KERJA PRAKTIK - EF234603

Perancangan dan Implementasi Backend pada myITS Academy Modul MKU dan myITS StudentOrganization di ITS

Direktorat Pengembangan Teknologi dan Sistem Informasi
Kampus Sukolilo, Gedung Pusat Riset Lantai 4, Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, East Java 60117
Periode: 13 Maret 2024 – 31 Desember 2024

Oleh:

Yoel Mountanus Sitorus 5025211078

Pembimbing Jurusan
Hadziq Fabroyir, S.Kom., Ph.D

Pembimbing Lapangan
Akhmad Budi Kurniawan

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

KERJA PRAKTIK – EF234603

Perancangan dan Implementasi Backend pada myITS Academy Modul MKU dan myITS StudentOrganization di ITS

Direktorat Pengembangan Teknologi dan Sistem Informasi
Kampus Sukolilo, Gedung Pusat Riset Lantai 4, Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, East Java 60117
Periode: 13 Maret 2024 - 31 Desember 2024

Oleh:

Yoel Mountanus Sitorus 5025211078

Pembimbing Jurusan
Hadziq Fabroyir, S.Kom., Ph.D

Pembimbing Lapangan
Akhmad Budi Kurniawan

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

DAFTAR ISI

DAFTAR KODE SUMBER

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
KERJA PRAKTIK

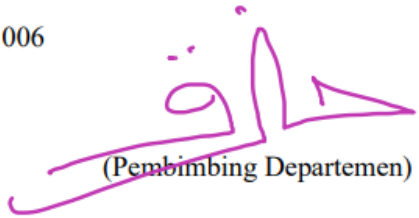
Perancangan dan Implementasi Golang pada myITS Academy Modul MKU dan myITS Student Organization di ITS

Oleh:

Yoel Mountanus Sitorus 5025211078

Disetujui oleh Pembimbing Kerja Praktik:

1. Hadziq Fabroyir, S.Kom., Ph.D.
   NIP. 198602272019031006



(Pembimbing Departemen)

2. Akhmad Budi Kurniawan
   NIP. 198204152014091004

(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Perancangan dan Implementasi Backend pada myITS Academy Modul MKU dan myITS StudentOrganization di ITS

Nama Mahasiswa	: Yoel Mountanus Sitorus
NRP	: 5025211078
Departemen	: Teknik Informatika FTEIC-ITS
Pembimbing Departemen	: Hadziq Fabroyir, S.Kom., Ph.D.
Pembimbing Lapangan	: Akhmad Budi Kurniawan

ABSTRAK

Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) Institut Teknologi Sepuluh Nopember (ITS), adalah unit yang bertanggung jawab dalam pengembangan dan pengelolaan teknologi informasi di ITS. Saat melaksanakan Kerja Praktik, saya berkontribusi dalam pengembangan sistem myITS Academy, sebuah platform yang mendukung kebutuhan belajar mahasiswa secara daring, dan sistem myITS Organisasi Mahasiswa (Ormawa), yang mempermudah pengelolaan kegiatan organisasi mahasiswa di ITS. Pengguna utama kedua sistem ini adalah mahasiswa yang memanfaatkan teknologi untuk mendukung kegiatan akademik dan organisasi mereka di kampus.

Aplikasi dibuat dengan menggunakan framework Backend Gin-Golang dengan Arsitektur Domain-Driven Design (DDD). Bagian yang saya dan tim kerjakan adalah module MKU pada myITS Academy tentang penetapan kelas secara otomatis kepada mahasiswa di mata kuliah umum (MKU). Dan Module Ormawa pada myITS Ormawa yang mempunyai fitur-fitur yang beberapa diantaranya adalah CRUD Pengurus, Organisasi, Unit Anggaran, dan lainnya.

Kata Kunci: Website, DPTSI, Gin-Golang, Backend, myITS

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan YME atas berkat dan kuasa-Nya penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Perancangan dan Implementasi myITS Academy dan myITS Student Organization Sistem Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) Institut Teknologi Sepuluh Nopember

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Hadziq Fabroyir, S.Kom., Ph.D selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Akhmad Budi Kurniawan selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Kepada Tim DPTSI yang bersedia membimbing dan membantu penulis dalam pelaksaan Kerja Praktik.

Surabaya, 28 Desember 2024
Yoel Mountanus Sitorus

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini, teknologi informasi berkembang dengan sangat cepat dan membawa dampak besar di berbagai sektor. Teknologi baru terus bermunculan dengan fitur yang lebih baik, menggantikan teknologi lama yang tidak lagi memenuhi kebutuhan pengguna. Institut Teknologi Sepuluh Nopember (ITS) sebagai salah satu perguruan tinggi negeri di Indonesia, berperan tidak hanya dalam memberikan pendidikan tetapi juga dalam memanfaatkan teknologi terbaru untuk meningkatkan kualitas pembelajaran. ITS secara bertahap memperbarui teknologi yang digunakan agar tetap relevan dengan perkembangan zaman.

Pada kerja praktik ini, penulis mengambil peran sebagai pengembang backend yang bertanggung jawab melakukan migrasi sistem backend dari PHP ke Golang dengan menggunakan pendekatan *clean architecture*. Dan juga pada myITS Ormawa penulis melakukan pembuatan website Ormawa dengan Golang dengan pendekatan arsitektur yang sama. Tujuan dari migrasi ini adalah untuk meningkatkan kinerja aplikasi web, khususnya dalam hal kecepatan akses dan kemudahan pengembangan fitur. Pendekatan *clean architecture* juga mempermudah pengelolaan dan pengembangan sistem di masa mendatang, sehingga aplikasi dapat terus berkembang sesuai kebutuhan pengguna.

1.2 Tujuan

Tujuan dari pelaksanaan kerja praktik ini adalah untuk menambah pengalaman, portfolio, menyelesaikan

kewajiban nilai kerja praktik sebesar 4 sks, dan mengambil bagian tim dalam pengembangan system informasi di Ibu yang luhur ITS dengan memberikan kontribusi pada pengembangan website yang mendukung kegiatan dan organisasi dalam ITS.

