

BAB I

PENDAHULUAN

1.1 Latar Belakang

Universitas Pendidikan Ganesha (Undiksha) merupakan Universitas yang ada di Bali Utara, tepatnya di kota Singaraja, Kabupaten Buleleng. Undiksha memiliki komitmen dalam mencetak lulusan yang unggul dalam bidang Pendidikan maupun Non-Kependidikan. Falsafah Tri Hita Karana dijadikan landasan utama oleh Undiksha dalam penyelenggaraan tridharma perguruan tinggi, sebagai upaya menjadi universitas yang unggul di tahun 2045. Visi ini dijabarkan ke dalam misi Undiksha yang mencakup (1) Peningkatan kualitas sumber daya manusia; (2) Pengembangan dan penerapan IPTEK melalui penelitian; (3) Pengabdian kepada masyarakat yang kompetitif, kolaboratif, akomodatif, dan inovatif.

Dalam merealisasikan visi dan misi tersebut, tentunya Undiksha perlu adanya dukungan proses pengambilan keputusan yang strategis dan berbasis data yang akurat serta kredibel. Dari hasil proses tersebut akan digunakan sebagai proses pengambilan keputusan-keputusan yang penting dalam berbagai lingkup diantaranya lingkup akademik, keuangan, serta pengelolaan kelembagaan dari berbagai lapisan pimpinan, mulai di tingkat lembaga, fakultas, jurusan, dan program studi. Tentunya, kebutuhan akan data yang akurat, kredibel, dan mudah diakses menjadi hal yang krusial dalam proses pengambilan keputusan yang efektif dan efisien di tingkat eksekutif. Undiksha melalui Unit Penunjang Akademik Teknologi, Informasi dan Komunikasi (UPA TIK) Undiksha mengembangkan sebuah platform berbasis web yang bernama *Executive Information System* dalam

bentuk dashboard untuk mendukung efektivitas manajemen akademik dan keuangan.

Executive Information System (EIS) merupakan sistem informasi berbasis komputer sebagai pendukung eksekutif yang dirancang dengan fokus untuk membantu *monitoring*, *filtering* dan menata informasi eksekutif, sehingga konsumsi informasi menjadi lebih efektif dan relevan. EIS dikembangkan dengan dasar ide berupa alat untuk membantu pihak eksekutif dalam mengakses informasi yang relevan guna manajemen aktivitas mereka. Dalam pengembangannya, EIS memiliki tujuan umum meliputi : (1) mengurangi data yang membombardir eksekutif, (2) meningkatkan relevansi, ketepatan dan kegunaan informasi sebagai dasar tindakan eksekutif, (3) memfokuskan tim manajemen dalam menangani masalah-masalah krisis yang mempengaruhi kesuksesan perusahaan, (4) mempercepat tugas-tugas eksekutif, (5) sebagai indikator awal, misalnya terjadi pergeseran selera konsumen (Payamta, 1998).

Executive Information System (EIS) yang dikembangkan oleh UPA TIK Undiksha saat ini diperuntukkan sebagai sarana dalam mendukung proses manajemen akademik dan keuangan yang ditunjukkan kepada pihak eksekutif Undiksha. Namun untuk saat ini EIS hanya bisa diakses oleh pimpinan yang berstatus dosen saja, pimpinan yang berstatus dosen meliputi Rektor, Wakil Rektor, Dekan, Wakil Dekan, Ketua Jurusan, Koordinator Program Studi, dan Penjaminan Mutu. Sedangkan pimpinan yang di luar dosen seperti pegawai tidak bisa mengakses sistem ini, pimpinan yang termasuk pegawai yaitu Biro Akademik dan Keuangan. Dalam perjalanannya EIS Undiksha saat ini masih menghadapi beberapa tantangan khususnya dalam segi performa dan arsitektur *backend*.

