

RANCANG BANGUN BACKEND PADA SISTEM ALTERNATIF PERPUSTAKAAN UNDIKSHA MENGGUNAKAN GOLANG

BACKEND DESIGN OF UNDIKSHA LIBRARY ALTERNATIVE SYSTEM USING GOLANG

I Made Satria Ananda Wibawa^{1*}, Kadek Teguh Dermawan²

Program Studi Sistem Informasi, Universitas Pendidikan Ganesha, Bali, Indonesia^{1,2}

satria.ananda@student.undiksha.ac.id^{1*}, kdermawan@undiksha.ac.id²

ABSTRACT

Digital library information systems require a structured backend architecture to support efficient data management, security, and service integration. INLISLite, as a widely used library system, still has limitations in the flexibility of backend development and Application Programming Interface (API)-based service integration. This study aims to design and implement INLISLite's backend architecture using a Layered Architecture approach based on RESTful API. The research method used is Waterfall with stages of requirements analysis, system design, implementation, testing, and evaluation. The backend was developed using the Go programming language with the Gin framework and MySQL database. The backend architecture consists of router, middleware, handler, service, and repository layers to separate system responsibilities. Testing was conducted using black-box testing to verify the functionality of the API endpoint. The results of the study indicate that the developed backend architecture is able to increase modularity, ease of testing, and system readiness for further development.

Keywords: INLISLite, Backend System, Layered Architecture, Resourceful API, Digital Library System

ABSTRAK

Sistem informasi perpustakaan digital memerlukan arsitektur backend yang terstruktur untuk mendukung pengelolaan data, keamanan, serta integrasi layanan secara efisien. INLISLite sebagai sistem perpustakaan yang digunakan secara luas masih memiliki keterbatasan dalam fleksibilitas pengembangan backend dan integrasi layanan berbasis Application Programming Interface (API). Penelitian ini bertujuan untuk merancang dan mengimplementasikan arsitektur backend INLISLite menggunakan pendekatan Layered Architecture berbasis RESTful API. Metode penelitian yang digunakan adalah Waterfall dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta evaluasi. Backend dikembangkan menggunakan bahasa pemrograman Go dengan framework Gin dan basis data MySQL. Arsitektur backend terdiri dari lapisan router, middleware, handler, servic, dan repository untuk memisahkan tanggung jawab sistem. Pengujian dilakukan menggunakan black-box testing untuk mmverifikasi fungsionalitas endpoint API. Hasil penelitian menunjukkan bahwa arsitektur backend yang dikembangkan mampu mningkatkan modularitas, kmudahan pengujian, serta kesiapan sistem untuk pengembangan lanjutan.

Kata Kunci: INLISLite, Baccend System, Layered Architectur, Rstful API, Sistem Perpustakaan Digital

PENDAHULUAN

Perpustakaan memiliki peran yang penting dalam lingkungan perguruan tinggi sebagai pusat informasi, sumber referensi akademik, serta sarana pendukung kegiatan pendidikan, penelitian, dan pengabdian kepada masyarakat. Dalam era digital saat ini, teknologi informasi memainkan peran yang penting dalam berbagai aspek kehidupan, termasuk dalam bidang pendidikan, dimana teknologi tidakhanya berfungsi sebagai sarana penyampaian informasi, tetapi juga mampu meningkatkan efektivitas dan efisiensi

layanan pendidikan [1]. Perkembangan teknologi informasi tersebut telah mendorong perubahan besar dalam pengelolaan perpustakaan dari sistem konvensional menjadi perpustakaan digital yang menuntut layanan berbasis daring, akses informasi yang cepat, serta keamanan dan keandalan sistem yang tinggi. Kualitas sistem informasi perpustakaan digital menjadi salah satu indikator keberhasilan transformasi digital di lingkungan pendidikan tinggi.

Sistem Informasi merupakan suatu sistem dalam suatu organisasi yang

mengordinasikan kebutuhan pemrosesan transaksi sehari-hari, mendukung fungsi manajemen dan operasional, serta menyediakan informasi yang diperlukan untuk pengambilan keputusan. Sistem ini terdiri dari komponen-komponen yang saling terintegrasi dalam mengumpulkan, memproses, dan menyimpan data guna mendukung aktivitas organisasi [2]. Dalam pengembangan sistem informasi, backend memegang peranan penting sebagai pusat pengolahan data, pengelolaan logika bisnis, serta penghubung antara basis data dan frontend. Backend yang dirancang dengan baik harus dapat mendukung integrasi layanan, menjaga keamanan data, serta memudahkan proses pemeliharaan dan pengembangan sistem secara berkelanjutan [3].