1.3 Manfaat

Proses migrasi teknologi ini memberikan dampak positif, baik bagi penulis maupun ITS. Bagi penulis, proyek ini menjadi sarana untuk memahami penerapan teori ke dalam praktik, khususnya dalam pengembangan aplikasi dengan teknologi modern. Penulis juga mendapat wawasan tentang pentingnya efisiensi dan skalabilitas dalam sistem, serta bagaimana memilih solusi teknis yang tepat untuk kebutuhan spesifik. Selain itu, keterlibatan langsung dalam proyek ini membantu penulis mengasah kemampuan analisis dan adaptasi terhadap tantangan teknis yang kompleks.

Bagi ITS, implementasi teknologi baru ini meningkatkan kemampuan sistem dalam menangani kebutuhan akademik yang terus berkembang. Dengan performa yang lebih optimal, aplikasi ini diharapkan dapat mendukung operasional kampus secara lebih efektif, mempermudah akses pengguna, dan menciptakan pengalaman yang lebih intuitif bagi mahasiswa dan dosen..

1.4 Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana penerapan proses migrasi Sistem Akademik ITS dari PHP ke Golang dilakukan secara efektif?

2. Apa saja tantangan yang muncul selama proses migrasi teknologi backend?
3. Bagaimana penerapan clean architecture dapat mendukung keberlanjutan pengembangan dan pemeliharaan sistem aplikasi web di ITS?

1.5 Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan secara hybrid, di mana penulis memiliki fleksibilitas untuk bekerja dari mana saja (WFH) namun diwajibkan hadir di kantor untuk melakukan laporan progress mingguan. Lokasi kerja praktik berada di Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) - ITS, Gedung Pusat Riset Lantai 4, ITS Kampus Sukolilo, Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, Jawa Timur 60117. Kerja praktik dimulai pada tanggal 13 Maret 2024 dan berakhir pada tanggal 31 Desember 2024.

1.6 Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1 Perumusan Masalah

Tahap awal berfokus pada identifikasi kebutuhan dan masalah dalam dua sistem: myITS Academy dan myITS Ormawa. Untuk myITS Academy, penulis bersama tim DPTSI mengidentifikasi modul penting seperti mata kuliah umum (MKU) yang membutuhkan optimasi, khususnya dalam penetapan kelas otomatis bagi mahasiswa. Sedangkan untuk myITS Ormawa, penulis mengidentifikasi kebutuhan pengelolaan organisasi mahasiswa, termasuk fitur seperti manajemen pengurus, organisasi, unit anggaran, dan lainnya.

1.6.2 Studi Literatur

Setelah masalah diidentifikasi, penelitian dilakukan terhadap teknologi yang relevan, seperti framework Gin-Golang dan pendekatan Domain-Driven Design (DDD). Studi ini bertujuan untuk memahami bagaimana teknologi dan metode tersebut dapat diimplementasikan secara efektif dalam pengembangan sistem berbasis web.

1.6.3 Analisis dan Perancangan Sistem

Berdasarkan hasil penelitian dan diskusi, analisis kebutuhan dilakukan untuk kedua sistem. Pada myITS Academy, desain sistem difokuskan pada migrasi teknologi dari PHP ke Golang dengan pendekatan modular yang memisahkan logika bisnis, antarmuka pengguna, dan penyimpanan data. Sementara itu, pada myITS Ormawa, sistem dirancang dari nol menggunakan pendekatan DDD untuk memastikan struktur aplikasi yang terorganisir dan fleksibel dalam pengelolaan data organisasi mahasiswa.

Pada myITS Academy, Penulis melakukan migrasi sistem backend dari PHP ke Golang, dengan fokus pada modul MKU untuk otomatisasi penetapan kelas mahasiswa. Dan pada myITS Ormawa, Penulis membangun sistem baru dari awal menggunakan Gin-Golang, dengan fitur utama seperti CRUD pengurus, organisasi, unit anggaran, dan lainnya. Proses implementasi dilakukan secara bertahap, memastikan setiap fitur berfungsi dengan baik sesuai kebutuhan pengguna

1.6.4 Implementasi Sistem

Setelah analisis dan perancangan selesai, implementasi dimulai dengan migrasi fitur mahasiswa pada myITS Academy, seperti penetapan kelas

otomatis pada MKU, yang dilakukan secara bertahap untuk memastikan kesesuaian dengan teknologi PHP yang lama. Pada myITS Ormawa, sistem dibangun dari nol menggunakan Golang dan DDD, dimulai dengan pengembangan fitur dasar seperti manajemen pengurus dan organisasi. Proses implementasi dilakukan secara modular untuk memudahkan pengujian dan integrasi, serta memastikan setiap fitur berfungsi sesuai kebutuhan pengguna.

1.6.5 Pengujian dan Evaluasi

Setelah implementasi selesai, pengujian dilakukan untuk kedua sistem bersama tim QA DPTSI. Pada myITS Academy, pengujian memastikan hasil migrasi teknologi berjalan optimal tanpa mengurangi fungsionalitas. Pada myITS Ormawa, pengujian dilakukan untuk setiap modul, memastikan fitur seperti manajemen pengurus dan organisasi berfungsi sesuai spesifikasi.

1.6.6 Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7 Sistematika Laporan

1.7.1 Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2 Bab II Profil Perusahaan

Bab ini berisi gambaran umum DPTSI ITS mulai dari profil, lokasi perusahaan.

1.7.3 Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4 Bab IV Infrastruktur Sistem

Bab ini berisi mengenai penjelasan infrastruktur sistem yang akan digunakan dalam menyelesaikan proyek kerja praktik.

1.7.5 Bab V Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6 Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7 Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

BAB II
PROFIL PERUSAHAAN

2.1 Profil Dinas Pendidikan Provinsi Jawa Timur

Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) ITS adalah unit yang bertanggung jawab atas pengelolaan layanan teknologi informasi di Institut Teknologi Sepuluh Nopember (ITS). DPTSI mendukung berbagai aktivitas akademik, penelitian, dan manajerial guna mencapai visi dan misi institusi. Sejak didirikan pada 1982 sebagai UPT Pusat Komputer, unit ini telah berkembang pesat, dari menggunakan komputer mini hingga teknologi PC, serta mengelola ITS-net sejak 1999 untuk menghubungkan seluruh data dan informasi di ITS.