Berdasarkan penelitian terdahulu oleh (Novianti, 2024) yang berfokus pada redesign UI pada *executive information system* (EIS) Universitas Pendidikan Ganesha agar lebih responsif dan informatif. Berdasarkan hasil wawancara yang dilakukan pada penelitian tersebut, ditemukan bahwa pihak eksekutif, seperti Koordinator Prodi Sistem Informasi mengalami kendala dari segi kebutuhan data yang belum tersedia secara terintegrasi sehingga harus diminta manual ke UPA TIK. Pihak helpdesk UPA TIK juga mengkonfirmasi hal yang serupa termasuk kelengkapan data, cakupan aktor eksekutif, dan halaman yang tidak bisa diakses. Disisi lain, Ketua UPA TIK menegaskan bahwasannya sistem EIS ini masih belum optimal dalam memenuhi fungsinya. Temuan-temuan ini menunjukkan bahwa akar permasalahan pada EIS tidak hanya terletak pada aspek *User Interface* (UI), melainkan pada keterbatasan integrasi dan pengelolaan data pada sisi backend yang mengakibatkan keterlambatan, ketidakakuratan, serta keterbatasan penyajian informasi eksekutif.

Permasalahan tersebut juga dikonfirmasi melalui hasil wawancara seperti pada lampiran 3 bersama *programmer* yang bertanggung jawab dalam mengelola sistem EIS, disebutkan bahwa EIS masih memiliki kendala dalam aspek penyajian data dan performa sistem. Dalam proses wawancara tersebut programmer mendemonstrasikan permasalahan yang dialami yaitu ketika masuk ke salah satu menu di website EIS yang merupakan menu data mahasiswa, pada saat bernavigasi ke menu tersebut *loading* untuk merender halaman secara utuh membutuhkan waktu yang lama yaitu sekitar 11 detik. Disisi lain dijelaskan secara arsitektur, bahwa EIS Undiksha menggunakan arsitektur monolitik yang berbasis PHP dengan *Framework* Laravel dan proses integrasi data dilakukan dengan mengambil data

dari berbagai sumber seperti data mahasiswa, data pegawai, data dosen dan yang lainnya melalui tembakan API (*Application Programming Interface*) yang dilakukan secara berkala di malam hari, kemudian data tersebut disimpan di database inti yang berbasis SQL untuk nantinya diolah dan dikonsumsi dalam bentuk analitik dan visualisasi.

Permasalahan utama muncul ketika sistem harus menampilkan data analitik yang kompleks. Arsitektur yang digunakan saat ini masih bersifat monolitik yang berarti seluruh komponen dari logika bisnis, *frontend*, dan pengelolaan data menjadi satu kesatuan serta data yang digunakan berasal dari struktur relasional SQL, namun setelah dilakukan analisis terhadap struktur data yang digunakan ternyata tidak ada relasi antar tabel secara langsung pada tingkat skema. Pendekatan ini tidak efektif karena beban proses berpindah ke level aplikasi menyebabkan server aplikasi terbebani dan waktu respon meningkat, dimana selain melakukan proses *query* proses lain seperti logika *join*, *grouping*, dan agregasi data dilakukan langsung dalam sistem. Hal ini sejalan dengan apa yang disampaikan oleh (Stonebraker & Çetintemel, 2018) bahwa data warehouse berbeda dengan OLTP, OLTP diperuntukkan untuk aktivitas utama bisnis contohnya seperti menjual produk atau layanan, sedangkan data warehouse berfokus pada *query* ad-hoc yang cukup kompleks, dan disampaikan dengan jelas bahwa sistem basis data relasional yang digunakan pada OLTP tidak optimal untuk kebutuhan data *warehouse*, terutama untuk proses *query* ad-hoc dan analisis.

lengkap nantinya dilakukan proses query seperti *join* dan semacannya untuk mendapatkan informasi tersebut, dikarenakan pada tingkat skema tidak ada relasi sehingga menyebabkan proses relasi data sepenuhnya dikendalikan di tingkat lapisan aplikasi. Kondisi ini menyebabkan beban proses dan pengelolaan data ke sisi aplikasi terutama untuk proses-proses yang memerlukan transformasi data dari data yang berelasi ke data untuk kebutuhan analisis seperti *join*, *aggregation*, *grouping* dan lainnya.

