manfaat dari pelaksanaan magang ini dapat tercapai secara lebih optimal.

Dari sisi penguasaan teknologi *back end*, khususnya penggunaan Directus sebagai *Headless* CMS, meskipun penulis telah mempelajari dan mengaplikasikannya

dalam proyek, terdapat keterbatasan pemahaman mendalam terhadap fitur-fitur lanjutan Directus. Hal ini disebabkan oleh keterbatasan waktu pelatihan seta dokumentasi yang kurang lengkap, sehingga proses belajar menjadi kurang optimal. Oleh karena itu, disarankan agar proyek berikutnya diberikan modul pelatihan yang lebih terstruktur dan berjenjang mengenai penggunaan Directus.

DAFTAR PUSTAKA

Alwi, M. N. (2024). *Agile Scrum* . Https://Www.Dicoding.Com/Blog/Pengertian-Scrum-Dan-Contoh-Kasusnya/.

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).

Apriza, Z., & Sutarbi, T. (2025). Tantangan dan Solusi Pengelolaan Basis Data: Dari Keamanan Hingga Optimalisasi Query. *Jurnal Sains Student Research*, *3*(2), 448–454. https://doi.org/10.61722/jssr.v3i2.4331

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

Atallah, N. A., & Mardi. (2024). Penggunaan Metode Agile Scrum Pada Perancangan Sistem Informasi Surat Izin Penelitian di BAKESBANGPOL Lombok Tengah. *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, *2*(3), 371–384. https://doi.org/10.61132/neptunus.v2i3.276

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

Bik, M. F. R., & Asmunin. (2017). Implementasi Docker Untuk Pengelolaan Banyak Aplikasi Web (Studi Kasus : Jurusan Teknik Informatika UNESA). In *Jurnal Manajemen Informatika* (Vol. 7).

Christine, N. E., Priyatna, B., Hananto, A. L., & Hilabi, S. S. (2024). Penerapan Metode Agile Scrum Pada Sistem E-Posyandu Berbasis Web. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 2).

Christoper, Y. (2024). Pengaruh Laman Resmi Kementerian Terhadap Reputasi Studi Asosiatif Program Siinas Pada Kementerian Perindustrian. *Jurnal Ilmu Komunikasi*, *3*(2).

Costa, K. I. R. (2022). *Pengembangan dan Pembuatan Website: Sebuah Tinjauan Literatur*.

Haryanto, B., Ardiansyah, A., & Kurniasih, M. (2024). Pengenalan Database Nosql Dan Perbandingannya Dengan Database Relasional. *Jurnal IPSIKOM*, *12*(1), 2024.

Isron, M. H., & Putra, R. E. (2021). Implementasi Virtual Server Berbasis Container Pada Sistem Informasi Geografis Cagar Budaya Mojokerto. *Journal of Informatics and Computer Science*, *03*.

Ningsih, N., Muna, N., Nadziroh, F., Rianrachmatullah, E. F., & Afif, I. A. (2024). *Implementasi Microservices menggunakan REST API untuk UMKM berbasis Raspberry PI*. http://sistemasi.ftik.unisi.ac.id

Nurlailah, E., & Wardani, K. R. N. (2023). Perancangan Website Sebagai Media Informasi Dan Promosi Oleh-Oleh Khas Kota Pagaralam. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *8*(4), 1175–1185. https://doi.org/10.29100/jipi.v8i4.4006

Pratama, D. R., Irmawati, B., & Robbani, R. (2023). *Pengembangan Rest Api Siabang (Sistem Administrasi Pembangunan) Menggunakan Java*. http://begawe.unram.ac.id/index.php/JBTI/

Riana, E. (2021). Konsep Penerapan Metode Scrum dan RDC System Dalam Pengembangan System Mobile Taking Order Web. *Jurnal Media Informatika Budidarma*, *5*(1), 297. https://doi.org/10.30865/mib.v5i1.2688

Rizky, M., & Sugiarti, Y. (2022). Pengunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review. *Journal of Computer Science and Engineering (JCSE)*, *3*(1), 41–48. https://doi.org/10.36596/jcse.v3i1.353

Sarmanela, B. C., Samiya, A. N., & Yuliana, M. E. (2023). *Mengenal Backend Developer: Skill, dan Tanggung Jawab Menjadi Backend Developer*.

Sinulingga, S. U., Dewi, A. R., & Lubis, F. R. (2023). *Website Event Sebagai Daya Tarik Pariwisata Kabupaten Karo Menggunakan Fitur Multilanguage*.

LAMPIRAN

Lampiran A



Lampiran B



Lampiran C


kode
ntar Coding
sama Nikikode
+62 858-4262-0100
Our Vision
and Missio
POISON BREAK SERVICE
TALK
LESS
READ
MORE
MAKE
OFF THE WALL