Nama unit ini telah berubah beberapa kali seiring perkembangan teknologi dan tugas yang diemban. Pada 2006, unit ini menjadi ITS ICT Services, lalu pada 2012 bergabung dengan bagian Sistem Informasi BAPSI menjadi Badan Teknologi dan Sistem Informasi (BTSI). Pada 2013, BTSI berubah nama menjadi LPTSI, yang fokus pada penelitian dan pengembangan teknologi informasi, dan pada 2016, LPTSI resmi berganti nama menjadi DPTSI. DPTSI kini fokus pada pengelolaan infrastruktur TI, pengembangan aplikasi berbasis web, serta menjaga keamanan sistem informasi ITS.

2.2 Lokasi

Institut Teknologi Sepuluh Nopember, Kampus Sukolilo, Gedung Pusat Riset Lantai 4, Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, East Java 60117

[Halaman ini sengaja dikosongkan]

BAB III
TINJAUAN PUSTAKA

3.1 Pemrograman Web

Pemrograman web merupakan proses pengembangan aplikasi yang diakses melalui jaringan internet. Meskipun proyek ini fokus pada pengembangan backend, pemahaman tentang prinsip dasar pemrograman web tetap penting. Pemrograman web melibatkan penggunaan berbagai teknologi dan bahasa pemrograman untuk mengelola logika bisnis, interaksi pengguna, dan integrasi dengan database.

3.2 Golang

Go, atau Golang, adalah bahasa pemrograman sumber terbuka yang diciptakan oleh Google pada tahun 2009. Dirancang untuk kesederhanaan, kecepatan, dan kemampuan skalabilitas tinggi, Go merupakan bahasa yang diketik dan dikompilasi secara statis, sehingga mendukung eksekusi yang cepat serta deteksi kesalahan lebih awal. Fitur unggulan Go termasuk dukungan untuk konkurensi lewat goroutine dan channel, yang memudahkan pengelolaan banyak proses secara bersamaan, serta adanya garbage collector otomatis yang mengurangi beban pengelolaan memori. Dengan sintaks yang sederhana namun kuat, terinspirasi oleh bahasa C, Go mudah dipahami dan dipakai, sementara pustaka standarnya yang lengkap mengurangi ketergantungan pada alat eksternal. Banyak digunakan dalam pengembangan aplikasi web, layanan cloud, dan arsitektur mikro, Go telah diterima oleh berbagai perusahaan besar seperti Google, Dropbox, dan Uber, menunjukkan bahwa bahasa ini dapat diandalkan dalam penggunaan di lingkungan produksi.

3.3 Arsitektur Modular Monolitik

Arsitektur Modular Monolitik menggabungkan konsep monolitik tradisional (aplikasi sebagai satu unit terintegrasi) dengan pendekatan modular untuk meningkatkan keterbacaan dan kemudahan pemeliharaan. Dalam arsitektur ini, aplikasi dibagi menjadi modul-modul fungsional (seperti controller, model, dan service) yang terpisah secara logika, meskipun tetap berjalan dalam satu basis kode. Contoh implementasinya adalah struktur MVC (Model-View-Controller), di mana setiap layer memiliki tanggung jawab spesifik. Pendekatan ini cocok untuk proyek menengah yang membutuhkan skalabilitas terbatas tanpa kompleksitas infrastruktur microservices.

3.4 Framework Gin-Golang

Framework Gin-Golang adalah salah satu framework web paling populer di ekosistem Golang, dikenal karena performanya yang tinggi dan kemudahan penggunaan. Gin menyediakan fitur esensial seperti routing dinamis, middleware untuk autentikasi atau logging, serta dukungan validasi request. Framework ini sering dipadukan dengan GORM untuk operasi database, memungkinkan pengembang membangun API RESTful dengan cepat. Contoh implementasi routing di Gin melibatkan definisi endpoint dan handler fungsi, sementara middleware seperti CORS (Cross-Origin Resource Sharing) dapat diintegrasikan untuk keamanan komunikasi antar-domain.

3.5 GORM

GORM (Go Object-Relational Mapping) adalah library ORM untuk Golang yang mempermudah interaksi dengan database relasional seperti MySQL, PostgreSQL,

atau SQLite. Dengan GORM, pengembang dapat melakukan operasi CRUD (Create, Read, Update, Delete) menggunakan sintaksis Go, tanpa perlu menulis query SQL manual. Fitur unggulannya meliputi auto-migrasi skema database, relasi antar-tabel, dan transaksi yang aman. Integrasi GORM dengan Gin-Golang memungkinkan pembuatan aplikasi web yang terstruktur, di mana model data didefinisikan sekali dan digunakan secara konsisten di seluruh layer aplikasi. Kombinasi Gin dan GORM menjadi solusi efisien untuk proyek yang memprioritaskan kecepatan pengembangan dan maintainability.

3.6 CockroachDB

CockroachDB adalah sistem database relasional terdistribusi yang dirancang untuk skalabilitas horizontal dan ketahanan terhadap kegagalan. Database ini kompatibel dengan PostgreSQL, memungkinkan migrasi mudah dari sistem SQL tradisional. Keunggulan utamanya terletak pada kemampuan replikasi data otomatis di seluruh node, konsistensi kuat (ACID compliance), dan dukungan untuk transaksi multi-region. Cocok untuk aplikasi yang membutuhkan ketersediaan tinggi dan toleransi kesalahan, seperti sistem finansial atau layanan global. Integrasi dengan Golang dapat dilakukan melalui driver database standar atau ORM seperti GORM.