Untuk menjawab permasalahan ini, beberapa penelitian menyarankan pendekatan yang lebih modular seperti pada penelitian (Chandra & Tan, 2020) membahas mengenai proses pembuatan dashboard pada salah satu instansi dengan pendekatan pemisahan layanan antara *frontend* dan *backend*, pemisahan dilakukan untuk mempermudah pemeliharaan sistem, backend dibuat dengan API yang berfokus untuk mengakses informasi yang dibutuhkan dan frontend akan berfokus sebagai penampil terkait data yang dikonsumsi dari backend, hasilnya pendekatan ini mampu membantu proses bisnis yang berlangsung di instansi tersebut. Penelitian (Mukhlash et al., 2025) juga membahas mengenai pembuatan dashboard berbasis Internet of Things (IoT) guna mendukung operasional *Vertical Crab House*, pendekatan dilakukan dengan memisahkan layanan *frontend* dan *backend*, *backend* akan berperan sebagai pengelola data utama dan menangani logika bisnis serta manajemen produk kepiting guna memastikan distribusi yang efektif, dan *frontend* akan berfokus pada konsumsi data dari *backend*, melalui pendekatan ini sistem mampu dioptimalkan dari segi antarmuka pengguna menjadi intuitif dan pengelolaan data serta distribusi secara efisien. Dalam konteks ini, arsitektur sistem

EIS Undiksha perlu dilakukan upaya proses *reengineering* untuk memisahkan layanan *frontend* dan *backend*.

Proses *reengineering* dalam konteks penelitian ini yaitu pengembangan pada sisi arsitektur *backend* EIS Undiksha yang lebih efisien dan skalabel. *Backend* akan dirancang dengan memanfaatkan proses ETL (*Extract, Transform, Load*). ETL merupakan rangkaian proses yang melibatkan pengumpulan, penyaringan, pengolahan, dan penggabungan data dari berbagai jenis sumber data untuk menghasilkan data yang lebih terstruktur (Putra et al., 2020a). Penelitian oleh (Ghani & Kurniawan, 2024a) merancang data *warehouse* dengan pendekatan ETL yang bertujuan untuk menciptakan sistem yang memungkinkan analisis data secara efisien dengan mengintegrasikan data dari berbagai dimensi tabel, melalui pendekatan ini sistem mampu mengatasi permasalahan analisis data dan ketidakefisiensian akses melalui penyimpanan data yang terpusat dan rapi. Pada EIS Undiksha, pendekatan ETL ini akan menghasilkan data OLAP (*Online Analytical Processing*), dikarenakan hasil yang diharapkan berupa data OLAP yang sesuai dengan tujuan EIS untuk menyajikan data dalam bentuk analitik.

Pemisahan OLTP dan OLAP sangat penting dilakukan untuk memisahkan kebutuhan data transaksional dan analitik tidak saling tercampur serta membebani, menurut (Maysanjaya & Dermawan, 2024) OLTP dan OLAP memiliki peranan yang berbeda sesuai dengan kebutuhan, OLTP dijelaskan cocok untuk menangani proses transaksi secara *real-time* karena basis data ini mengutamakan integritas data dan konsistensi dengan struktur yang kaku dan terdefinisi dengan baik. Sebaliknya OLAP lebih baik diimplementasikan untuk menganalisis data secara lebih mendalam karena basis data ini berfungsi di lingkungan yang dinamis dengan