Pengembangan sistem informasi berbasis Application Programming Interface (API), khususnya RESTful API, telah menjadi pendekatan yang banyak diterapkan dalam perancangan backend sistem informasi modern. Penerapan RESTful API memungkinkan pemisahan yang jelas antara logika bisnis dan antarmuka pengguna, sehingga sistem menjadi lebih modular, mudah dikembangkan, serta mendukung integrasi layanan secara berkelanjutan. [4] menegaskan bahwa keberhasilan implementasi sistem informasi tidak hanya ditentukan oleh ketersediaan fitur, tetapi juga oleh fleksibilitas arsitektur backend dalam mendukung pengembangan dan pemeliharaan sistem. Hal ini menunjukkan bahwa arsitektur backend yang terstruktur menjadi faktor penting dalam meningkatkan efektivitas sistem informasi.

Sejalan dengan pandangan tersebut, penelitian lain yang membahas pengembangan backend sistem informasi berbasis RESTful API menunjukkan bahwa arsitektur backend yang modular mampu meningkatkan keamanan data, efisiensi pengolahan informasi, serta kemudahan pengujian sistem. Studi yang dilakukan oleh [5] membuktikan bahwa penerapan RESTful API dengan mekanisme

otentikasi, otorisasi, dan pemisahan lapisan sistem mampu meminimalkan kesalahan input serta meningkatkan keandalan sistem dalam jangka panjang. Selain itu, penelitian terapan pada sistem layanan digital institusi pendidikan menegaskan bahwa meskipun sistem telah berjalan secara fungsional, kualitas struktur backend tetap berpengaruh signifikan terhadap efektivitas dan keberlanjutan sistem.

INLISLite merupakan sistem yang telah banyak diterapkan di berbagai institusi pendidikan di Indonesia. Meskipun demikian, sistem INLISLite ini masih belum sepenuhnya optimal. Berdasarkan hasil wawancara dengan pustakawan perpustakaan Universitas Pendidikan Ganesha, ditemukan sejumlah kendala dalam penggunaan INLISLite, khususnya pada aspek keamanan sistem dan keterbatasan fleksibilitas pengembangan layanan. Temuan tersebut sejalan dengan hasil penelitian terdahulu yang menunjukkan bahwa INLISLite masih memiliki berbagai keterbatasan, antara lain pada kesesuaian konten informasi dengan kebutuhan pengguna. Penelitian yang dilakukan [6] menunjukkan bahwa konten informasi dalam INLISLite belum sepenuhnya sesuai dengan kebutuhan pengguna, sehingga tingkat kepuasan pemustaka belum optimal.

Selain permasalahan pada sisi pengalaman pengguna, keterbatasan INLISLite juga terlihat pada aspek fleksibilitas pengembangan sistem, khususnya pada arsitektur backend. Backend berperan sebagai inti sistem yang mengelola logika bisnis, manajemen basis data, serta mekanisme autentikasi dan otorisasi pengguna [7]. Sistem backend yang kurang modular menyulitkan proses integrasi layanan baru, pengembangan fitur tambahan, serta penerapan mekanisme pengujian yang sistematis.

Dalam pengembangan sistem dengan kebutuhan yang relatif stabil dan ruang lingkup yang telah terdefinisi dengan jelas, metode Waterfall masih relevan untuk

digunakan. Penelitian yang dilakukan oleh [8] dan [9] membuktikan bahwa penerapan metode Waterfall pada pengembangan sistem berbasis web mampu memberikan alur kerja yang terstruktur, terdokumentasi dengan baik, serta memudahkan proses evaluasi pada setiap tahapan pengembangan. Karakteristik tersebut sesuai dengan kebutuhan pengembangan sistem perpustakaan institusi.

Berdasarkan Permasalahan dan penelitian terdahulu tersebut, Undiksha memerlukan pengembangan sistem perpustakaan digital yang lebih adaptif, aman, dan berkelanjutan melalui perancangan arsitektur backend yang mampu memisahkan tanggung jawab sistem secara jelas dan mendukung integrasi layanan berbasis Application Programming Interface (API).

Oleh karena itu, penelitian ini berfokus pada perancangan dan implementasi arsitektur backend sistem perpustakaan berbasis INLISLite menggunakan pendekatan Layered Architecture dan RESTful API. Pendekatan ini dipilih untuk meningkatkan modularitas sistem, keamanan data, serta kemudahan pengujian dan pemeliharaan kode. Melalui penelitian ini, diharapkan dapat dihasilkan arsitektur backend yang lebih terstruktur dan siap mendukung transformasi digital perpustakaan Undiksha secara berkelanjutan.