3.7 SQL Server DB

SQL Server DB merupakan sistem manajemen basis data relasional (RDBMS) yang dikembangkan oleh Microsoft. Sistem ini dikenal karena kemampuannya dalam menyimpan, mengelola, dan mengambil data secara

efisien, serta mendukung transaksi kompleks dengan fitur ACID (Atomicity, Consistency, Isolation, Durability). SQL Server menyediakan alat keamanan canggih seperti enkripsi data, autentikasi terintegrasi, dan kontrol akses granular, menjadikannya pilihan populer untuk aplikasi enterprise. Selain itu, SQL Server mendukung integrasi dengan berbagai platform dan bahasa pemrograman melalui teknologi seperti T-SQL, .NET, dan layanan analitik (SSAS, SSIS, SSRS). Skalabilitasnya yang baik memungkinkan pengembangan aplikasi dari skala kecil hingga besar dengan dukungan cloud melalui Azure SQL Database

3.8 Clean Architecture

Clean Architecture adalah pendekatan desain perangkat lunak yang menekankan pemisahan tanggung jawab (separation of concerns) melalui struktur lapisan modular. Konsep ini memisahkan domain bisnis inti dari infrastruktur eksternal (seperti UI, basis data, atau framework) untuk meningkatkan kemandirian dan kejelasan kode. Prinsip utamanya adalah ketergantungan yang mengarah ke dalam (dependency rule), di mana lapisan luar bergantung pada lapisan dalam, tetapi tidak sebaliknya. Hal ini memudahkan pengujian, pemeliharaan, dan adaptasi teknologi tanpa mengganggu logika bisnis. Clean Architecture cocok untuk proyek jangka panjang yang memprioritaskan fleksibilitas dan mengurangi risiko perubahan kebutuhan teknis.

3.9 Command Query Responsibility Segregation (CQRS)

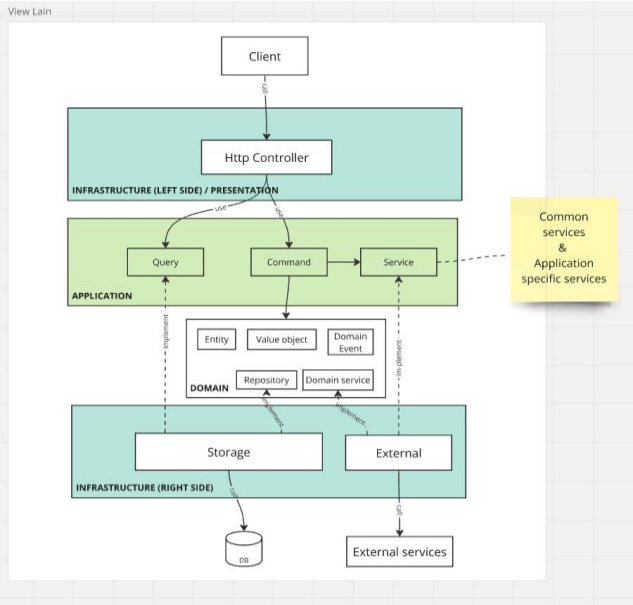
CQRS adalah pola arsitektur yang memisahkan operasi baca (query) dan tulis (command) dalam sistem perangkat lunak. Dengan membedakan model untuk membaca dan mengubah data, CQRS memungkinkan optimasi performa serta skalabilitas, terutama pada aplikasi dengan beban asinkron tinggi. Pola ini sering dipadukan dengan Event Sourcing untuk melacak perubahan status sistem secara historis. Meskipun meningkatkan kompleksitas implementasi, CQRS efektif diterapkan dalam skenario di mana kebutuhan query dan command memiliki karakteristik berbeda (misalnya: sistem dengan update jarang tetapi pembacaan intensif). Keuntungan utamanya termasuk fleksibilitas dalam mengelola konsistensi data dan kemampuan untuk menyesuaikan infrastruktur sesuai kebutuhan operasi spesifik.

[Halaman ini sengaja dikosongkan]

BAB IV
ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM

4.1 Analisis Sistem

Pada bab ini akan dijelaskan mengenai Infrastruktur yaitu mengenai design sistem backend dan fungsionalitas pada Modul MKU di myITS Academics dan myITS SIM Ormawa yaitu analisis dari infrastruktur sistem yang akan dibangun. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum aplikasi dan analisis kebutuhan.



Gambar 4.1 Aliran arsitektur pada sistem

4.1.1 Definisi Umum Aplikasi

myITS adalah sistem informasi akademik secara umum digunakan untuk mengakomodasi segala kegiatan mahasiswa dan Masyarakat ITS. myITS Academics adalah salah satu bagian dari myITS yang mengurus segala tentang mahasiswa, nilai, dan dosen berhubungan erat dengan sistem Belajar mengajar dalam ITS dan myITS Ormawa mengurus sistem dan management organisasi Mahasiswa di ITS.

4.2 Perancangan Infrastruktur Sistem

4.2.1 Desain Sistem

Arsitektur yang digunakan dalam backend setiap bagian dari myITS seperti myITS Academics maupun myITS Ormawa (*Organization*) menggunakan pendekatan arsitektur modular monolitik untuk menyimbangkan kemudahan untuk pengelolaan dan fleksibilitas. Dalam myITS Ormawa sendiri hanya memiliki satu modul. Walaupun dibangun dalam satu kesatuan monolitik, struktur internalnya mengadopsi prinsip modular seperti *microservices*, dimana setiap service dikemas dalam modul independent. Pendekatan ini meminimalisir komplekstias infrastruktur dibanding arsitektur microservices murni. penerapan *Clean Architecture* dalam setiap modul, yang mengisolasi logika bisnis inti dari dependensi eksternal seperti basis data atau antarmuka pengguna dan juga mengadopsi pola *Command Query Responsibility Segregation* (CQRS) untuk memisahkan operasi pembacaan data (*read / query*) dan perubahan data (*write / command*) untuk memastikan optimasi performa serta konsistensi data tanpa mengorbankan skalabilitas.

Dari sisi implementasi, aplikasi backend dikembangkan menggunakan Golang untuk

mengoptimalkan kecepatan eksekusi dan efisiensi resource. Framework Gin dimanfaatkan untuk menangani manajemen request HTTP secara ringan, sementara GORM berperan sebagai jembatan antara objek aplikasi dan basis data yang digunakan seperti CockroachDB dan SQL Server. Untuk myITS Academics sendiri menggunakan CockroachDB dipilih sebagai database utama karena fitur clustering-nya yang mendukung data terdistribusi, sehingga memastikan ketersediaan dan konsistensi data.

Pada lapisan infrastruktur, HTTP Controller bertindak sebagai entry point yang mengarahkan permintaan ke query handler atau command handler sesuai jenis operasi. Lapisan query fokus pada pengambilan data melalui storage layer (*repository*) tanpa mengubah informasi pada DB transaction, sedangkan command menangani perubahan data dengan melibatkan domain layer untuk validasi logika bisnis. Integrasi antar-layer dirancang dengan ketergantungan satu arah (dependency inversion) untuk memudahkan pengujian dan modifikasi. Dengan struktur ini, sistem tetap responsif terhadap perubahan kebutuhan teknis maupun bisnis, sekaligus menjaga stabilitas melalui isolasi risiko pada modul tertentu.