struktur yang lebih fleksibel dibandingkan dengan OLTP. Dalam mendukung kebutuhan OLTP dan OLAP basis data yang digunakan juga berbeda, OLTP sangat cocok menggunakan basis data relasional yaitu SQL (*Structured Query Language*) karena sudah dirancang guna mendukung integrasi data dan konsistensi melalui aturan ACID (*Atomicity, Consistency, Isolation, Durability*) yang menjaga keamanan dan keandalan data, sedangkan OLAP lebih cocok menggunakan basis data NoSQL (*Not Only SQL*) karena lebih fleksibel dan efisien dalam menangani data yang semi-terstruktur atau tidak terstruktur. Secara garis besar pemisahan data OLTP dan OLAP sangat penting dilakukan begitu juga dengan penggunaan basis data yang sesuai, seperti SQL cocok untuk OLTP dan NoSQL cocok untuk OLAP.

Dalam menjembatani komunikasi antar *frontend* dengan kebutuhan data yang sebagian besar merupakan data OLAP pada sistem EIS, akan dikembangkan RESTful API sebagai solusinya. Melalui RESTful API aplikasi tidak lagi bergantung untuk melakukan *query* secara langsung ke database dan mengirolahnya untuk menyajikan data analitik, pendekatan ini juga akan membuat sistem menjadi lebih terdistribusi dalam segi pengelolaan beban sistem dan kemudahan pengembangan yang lebih terstruktur dan berkelanjutan. Menurut (Biehl, 2016) API yang dirancang dengan mematuhi semua *constraints* dari REST dinamakan RESTful API, guna menjembatani komunikasi antar sistem melalui metode HTTP dengan menggunakan beberapa metode seperti GET, POST, PUT, dan DELETE. Keunggulan yang diberikan RESTful API yaitu dari sisi proses implementasi yang sederhana dan mampu meningkatkan skalabilitas serta fleksibilitas sistem.

Dalam upaya mendukung proses pengembangan arsitektur backend yang modular dan kompleks ini, tentunya membutuhkan pendekatan metode

pengembangan perangkat lunak yang adaptif dan iteratif. Salah satu metode pengembangan perangkat lunak yang efektif dalam kasus ini yaitu metode *Agile* dengan kerangka kerja *Scrum*, seperti penelitian yang dilakukan oleh (Firdaus & Prayudi, 2024) menggunakan kerangka kerja *Scrum* untuk mengembangkan sistem *backend* untuk mendukung aplikasi Samakaki yaitu media sosial yang ramah terhadap anak, metode ini dipilih karena memudahkan proses pengembangan dalam mengetahui langkah-langkah atau *workflow* yang harus dikerjakan dalam proyek serta metode ini efektif untuk digunakan. Disisi lain penelitian oleh (Nurmasani et al., 2024) juga menggunakan metode *Agile* dengan kerangka kerja *Scrum* dalam pengembangan sistem pencatatan magang, pemilihan metode ini karena *Scrum* memiliki konsep pengerjaan yang bertahap atau *incremental* dan memastikan hasil dari perangkat lunak sesuai dengan yang diharapkan dengan penerapan pengulangan pada tahapan-tahapan tertentu yang perlu disesuaikan, melalui pendekatan dengan metode ini yang terdiri dari tahapan *product backlog*, *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective* produk yang dihasilkan sesuai dengan keinginan *stakeholder*.

Secara garis besar, dapat disimpulkan bahwa sistem EIS Undiksha saat ini masih mengalami kendala dari segi performa dan pengelolaan data. Pokok permasalahan tersebut menjadi landasan pengoptimalan *Executive Information System* yang berfokus pada sisi *backend*. Melalui penelitian ini bertujuan untuk melakukan proses *reengineering* arsitektur EIS Undiksha dengan menerapkan prinsip pemisahan layanan *frontend* dan *backend*, dimana sisi *backend* menjadi fokus utama pada penelitian ini dengan menerapkan proses ETL untuk menghasilkan informasi yang dibutuhkan EIS yaitu sebagian besar berupa data OLAP yang diolah

dari berbagai sumber dan nantinya akan disimpan di database NoSQL, selanjutnya agar data dapat dikonsumsi oleh *frontend* dilakukan pengembangan RESTful API sebagai jembatan antara *frontend* dan *database*. Dalam proses pengembangan yang modular dan kompleks digunakan pendekatan metode *Agile* dengan kerangka kerja *Scrum*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijabarkan, maka identifikasi masalah yang dapat diuraikan yaitu :