METODE PENELITIAN

Penelitian ini menggunakan metode pengembangan perangkat lunak Waterfall, yaitu model pengembangan sistem dalam membangun sistem informasi. Model ini dipilih karena sesuai diterapkan pada pengembangan sistem dengan kebutuhan yang sudah jelas dan terdefinisi sejak awal. Pendekatan ini memungkinkan setiap tahapan pengembangan diselesaikan secara sistematis sebelum dilanjutkan ke tahap berikutnya, sehingga meminimalkan kesalahan pada proses implementasi [10]. Metode waterfall menerapkan proses pengembangan secara bertahap (one by one

phased), dimana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya, sehingga dapat meminimalkan potensi kesalahan pada proses implementasi. Dengan alur yang terstruktur tersebut, proses pengembangan menjadi lebih terarah dan sistematis serta mampu meningkatkan kualitas hasil pengembangan sistem [11]. Tahapan dalam model waterfall mencakup lima langkah utama [12], seperti terlihat pada gambar 1 berikut ini:



Gambar 1. Alur Metode Waterfall

2.1 Analisis kebutuhan

Tahap analisis kebutuhan dilakukan melalui identifikasi dan diskusi awal dengan staf Teknologi Informasi (TI) perpustakaan. Pada tahap ini, dilakukan pembahasan secara mendalam terhadap proses bisnis yang berjalan pada sistem INLISLite, meliputi alur peminjaman koleksi, pengelolaan katalog, pencatatan kunjungan, manajemen keanggotaan, serta mekanisme pelaporan yang digunakan sebelumnya. Selain itu, dilakukan identifikasi permasalahan yang sering dihadapi oleh pustakawan dan pengguna, pemetaan kebutuhan fitur baru yang dinilai lebih relevan, serta evaluasi terhadap komponen sistem yang sudah tidak efektif. Hasil dari tahap ini digunakan sebagai dasar dalam menentukan arah pengembangan sistem, prioritas fitur, serta kebutuhan arsitektur backend agar sistem perpustakaan digital yang dikembangkan mampu berjalan secara optimal dan sesuai dengan kebutuhan operasional.

2.2 Perancangan sistem

Menyusun use case untuk memetakan interaksi antara pengguna dengan sistem serta menggambarkan fungsi-fungsi utama yang harus disediakan oleh backend. Perancangan use case ini sangat penting karena menjadi dasar dalam menentukan kebutuhan fungsional, alur layanan, dan batasan operasional setiap modul backend yang akan dikembangkan.

2.3 Implementasi

Melakukan penyaringan tabel dari database lama yang kemudian disesuaikan dengan modul atau fitur utama yang akan diterapkan pada sistem baru. Selanjutnya mengembangkan sistem backend menggunakan bahasa pemrograman golang dan framework gin. Implementasi difokuskan pada pengembangan Restful API yang mendukung pemisahan logika bisnis dan akses data secara terstruktur sesuai dengan arsitektur yang telah dirancang.

2.4 Pengujian

Menggunakan metode black-box testing, yaitu pengujian sistem berdasarkan kesesuaian antara input dan output tanpa memperhatikan struktur kode sumber. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi backend berjalan sesuai dengan kebutuhan pengguna dan spesifikasi sistem yang telah ditetapkan pada tahap sebelumnya.

2.5 Pemeliharaan

Melakukan penyempurnaan dan perbaikan sistem berdasarkan hasil uji coba dan umpan balik dari pengguna lapangan atau mentor dari Unit Penunjang Akademik (UPA) Teknologi Informasi dan Komunikasi (TIK). Tahap ini bertujuan untuk menjaga kestabilan sistem dan memastikan backend dapat terus dikembangkan sesuai kebutuhan di masa mendatang.

HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan Sistem

Kebutuhan sistem terdiri dari kebutuhan fungsional dan non-fungsional.

Kebutuhan fungsional berkaitan dengan kinerja sistem saat inputan tertentu. Sedangkan kebutuhan non-fungsional merupakan kebutuhan yang diperlukan untuk membantu pembuatan sistem, meliputi kebutuhan perangkat keras dan perangkat lunak. Adapun kebutuhan fungsional dari sistem ini dibagi menjadi dua peran yaitu, admin dan anggota. Berikut merupakan kebutuhan fungsional aplikasi dari sisi admin:

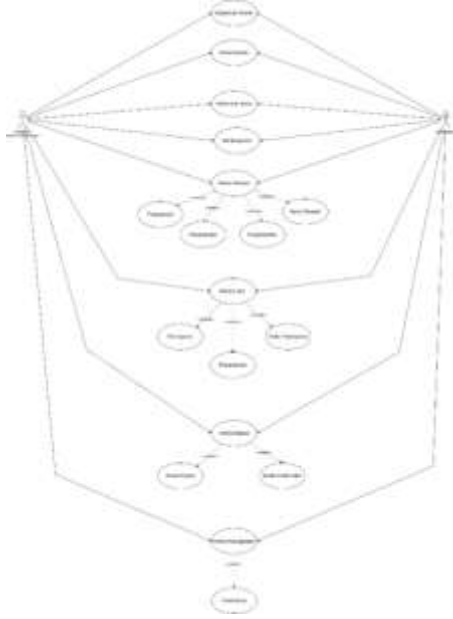
- a. Admin dapat masuk ke dalam sistem melalui proses login.
- b. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait anggota.
- c. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait catalog buku.
- d. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait koleksi buku.
- e. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait sirkulasi.
- f. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait peminjaman dan pengembalian loker.
- g. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait pengelola (admin) lainnya.
- h. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait buku tamu.
- i. Admin dapat melihat, mengubah, menambah, dan menghapus data terkait bacaditempat.
- j. Admin dapat melihat dan mengunduh data laporan.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi anggota:

- a. Anggota dapat masuk ke dalam sistem melalui proses login anggota.
- b. Anggota dapat melihat dan mengunduh kartu identitas perpustakaan.
- c. Anggota dapat melakukan sirkulasi peminjaman, pengembalian buku.

3.2 Perancangan Sistem

Perancangan sistem ini direpresentasikan dengan dua diagram yaitu use case dan ERD (Entity Relational Diagram). Pada diagram use case berfungsi untuk menjelaskan proses kegiatan secara berurutan dalam sistem, mampu menggambarkan proses bisnis, mampu menampilkan kegiatan dalam suatu proses dan sebagai jembatan antara pengembang perangkat lunak dan klien untuk memetakan sebuah sistem. Berikut merupakan diagram use case dari sistem.

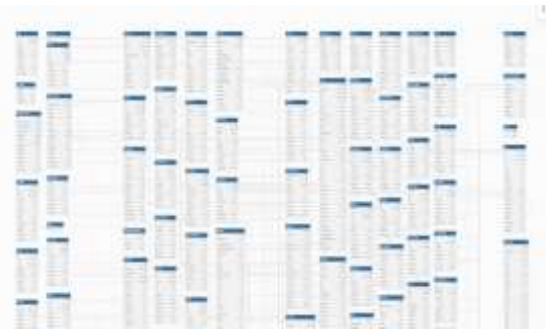


Gambar 2. Diagram use case sistem

Pada gambar 1 diperlihatkan diagram use case dari sistem yang meliputi 2 aktor, yaitu admin dan anggota. Setiap aktor melakukan login ke sistem. Setelah login, setiap aktor melakukan fungsi masing-masing yang dijelaskan sebagai berikut.

Admin memiliki peran untuk mengelola data anggota, catalog, koleksi, sirkulasi, loker, baca ditempat, buku tamu, laporan yang meliputi melihat, mengupdate, membuat, dan menghapus. Selanjutnya Anggota memiliki peran untuk mengunduh kartu, dan melakukan sirkulasi peminjaman serta pengembalian buku.

Perancangan berikutnya adalah rancangan database yang berguna untuk melihat relasi antar variabel pada masing-masing tabel yang akan digunakan. Berikut merupakan rancangan tabel database pada sistem.



Gambar 3. Struktur database

Entity Relationship Diagram (ERD) pada sistem perpustakaan yang dikembangkan mengadopsi struktur basis data INLISLite dengan fokus pada pengelolaan keanggotaan, katalog koleksi, sirkulasi peminjaman, kunjungan perpustakaan, serta aturan dan otorisasi layanan. Perancangan ERD berperan penting dalam proses analisis dan perancangan sistem karena dapat merepresentasikan kebutuhan data secara komprehensif serta menampilkan keterkaitan antarentitas secara jelas [13]. Selain itu, perancangan basis data bertujuan untuk meningkatkan kemudahan dalam proses akses dan pengelolaan data, serta mendukung kecepatan, efisiensi penyimpanan, keakuratan, ketersediaan, kelengkapan, keamanan, dan pemakaian data secara bersamaan [14]. Dengan demikian, pemodelan data yang sistematis melalui ERD dapat menjamin konsistensi dan integritas data serta mendukung efektivitas proses bisnis pada sistem perpustakaan.

Tabel user berperan sebagai entitas akun yang digunakan untuk autentikasi dan pengelolaan sistem. Tabel ini berelasi dengan tabel member, dimana setiap data anggota perpustakaan memiliki keterkaitan dengan akun pengguna. Selanjutnya, tabel member berelasi dengan berbagai tabel master seperti jenis_anggota, status_anggota, jenis_kelamin, masa_berlaku_anggota, master_pekerjaan, master_pendidikan, master_fakultas, master_jurusan, master_program_studi, dan master_jenis_identitas. Relasi ini digunakan untuk mengelompokkan dan mengklasifikasikan data anggota secara

terstruktur serta menghindari redundansi data.

Dalam pengelolaan koleksi, tabel catalog berfungsi sebagai entitas utama yang menyimpan data bibliografi bahan pustaka. Tabel ini berelasi dengan tabel collection, dimana satu data katalog dapat memiliki lebih dari satu koleksi fisik. Selanjutnya, tabel collection berelasi dengan tabel collectionstatus untuk menentukan status ketersediaan koleksi, serta dengan tabel location dan location_library untuk menunjukkan lokasi fisik penyimpanan koleksi di perpustakaan. Tabel catalog juga berelasi dengan tabel collectioncategory, collectionsource, worksheet, dan kata_sandang sebagai data pendukung dalam pengelompokan dan pengolahan katalog.