4.2.2 Fungsionalitas myITS Academics Modul MKU dan myITS Ormawa (*Organization*)

myITS Academics merupakan transformasi digital dari sistem SIAKAD (Sistem Informasi Akademik) ITS yang telah menjadi tulang punggung operasional akademik sejak 2005. Jika SIAKAD berfokus pada fungsi dasar seperti pengelolaan KRS (Kartu Rencana Studi), nilai, dan jadwal, myITS Academics memperluas cakupan dengan integrasi fitur yang lebih holistik dan responsif terhadap kebutuhan modern.

Evolusi ini mencakup penggabungan layanan terdistribusi ke dalam platform terpusat, peningkatan antarmuka pengguna berbasis role (mahasiswa, dosen, admin), serta otomatisasi proses administratif yang sebelumnya manual. Fitur-fitur baru seperti analitik kinerja akademik, real-time reporting, dan integrasi dengan sistem eksternal (e.g., learning management systems) menjadi pembeda utama, mendukung transparansi dan efisiensi institusi.

Seperti yang sudah di bilang sebelumnya myITS Academics terdiri dari beberapa modul yang mempunyai tujuan dan fungsi nya masing-masing. Modul yang dikerjakan dalam Kerja Praktik ini adalah Modul MKU (Mata Kuliah Umum). Modul ini berfungsi utama menjadi Modul yang mendistribusikan kelas mata kuliah umum kepada Kelompok-kelompok bidang studi, secara otomatis assign mahasiswa dan membuat beberapa kelas. Di Modul ini mahasiswa tidak perlu melakukan apa-apa dan yang memegang kendali adalah Admin dan Kepengurusan yang mengurus MKU. Admin dan pengurus dapat membuat Kelompok bidang studi pada setiap mata kuliah, dapat membuat kelas secara otomatis dari Kelompok yang sudah dibuat, dan terakhir dapat assign mahasiswa dari setiap Kelompok bidang studi tersebut secara otomatis juga.

Pada bagian kedua yaitu myITS Ormawa sendiri saat buku ini dituliskan fungsionalitas yang dibuat adalah fungsionalitas yang hanya dapat diakses oleh admin myITS Ormawa seperti, CRUD Ormawa, jenis ormawa, skala ormawa, CRUD Unit Anggaran, dan CRUD Anggaran Organisasi.

[Halaman ini sengaja dikosongkan]

BAB V
IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi pengembangan dari sistem Backend API menggunakan Golang untuk mendukung operasional modul MKU pada myITS Academics dan myITS Ormawa. Implementasi ini secara garis besar dibagi menjadi dua bagian utama yaitu myITS Academics modul MKU dan myITS Ormawa.

5.1 myITS Academics modul MKU

Implementasi pada modul MKU di myITS akademik yang terdiri atas beberapa API utama yaitu seperti ini:

5.1.1 Melihat List Kurikulum MKU

Mendapatkan list dari kurikulum yang ada dalam suatu tahun ajaran tertentu.

```
// GET LIST KURIKULUM MKU
1: Authenticate user (role: admin/authorized)
2: If authentication fails, return error

3: Validate input parameter 'tahun' (integer, required)
4: If invalid, return error

5: Query database:
   - JOIN tables: matkul_kurikulum, kurikulum, mata_kuliah
   - SELECT: id_mk, kode_mk, nama, nama_en, sks_mk
   - FILTER:
     - tahun_kurikulum = input 'tahun'
     - id_bidang_studi = "9d79b848..." (hardcoded MKU)

6: If query returns ErrRecordNotFound:
   - Return empty list + custom error "Tidak ada kurikulum MKU"

7: If other database errors:
```

```
   - Return error message with details

8: Map raw database records to KurikulumMkuList struct:
   - Loop tiap record
   - Assign: id, kode, nama, nama_en, sks

9: Return mapped list + nil error
```

Kode Sumber 5.1. Pseudo code API melihat list kurikulum MKU

5.1.2 CRUD Kelompok dan Anggota Bidang Studi dalam Kelompok MKU

Kelompok MKU adalah Kelompok yang terdiri dari beberapa bidang studi (jurusan) yang nantinya mahasiswa dalam jurusan tersebut akan secara otomatis di distribusikan kedalam beberapa kelas dengan acuan kelompok MKU tersebut, berikut adalah implementasinya:

```
// GET KELOMPOK MKU BERDASARKAN ID MK
1: Authenticate user (role: admin/authorized)
2: If authentication fails, return error

3: Validate input parameter 'idMk' (string, required)
4: If invalid, return error

5: Query database kelompok_mku:
   - JOIN tabel: bidang_studi, matkul_kurikulum, mata_kuliah,
mahasiswa
   - SELECT:
     * smt, sks_mk, nama bidang studi, id_mk, kode_mk
     * COUNT mahasiswa yang memenuhi syarat agama dan angkatan
   - FILTER:
     * id_mk = input 'idMk'
     * Exclude prodi dengan id_siakad_prodi_lama = '__TPB'
   - GROUP BY smt, sks_mk, nama bidang studi, id_mk

6: If query returns ErrRecordNotFound:
```

```
   - Return empty list + custom error "Tidak ada kelompok MKU"

7: If other database errors:
   - Return error message dengan detail

8: Map hasil query ke struktur KelompokMku:
   - Untuk tiap kelompok:
     a. Konversi SKS dari string ke integer
     b. Hitung jumlah mahasiswa dari COUNT
     c. Susun nested structure BidangStudi

9: Return mapped list + nil error

// BUAT KELOMPOK BARU
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Generate new UUID for kelompok_id
4: Validate input parameters:
   - id_matkul_kurikulum (required string)
   - nama_kelompok (required string)
5: If invalid, return error

6: Create Kelompok entity:
   - id_kelompok = generated UUID
   - id_matkul = validated id_matkul
   - nama = nama_kelompok
7: If entity creation fails, return error

8: Save entity to database via repository
9: If save fails, return error

10: Return generated kelompok_id

// UPDATE KELOMPOK
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameters:
   - kelompok_id (required string)
   - nama_kelompok (required string)
```

```
4: If invalid, return error

5: Query existing kelompok by kelompok_id
6: If not found, return error

7: Create updated kelompok entity:
   - id = existing kelompok_id
   - id_matkul = existing id_matkul (tidak diubah)
   - nama = new nama_kelompok
8: If entity creation fails, return error

9: Save updated entity to database
10: If update fails, return error

11: Return success

// HAPUS KELOMPOK
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameter kelompok_id (required string)
4: If invalid, return error

5: Query kelompok from database by kelompok_id
6: If not found, return error

7: Delete kelompok from database
8: If delete fails, return error

9: Return success

// TAMBAH PRODI KE KELOMPOK
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameters:
   - id_kelompok (required string)
   - id_bidang_studi (required string)
4: If invalid, return error

5: Generate composite ID dari id_kelompok + id_bidang_studi
```

```
6: If ID generation fails, return error

7: Create KelompokProdi entity:
   - composite_id = generated ID
   - (additional fields jika ada)
8: If entity creation fails, return error

9: Save entity ke database
10: If save fails, return error

11: Return success

// HAPUS PRODI DARI KELOMPOK
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameters:
   - id_kelompok (required string)
   - id_bidang_studi (required string)
4: If invalid, return error

5: Query KelompokProdi by composite ID (id_kelompok +
id_bidang_studi)
6: If not found, return error

7: Delete entity dari database
8: If delete fails, return error

9: Return success
```