1. Belum adanya pemisahan antara *frontend* dan *backend* karena masih menggunakan arsitektur monolitik sehingga menyebabkan beban server meningkat.
2. Belum ada pemisahan data secara arsitektur untuk data OLAP sebagai data analitik sehingga menyebabkan proses *query* menjadi berat dan memakan waktu yang lama.

Berdasarkan hasil identifikasi masalah, maka ditarik rumusan masalah yang menjadi dasar pada penelitian ini yaitu:

1. Bagaimana rancangan arsitektur *backend* modular pada *Executive Information System* (EIS) dengan ETL dan RESTful API untuk optimalisasi pengelolaan dan penyajian data OLAP?
2. Bagaimana hasil evaluasi sistem berdasarkan rancangan arsitektur *backend* modular yang dikembangkan?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang dijabarkan, maka tujuan dari penelitian ini sebagai berikut :

1. Merancang arsitektur backend modular pada *Executive Information System* (EIS) dengan ETL dan RESTful API untuk optimalisasi pengelolaan dan penyajian data OLAP.
2. Hasil evaluasi sistem berdasarkan rancangan arsitektur backend modular yang dikembangkan.

1.4 Ruang Lingkup Penelitian

Penelitian ini berfokus pada pengelolaan data pada arsitektur backend dalam *Executive Information System* (EIS) Undiksha, dengan ruang lingkup yang dibahas mencakup:

1. Mengembangkan sistem *backend* dengan pendekatan arsitektur yang modular.
2. Mengelola data dari data sumber agar sesuai dengan kebutuhan informasi eksekutif dan lebih terstruktur melalui penerapan ETL (*Extract, Transform, Load*).
3. Data disimpan terpisah berdasarkan jenisnya:
 - OLTP sebagai data transaksi disimpan pada *database SQL*
 - OLAP sebagai data analitik disimpan pada *database NoSQL*
4. Menyediakan akses komunikasi antara *frontend* dan *database backend* melalui pengembangan *RESTful API*.

5. Fokus utama pada untuk meningkatkan efisiensi pengelolaan dan penyajian data pada sisi *backend* EIS Undiksha, tanpa mencakup rekonstruksi pada sisi *frontend*.
6. *Testing* dilakukan berdasarkan jenis dari pengujian perangkat lunak (*software testing*) diantaranya *unit testing*, *integration testing*, dan *load testing* untuk mengetahui seberapa baik arsitektur *backend* yang telah dikembangkan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat bagi pihak-pihak yang terlibat. Adapun manfaat dari penelitian ini sebagai berikut:

1. Bagi Universitas Pendidikan Ganesha

Sistem yang lebih terstruktur dan efisien diharapkan mempercepat akses dan proses data eksekutif serta memudahkan manajemen dalam proses pengambilan keputusan yang berbasis data yang akurat.

2. Bagi UPA TIK Undiksha

Hal ini diharapkan dapat membantu UPA TIK Undiksha dalam mengoptimalkan *Executive Information System* (EIS) Undiksha melalui penerapan ETL dan RESTful API. Sehingga performa sistem menjadi lebih optimal dengan adanya pengelolaan data yang lebih terstruktur dan modular.

3. Bagi Peneliti

Peneliti diharapkan dapat menjadikan sarana untuk mengimplementasikan ilmu yang didapat selama perkuliahan, khususnya dalam

implementasi proses ETL dan pengembangan RESTful API serta memberikan kontribusi dalam pengembangan arsitektur backend yang modular dan efisien. Sehingga dapat memberikan kebermanfaatan bagi sasaran.