Proses sirkulasi peminjaman direpresentasikan oleh tabel collectionloan, yang berelasi dengan tabel member sebagai peminjam koleksi. Setiap proses peminjaman dapat memiliki lebih dari satu item koleksi yang dicatat pada tabel collectionloanitem, dimana tabel ini berelasi langsung dengan tabel collection. Untuk mendukung proses perpanjangan masa pinjam, tabel collectionloan berelasi dengan tabel collectionloanextend yang menyimpan riwayat perpanjangan peminjaman berdasarkan transaksi yang terjadi.

Aktivitas kunjungan dan layanan perpustakaan dicatat melalui tabel bacaditempat dan tujuan_kunjungan. Tabel bacaditempat berelasi dengan tabel member dan collection untuk mencatat aktivitas membaca koleksi di tempat. Sementara itu, tabel tujuan_kunjungan berelasi dengan tabel memberguesses dan groupguesses untuk merekam data kunjungan anggota maupun pengunjung secara berkelompok berdasarkan tujuan kunjungan yang dipilih.

Pengelolaan fasilitas perpustakaan direpresentasikan oleh tabel locker yang berelasi dengan tabel member, master_loker, dan master_pelanggaran_locker. Relasi ini digunakan untuk mencatat penggunaan

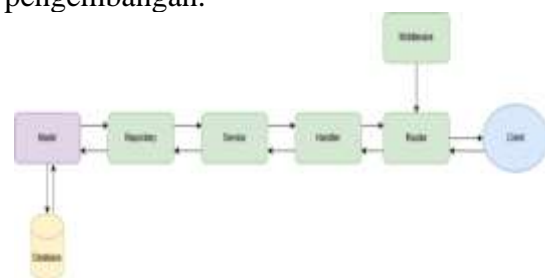
locker oleh anggota serta pelanggaran yang terjadi selama penggunaan fasilitas tersebut. Selain itu, tabel pelanggaran berelasi dengan tabel member dan jenis_pelanggaran untuk mencatat pelanggaran yang dilakukan oleh anggota, sedangkan tabel jenis_denda dan currency digunakan untuk mendukung pencatatan denda yang dikenakan.

Aturan dan kebijakan peminjaman koleksi diatur melalui tabel collectionrule, yang berelasi dengan tabel jenis_anggota, collectioncategory, dan location. Relasi ini memungkinkan sistem menentukan ketentuan peminjaman yang berbeda berdasarkan jenis anggota, kategori koleksi, dan lokasi koleksi. Selain itu, tabel memberloanauthorizecategory dan memberloanauthorizelocation berelasi dengan tabel member untuk memberikan otorisasi khusus kepada anggota tertentu dalam melakukan peminjaman koleksi pada kategori atau lokasi tertentu.

3.3 Implementasi

a. Implementasi Arsitektur

Dalam pembuatan sistem ini, tahap awal yang dilakukan yaitu melakukan instalasi framework Gin agar bisa digunakan pada project Golang ini. Selanjutnya dilakukan penginstalan library-library yang dibutuhkan seperti mysql yang digunakan untuk menghubungkan project dengan database dan mempermudah proses pengembangan.



Gambar 4. Arsitektur sistem

Gambar 4 merupakan arsitektur sistem yang digunakan. Secara umum, rancangan arsitektur backend ini terbagi menjadi enam komponen utama, yaitu: Client, Router, Middleware, Handler, Service, Repository, serta Database sebagai penyimpanan data utama. Proses alur kerja

sistem dimulai dari Client, yang dapat berupa aplikasi web atau aplikasi lain yang mengakses backend melalui protokol HTTP. Client mengirimkan request ke backend yang selanjutnya diterima oleh Router. Router berfungsi sebagai pengatur jalur request, yang memetakan endpoint RESTful API ke handler yang sesuai berdasarkan metode HTTP seperti GET, POST, PUT, dan DELETE.

Sebelum request diproses lebih lanjut, Router akan melewatkan request tersebut ke Middleware. Middleware berperan sebagai lapisan pendukung yang menangani proses lintas fungsi (cross-cutting concerns), seperti autentikasi dan otorisasi (contohnya PrivateRoute atau JWT), validasi awal request, serta pengukuran waktu eksekusi. Dengan adanya middleware, logika pendukung tidak tercampur langsung dengan logika bisnis utama.