Kode Sumber 5.2. Pseudo code API Kelompok Bidang Studi MKU

Konfigurasi lengkap website ppdbjatim.net di NGINX dapat dilihat pada Lampiran 3.

5.1.3 API Lihat detail kelas MKU (Mata Kuliah Umum)

Admin dapat melihat detail kelas MKU yang sudah ada termasuk melihat dosen yang mengajar dan mahasiswa yang tergabung di dalamnnya.

```
// GET DETAIL KELAS MKU
1: Authenticate user (role: admin/dosen/mahasiswa)
2: If authentication fails, return error

3: Validate input parameter kelas_id (required string)
4: If invalid, return error

5: Query utama detail kelas:
   - JOIN tabel: kelas, bidang_studi, mata_kuliah, kurikulum,
jenis_matkul
   - SELECT:
     * info kelas (id, nama, sks, daya tampung)
     * info mata kuliah (kode, nama, SKS)
     * info prodi (id_bidang_studi, nama prodi)
   - FILTER: id_kelas = input kelas_id

6: If query error, return error

7: Query dosen pengajar:
   - JOIN tabel: akt_ajar, pegawai
   - SELECT:
     * data dosen (id, nama, gelar, NIDN)
     * info pengajaran (urutan, sks ajar)
   - FILTER: id_kelas = input kelas_id
   - ORDER BY urutan

8: If query error, return error

9: Query prodi lain yang memiliki kuota:
   - JOIN tabel: kuota_kelas, unit_organisasi
   - SELECT:
     * info prodi (id, nama, kode)
     * kuota dan terisi
```

```
   - FILTER: id_kelas = input kelas_id

10: If query error, return error

11: Combine semua hasil query ke response object:
    - Detail kelas
    - Dosen pengajar
    - Prodi lain dengan kuota

12: Return combined response
```