Setelah melalui middleware, request diteruskan ke Handler. Handler bertugas sebagai penghubung antara HTTP layer dan logika aplikasi. Pada lapisan ini dilakukan proses parsing request (query parameter, path parameter, dan body JSON), validasi data dasar, serta penyusunan response HTTP. Handler tidak mengandung logika bisnis yang kompleks, melainkan hanya mengatur alur masuk dan keluar data. Selanjutnya, Handler akan memanggil Service layer. Service merupakan inti dari logika bisnis aplikasi INLISLite. Dengan adanya Service layer, aturan bisnis menjadi terpusat dan mudah dikembangkan atau diuji secara terpisah.

Service kemudian berinteraksi dengan Repository layer. Repository berfungsi sebagai abstraksi akses data ke database menggunakan ORM (GORM). Pada lapisan ini terdapat seluruh query database, baik berupa operasi CRUD maupun query kompleks.

b. Implementasi Backend

Desain backend pada sistem ini disusun dengan mengacu pada standar RESTful API, yaitu menggunakan pola

HTTP Method / api / resource, sehingga setiap endpoint merepresentasikan fungsi dan sumber daya (resource) yang jelas. Pendekatan ini bertujuan untuk menjaga konsistensi penamaan, kemudahan pemahaman, serta kemudahan integrasi antara frontend dan backend. Seluruh endpoint yang dikembangkan mencakup modul-modul utama yang dibutuhkan oleh sistem perpustakaan digital sebagai alternatif INLISLite.



Gambar 5 merupakan struktur routes sistem backend. Seluruh endpoint didaftarkan dan dikelola melalui mekanisme routing pada backend dengan pembungkus utama /api sebagai prefiks global. Prefiks ini berfungsi untuk mengelompokkan seluruh layanan API dalam satu namespace yang konsisten, sehingga memudahkan pengelolaan versi API dan pengembangan lanjutan. Setelah prefiks utama, endpoint dikelompokkan kembali berdasarkan modul dan tingkat aksesnya.

Pada implementasinya, sistem membedakan antara public route dan private route. Public route dapat diakses tanpa autentikasi, seperti endpoint untuk proses login. Sementara itu, private route hanya dapat diakses oleh pengguna yang telah terautentikasi menggunakan mekanisme JSON Web Token (JWT). Pembagian ini bertujuan untuk menjaga keamanan sistem serta memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat mengonsumsi data tertentu.

3.4 Pengujian

Sistem yang dikembangkan diuji menggunakan metode *black-box*. Blackbox testing merupakan metode pengujian perangkat lunak yang difokuskan pada

pengujian fungsionalitas sistem tanpa mengetahui struktur intrnal kode program. Pengujian dilakukan brdasarkan input dan output, dengan tujuan memastikan bahwa sistem bekerja sesuai dengan kebutuhan

pengguna dan spesifikasi yang telah ditentukan [15]. Berikut hasil pengujian yang akan ditampilkan pada tabel 1 yang sudah dilakukan.

Tabel 1. Hasil pengujian menggunakan black-box testing

No	Pengujian	Case	Output (diharapkan)	Hasil	Kesimpulan
1.	Entri data koleksi	Memasukkan data koleksi melalui fitur create koleksi	Menampilkan data koleksi baru yang berhasil disimpan	Sesuai	Valid
2.	Mengubah data koleksi	Memasukkan data perubahan koleksi	Menampilkan data koleksi yang telah diperbarui	Sesuai	Valid
3.	Menghapus data koleksi	Memasukkan ID koleksi yang akan dihapus	Menampilkan pesan bahwa data koleksi telah dihapus	Sesuai	Valid
4.	Menampilkan daftar koleksi	Mengakses fitur daftar koleksi	Menampilkan seluruh data koleksi	Sesuai	Valid
5.	Menambahkan koleksi ke keranjang	Memilih koleksi dan menambahkannya ke keranjang	Data koleksi masuk ke keranjang	Sesuai	Valid
6.	Entri data katalog	Memasukkan data katalog	Menampilkan data katalog baru	Sesuai	Valid
7.	Menampilkan daftar katalog	Mengakses fitur daftar katalog	Menampilkan seluruh data katalog	Sesuai	Valid
8.	Menambahkan katalog ke keranjang	Memilih katalog dan menambahkannya ke keranjang	Data katalog masuk ke keranjang	Sesuai	Valid
9.	Entri data anggota	Memasukkan data anggota	Menampilkan data anggota baru	Sesuai	Valid
10.	Menampilkan daftar anggota	Mengakses fitur daftar anggota	Menampilkan seluruh data anggota	Sesuai	Valid
11.	Menampilkan daftar perpanjangan anggota	Mengakses fitur daftar perpanjangan	Menampilkan data perpanjangan anggota	Sesuai	Valid
12.	Menambahkan anggota ke keranjang	Memilih anggota dan menambahkannya ke keranjang	Data anggota masuk ke keranjang	Sesuai	Valid
13.	Entri peminjaman koleksi	Memasukkan data peminjaman	Menampilkan data peminjaman baru	Sesuai	Valid
14.	Menampilkan daftar peminjaman	Mengakses fitur daftar peminjaman	Menampilkan seluruh data peminjaman	Sesuai	Valid
15.	Entri perpanjangan peminjaman	Memasukkan data perpanjangan	Menampilkan data perpanjangan	Sesuai	Valid
16.	Entri pengembalian koleksi	Memasukkan data pengembalian	Menampilkan data pengembalian	Sesuai	Valid

17.	Menampilkan daftar pengembalian	Mengakses fitur daftar pengembalian	Menampilkan data pengembalian	Sesuai	Valid
18.	Menampilkan data pelanggaran	Mengakses fitur data pelanggaran	Menampilkan data pelanggaran anggota	Sesuai	Valid
19.	Peminjaman loker	Memasukkan data peminjaman loker	Menampilkan data peminjaman loker	Sesuai	Valid
20.	Pengembalian loker	Memasukkan data pengembalian loker	Menampilkan data pengembalian loker	Sesuai	Valid
21.	Buku tamu anggota	Memasukkan data kunjungan anggota	Menampilkan data buku tamu anggota	Sesuai	Valid
22.	Buku tamu rombongan	Memasukkan data kunjungan rombongan	Menampilkan data buku tamu rombongan	Sesuai	Valid
23.	Baca di tempat	Memasukkan data anggota yang membaca di tempat	Menampilkan data baca di tempat	Sesuai	Valid
24.	Pengembalian koleksi baca di tempat	Memasukkan data pengembalian koleksi	Menampilkan status koleksi kembali tersedia	Sesuai	Valid
25.	Laporan katalog	Mengakses fitur laporan katalog	Menghasilkan file laporan katalog	Sesuai	Valid
26.	Laporan koleksi	Mengakses fitur laporan koleksi	Menghasilkan file laporan koleksi	Sesuai	Valid
27.	Laporan anggota	Mengakses fitur laporan anggota	Menghasilkan file laporan anggota	Sesuai	Valid
28.	Laporan sirkulasi	Mengakses fitur laporan sirkulasi	Menghasilkan file laporan sirkulasi	Sesuai	Valid
29.	Laporan buku tamu	Mengakses fitur laporan buku tamu	Menghasilkan file laporan buku tamu	Sesuai	Valid
30.	Laporan baca di tempat	Mengakses fitur laporan baca di tempat	Menghasilkan file laporan baca di tempat	Sesuai	Valid
31.	Laporan loker	Mengakses fitur laporan loker	Menghasilkan file laporan loker	Sesuai	Valid
32.	Menambah data user	Memasukkan data user	Menampilkan data user baru	Sesuai	Valid
33.	Mengubah data user	Memasukkan perubahan data user	Menampilkan data user yang telah diperbarui	Sesuai	Valid
34.	Menghapus data user	Memasukkan ID user	Menampilkan pesan bahwa data user telah dihapus	Sesuai	Valid
35.	Menampilkan data user	Mengakses fitur daftar user	Menampilkan seluruh data user	Sesuai	Valid

3.5 Pengujian

Dalam pengembangan sistem informasi perpustakaan yang mengadopsi struktur INLISLite, tahap pemeliharaan bertujuan untuk menjaga stabilitas dan keamanan sistem setelah implementasi. Pemeliharaan dilakukan untuk memastikan seluruh fitur, seperti pengelolaan katalog,

koleksi, sirkulasi, layanan kunjungan, loker, serta pembuatan laporan tetap berjalan sesuai kebutuhan operasional. Kegiatan pemeliharaan mencakup perbaikan kesalahan sistem, penyesuaian fitur berdasarkan umpan balik pengguna, serta pemantauan performa dan keamanan data. Dengan dilakukannya proses

pemeliharaan yang terstruktur dan berkelanjutan, sistem akan berfungsi secara optimal dan mampu memenuhi kebutuhan pengguna dalam jangka panjang.

KESIMPULAN

Berdasarkan hasil pengerjaan backend sistem INLISlite, dapat disimpulkan bahwa pengembangan backend telah diselesaikan sesuai dengan hasil diskusi awal bersama pihak perpustakaan dan mentor di UPA TIK Undiksha. Pengelolaan data pada sistem ini disederhanakan melalui API yang dirancang sesuai dengan fungsinya, sehingga mampu mendukung kebutuhan perpustakaan Undiksha secara lebih efisien dan efektif. Namun demikian penelitian ini masih memiliki keterbatasan pada integrasi dengan sistem SSO Undiksha dan sistem Opac. Untuk pengembangan selanjutnya, disarankan agar dilakukan integrasi lanjutan dengan sistem SSO dan Opac sehingga sistem dapat memberikan layanan yang optimal dan terus memenuhi kebutuhan civitas akademika Undiksha dengan baik.

UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

DAFTAR PUSTAKA

- [1] N. M. Mahardika, I. M. E. Listartha, and I. N. T. A. Putra, "RANCANG BANGUN LEARNING MANAGEMENT SYSTEM DALAM PENGELOLAAN KURSUS BAHASA INGGRIS (STUDI KASUS: ONE STEP SOLUTION (OSS) BALI)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3, Jul. 2025, doi: 10.23960/jitet.v13i3.6730.
- [2] N. Made *et al.*, "SISTEM INFORMASI INVENTORY PADA PT. DJAYA BUAH BERSINAR DENPASAR BERBASIS WEB," *INSERT: Information System and Emerging Technology Journal*, vol. 3, no. 2, 2022.
- [3] E. Nurhayati and A. Agussalim, "Rancang Bangun Back-end API pada Aplikasi Mobile AyamHub Menggunakan Framework Node JS Express," *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 3, p. 524, Jul. 2023, doi: 10.26418/justin.v11i3.66823.
- [4] Y. Muharam and T. Hidayat, "PENGEMBANGAN APLIKASI BACK-END E-COMMERCE MENGGUNAKAN REST API GOLANG UNTUK OPTIMALISASI KINERJA SERVER," 2024.
- [5] G. Saputra, E. Didik Widiyanto, A. Fatchur Rochim, E. D. Widiyanto, A. F. Rochim, and R. Bangun Backend Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja Laravel Berbasis, "RESTful API," *Jurnal Teknik Komputer*, vol. 2, no. 1, pp. 91–98, 2023, doi: 10.14710/jtk.v2i1.38320.
- [6] A. Rahmi and N. Najamudin, "Efektivitas Penerapan Sistem Aplikasi INLISLITE (Integrated Library System) Pada Kegiatan Penginputan Data Dinas Perpustakaan dan Kearsipan Kabupaten Aceh Barat," *Journal of Social Politics and Governance (JSPG)*, vol. 4, no. 2, pp. 128–141, Dec. 2022, doi: 10.24076/jspg.v4i2.931.
- [7] Misna Asqia, "Development of a Backend Architecture for an Online Mental Counselling Platform to Enhance Performance and Security," *The Indonesian Journal of Computer Science*, vol. 14, no. 2, Apr. 2025, doi: 10.33022/ijcs.v14i2.4748.
- [8] N. Made Mila Rosa Desmayani, N. Wayan Wardani, P. Gede Surya Cipta Nugraha, and G. Surya Mahendra, "Sistem Informasi Laporan Keuangan pada Salon Berbasis Website Dengan Metode SDLC,"

- Jurnal Sistem Informasi dan Komputer Terapan Indonesia (JSIKTI)*, vol. 4, no. 2, pp. 68–77, 2021, doi: 10.22146/jsikti.xxxx.
- [9] I. Putu *et al.*, “RANCANG BANGUN SISTEM INFORMASI PENJUALAN DAN PIUTANG BERBASIS WEBSITE PADA TOKO INTI ALAM,” *INSERT: Information System and Emerging Technology Journal*, vol. 3, no. 2, 2022.
- [10] T. Anjarsari and F. Ardiani, “Application of Rest Api Technology in Android-Based Beauty Salon Service Reservation System,” *Journal of Computer Science and Technology Studies*, 2023, doi: 10.32996/jcsts.
- [11] D. Made, D. U. Putra, G. S. Mahendra, and E. Mulyadi, “SISTEM INFORMASI PENERIMAAN SISWA BARU PADA SMP NEGERI 3 CIBAL BERBASIS WEB,” *INSERT: Information System and Emerging Technology Journal*, vol. 3, no. 1, 2022.
- [12] ad Muzakki Nurjamil, “Rancang Bangun Back-End Sistem Informasi Pertanian Berbasis Website di Kabupaten Sumbawa (Back-End Development of a Web-Based Agricultural Information System in Sumbawa),” *JEET: Journal of Engineering and Emerging Technology*, vol. 3, no. 1, 2025.
- [13] I. R. D. Muhammad and I. V. Papatungan, “Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System,” *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 3, no. 2, pp. 79–87, Jan. 2024, doi: 10.20885/snati.v3.i2.35.
- [14] P. Risma, E. Pande, N. Tri, A. Putra, N. Wayan, and S. Putri, “RANCANG BANGUN SISTEM INFORMASI PENGAJUAN KREDIT PADA BUM DESA BERSAMA SANTHI SEDANA,” *TELEMATIKA*, vol. 17, no. 2, pp. 171–181, 2020.
- [15] G. S. Mahendra and I. K. A. Asmarajaya, “Evaluation Using Black Box Testing and System Usability Scale in the Kidung Sekar Madya Application,” *Sinkron*, vol. 7, no. 4, pp. 2292–2302, Oct. 2022, doi: 10.33395/sinkron.v7i4.11755.