Kode Sumber 5.3. Pseudo code API melihat detail kelas MKU

5.1.4 API Rencana MKU

Ini adalah bagian utama dari bagian yang saya kerjakan, yaitu perencanaan MKU dimana admin dapat secara otomatis merencanakan MKU, assign mahasiswa berdasarkan Kelompok Prodi yang telah ada. Membuat kelas, dan finalisasi menjadi FRS, dan Rollback (membatalkan rencana).

```
// BUAT RENCANA PESERTA MKU
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameters:
   - smt_id (required)
   - mk_id (required)
   - ukuran_kelas (integer > 0)
4: If invalid, return error

5: Get mata kuliah details by mk_id
6: If error (e.g., not found), return error

7: Delete existing rencana peserta for smt_id + mk_id
8: If delete fails, return error
```

```
9: Get bidang studi terkait mata kuliah dan semester
10: If error, return error

11: Get semua kelompok untuk mk_id
12: If error, return error

13: Initialize mapping peserta per kelompok & bidang studi:
    - Untuk tiap kelompok:
      a. Buat entry untuk setiap bidang studi di kelompok
tersebut
      b. Inisialisasi array peserta kosong

14: Untuk tiap bidang studi yang relevan:
    a. Get mahasiswa yang memenuhi syarat:
       - Semester masuk sesuai
       - Bidang studi match
       - Filter agama (jika MK agama)
    b. Jika error, return error

    c. Assign mahasiswa ke kelompok dan bidang studi:
       - Loop semua kelompok
       - Cari bidang studi yang sesuai
       - Tambahkan mahasiswa ke mapping

    d. Sort peserta per bidang studi berdasarkan NRP

15: Generate distribusi kelas:
    - Untuk tiap kelompok:
      a. Hitung total peserta di semua bidang studi kelompok
      b. Jika 0 peserta, skip
      c. Hitung jumlah kelas: total_peserta / ukuran_kelas (min
1)
      d. Distribusi mahasiswa ke kelas dengan round-robin:
         - Setiap kelas ke-i mengambil mahasiswa indeks i,
i+kelas, i+2*kelas, dst.

16: Simpan rencana peserta ke database
17: If insert error, return error

18: Return success
```

```
// BUAT KELAS DARI RENCANA PESERTA
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameters:
   - mk_id (required)
   - smt_id (required)
   - user_id (required)
4: If invalid, return error

5: Start database transaction

6: Update daya tampung kelas yang sudah ada:
   - Hitung jumlah peserta dari tabel rencana_mku
   - Update daya_tampung dan updater di tabel kelas

7: Insert kelas baru yang belum ada:
   - Generate UUID untuk id_kelas
   - Ambil data dari rencana_mku (kelas yang belum terdaftar)
   - Set id_bidang_studi ke hardcoded MKU ID
   - Hitung daya_tampung berdasarkan jumlah peserta
   - Insert ke tabel kelas

8: Hapus kuota kelas yang sudah tidak relevan:
   - Cari kelas yang peserta rencana_mku-nya sudah dihapus
   - Delete entri di tabel kuota_kelas

9: Insert kuota kelas baru:
   - Hitung kuota per prodi dari data mahasiswa di rencana_mku
   - Insert ke tabel kuota_kelas untuk kombinasi kelas + prodi
yang belum ada

10: Commit transaction
11: If any step fails, rollback dan return error

12: Return success

// FINALISASI FRS
1: Authenticate user (role: admin)
2: If authentication fails, return error
```

```
3: Validate input parameters:
   - mk_id (required)
   - smt_id (required)
   - user_id (required)
4: If invalid, return error

5: Start database transaction

6: Log perubahan kelas di antrian_frs:
   - Insert antrian untuk mahasiswa yang pindah kelas
   - Hanya untuk kuliah tanpa nilai yang kelasnya berbeda dengan
rencana

7: Hapus kuliah yang tidak sesuai rencana:
   - Delete kuliah yang kelasnya berbeda dan belum ada nilai

8: Buat FRS baru untuk mahasiswa yang belum punya:
   - Generate FRS dengan nilai default (SKS=0, IPS=0)
   - Hanya untuk mahasiswa tanpa FRS di semester tersebut

9: Log penambahan kelas di antrian_frs:
   - Insert antrian untuk penambahan kelas baru sesuai rencana

10: Assign mahasiswa ke kelas di tabel kuliah:
    - Ambil data dari rencana_mku
    - Pastikan tidak ada duplikasi assign

11: Update jumlah terisi kelas:
    - Hitung total mahasiswa per kelas
    - Update field jml_terisi di tabel kelas

12: Update kuota terisi per prodi:
    - Hitung mahasiswa per prodi di tiap kelas
    - Update field terisi di tabel kuota_kelas

13: Commit transaction
14: If any step fails:
    - Rollback transaction
    - Return error dengan detail

15: Return success
```

```
// BATAL RENCANA MKU
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameters:
   - mk_id (required)
   - smt_id (required)
4: If invalid, return error

5: Start database transaction

6: Hapus data terkait secara berurutan:
   a. Antrian FRS yang terkait kelas MKU
   b. Data kuliah (FRS) mahasiswa
   c. Kuota kelas per prodi
   d. Data aktivitas mengajar dosen (akt_ajar)
   e. Data kelas itu sendiri
   f. Data rencana peserta awal

7: Untuk tiap operasi hapus:
   - Gunakan CTE untuk identifikasi kelas yang terkait mk_id +
smt_id
   - Hapus data child sebelum parent untuk jaga referential
integrity
   - Gunakan hardcoded bidang_studi MKU ID

8: Jika semua operasi sukses:
   - Commit transaction
   - Return success

9: Jika ada error:
   - Rollback transaction
   - Return error dengan detail operasi yang gagal

Status Operasi:
- Sukses: Seluruh data terkait terhapus
- Gagal: Transaksi rollback, data tetap konsisten
```

Kode Sumber 5.4. Pseudo code API buat rencana, buat kelas, buat FRS, dan batal MKU

5.1.5 API Lihat Rencana Peserta

Admin dapat melihat Rencana Peserta dengan semua peserta yang ada didalam.

```
// GET RENCANA PESERTA
1: Authenticate user (role: admin/dosen)
2: If authentication fails, return error

3: Validate input parameters:
   - mk_id (required)
   - smt_id (required)
4: If invalid, return error

5: Query data rencana peserta:
   - JOIN tabel: rencana_mku, mata_kuliah, kelas, kuliah
(subquery)
   - SELECT:
     * Nama matkul, kode, nama kelas
     * Jumlah peserta rencana (count mahasiswa)
     * Daya tampung kelas
     * Jumlah peserta FRS (count dari kuliah)
   - FILTER:
     * id_mk = input mk_id
     * id_smt = input smt_id
     * bidang_studi = hardcoded MKU ID
   - GROUP BY: id_mk, id_smt, nama_kelas

6: If query returns no data:
   - Return empty list + error "Tidak ada rencana"

7: Map hasil query ke response object:
   - Tambahkan field Jenis = "Paket" (hardcoded)
   - Konversi tipe data jika diperlukan

8: Return list rencana peserta
```

Kode Sumber 5.5. Pseudo code API melihat rencana peserta MKU

5.1.6 API Lihat State dari Rencana MKU

Admin dapat melihat State saat ini dari Perencanaan MKU (rencana peserta, kelas, frs) terhadap suatu MKU tertentu

```
// CEK STATUS RENCANA PESERTA MKU
1: Authenticate user (role: admin/dosen)
2: If authentication fails, return error

3: Validate input parameters:
   - mk_id (required)
   - smt_id (required)
4: If invalid, return error

5: Check Finalisasi FRS:
   - Query jumlah mahasiswa yang sudah difinalisasi (kuliah)
   - JOIN rencana_mku + kelas + kuliah
   - Filter bidang_studi = MKU ID
   - Jika jumlah > 0 → return status 3 (Sudah FRS)

6: Check Rencana vs Kelas Aktif:
   - Query jumlah rencana yang sudah punya kelas
   - JOIN rencana_mku + kelas (nama kelas match)
   - Filter bidang_studi = MKU ID
   - Jika jumlah > 0 → return status 2 (Sudah Generate Kelas)

7: Check Rencana Exists:
   - Query total rencana peserta tanpa kelas
   - Jika jumlah > 0 → return status 1 (Ada Rencana Belum
Generate)

8: Default:
   - Return status 0 (Tidak Ada Rencana)

Status Code:
0 = Tidak ada rencana
1 = Ada rencana mentah
2 = Sudah generate kelas
```

```
3 = Sudah final FRS
```

Kode Sumber 5.6. Pseudo code API melihat state rencana MKU

5.2 myITS Ormawa (*Organization*)

ITS Ormawa (*Organization*) adalah aplikasi web yang mengatur organisasi-organisasi mahasiswa yang ada di ITS. Berikut adalah kode yang digunakan:

5.2.1 CRUD Ormawa

Implementasi CRUD untuk entity ormawa:

```
// BUAT ORMAWA BARU
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input:
   - id_skala_ormawa (required)
   - id_jenis_ormawa (required)
   - nama (required, unique)
4: Jika invalid, return error

5: Cek keberadaan Skala Ormawa:
   - Query by id_skala_ormawa
   - Jika tidak ada, return error "Skala tidak ditemukan"

6: Cek keberadaan Jenis Ormawa:
   - Query by id_jenis_ormawa
   - Jika tidak ada, return error "Jenis tidak ditemukan"

7: Cek duplikasi nama:
   - Query nama di repo
   - Jika sudah ada, return error "Nama sudah terdaftar"

8: Generate UUID sebagai id_ormawa baru

9: Buat entity Ormawa:
   - id = generated UUID
   - nama = input nama
```

```
   - is_olahraga = input boolean
   - skala = entity Skala
   - jenis = entity Jenis
   - version = 1

10: Simpan entity ke repository
11: Jika gagal simpan, return error "Gagal menyimpan data"

12: Return id_ormawa baru

// EDIT ORMAWA
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input:
   - id_ormawa (required)
   - id_skala_ormawa (required)
   - id_jenis_ormawa (required)
   - nama (required, unique)
4: Jika invalid, return error

5: Cek keberadaan Ormawa:
   - Query by id_ormawa
   - Jika tidak ada, return error "Ormawa tidak ditemukan"

6: Cek keberadaan Skala Ormawa:
   - (Langkah sama seperti Buat Ormawa)

7: Cek keberadaan Jenis Ormawa:
   - (Langkah sama seperti Buat Ormawa)

8: Cek duplikasi nama (kecuali untuk record yang sama):
   - Query nama dengan exclude id_ormawa saat ini
   - Jika ada, return error "Nama sudah digunakan"

9: Update entity Ormawa:
   - nama = input baru
   - is_olahraga = input baru
   - skala = entity Skala baru
   - jenis = entity Jenis baru
   - version += 1 (optimistic lock)
```

```
10: Simpan perubahan ke repository
11: Jika gagal update, return error "Gagal memperbarui data"

12: Return success

// HAPUS ORMAWA
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validate input parameter id_ormawa (required)
4: If invalid, return error "ID tidak valid"

5: Cek keberadaan ormawa:
   - Query by id_ormawa
   - Jika tidak ada, return error "Ormawa tidak ditemukan"

6: Lakukan soft delete:
   - Set deleted_at = waktu sekarang
   - Jika gagal, return error "Gagal menghapus data"

7: Return success

// GET LIST ORMAWA
1: Authenticate user (role: admin/user)
2: If authentication fails, return error

3: Build base query:
   - SELECT ormawa.*, skala.nama as skala, jenis.nama as jenis
   - LEFT JOIN skala_ormawa dan jenis_ormawa
   - Filter deleted_at IS NULL (hanya data aktif)

4: Apply filters:
   a. Jika ada filter nama:
      - WHERE nama LIKE '%{search}%'
   b. Jika ada filter skala:
      - WHERE skala.nama = {value}

5: Execute query
6: Jika error, return error
```

```
7: Map hasil ke response:
   - Convert UUID ke format string
   - Tentukan status aktif (is_aktif) berdasarkan deleted_at
   - Include nama skala dan jenis dari hasil JOIN

8: Return list ormawa
```

Kode Sumber 5.7. Pseudo code API CRUD Ormawa

5.2.2 CRUD Unit Anggaran

Implementasi CRUD untuk entity unit anggaran:

```
// BUAT UNIT ANGGARAN
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input:
   - nama (required, unique)
4: Jika invalid, return error "Input tidak valid"

5: Generate ID baru via repository
6: Jika gagal generate ID, return error

7: Cek duplikasi nama:
   - Query by nama
   - Jika sudah ada, return error "Nama sudah terdaftar"

8: Buat entity baru:
   - id = generated ID
   - nama = input nama
   - created_at = waktu sekarang
   - updated_at = waktu sekarang
   - version = 1

9: Simpan entity ke repository
10: Jika gagal simpan, return error "Gagal menyimpan data"

11: Return ID baru

// EDIT UNIT ANGGARAN
1: Authenticate user (role: admin)
```

```
2: If authentication fails, return error

3: Validasi input:
   - id (required)
   - nama (required, unique)
4: Jika invalid, return error

5: Cek keberadaan unit:
   - Query by id
   - Jika tidak ada, return error "Data tidak ditemukan"

6: Cek duplikasi nama (exclude current id):
   - Query nama dengan filter id != current
   - Jika ada, return error "Nama sudah digunakan"

7: Update entity:
   - nama = input baru
   - created_at = tetap dari data lama
   - updated_at = waktu sekarang
   - version = version lama + 1

8: Simpan perubahan ke repository
9: Jika gagal update, return error "Gagal memperbarui data"

10: Return success

// HAPUS UNIT ANGGARAN
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input parameter id (required)
4: Jika invalid, return error "ID tidak valid"

5: Cek keberadaan unit:
   - Query by id
   - Jika tidak ada, return error "Data tidak ditemukan"

6: Hapus data dari repository
7: Jika gagal hapus, return error "Gagal menghapus data"

8: Return success
```

```
// GET LIST UNIT ANGGARAN
1: Authenticate user (role: admin/user)
2: If authentication fails, return error

3: Build base query:
   - SELECT semua field dari tabel unit_anggaran_organisasi

4: Apply filters:
   - Jika ada parameter nama: WHERE nama LIKE '%{filter}%'

5: Execute query
6: Jika error, return error

7: Map hasil ke response:
   - Konversi format ID jika diperlukan
   - Hitung is_aktif: expired_at > waktu sekarang

8: Return list unit anggaran
```

Kode Sumber 5.8. Pseudo code API CRUD Unit Anggaran

5.2.3 CRUD Anggaran Organisasi

Implementasi CRUD untuk entity anggaran organisasi:

```
// BUAT ANGGARAN ORGANISASI
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input:
   - unit_anggaran_id (required)
   - tahun_anggaran (required, numeric)
   - pagu (required, numeric)
   - jenis (required, enum)
4: Jika invalid, return error

5: Generate ID baru via repository
6: Jika gagal generate ID, return error

7: Buat entity baru:
```

```
   - id = generated ID
   - unit_anggaran_id = valid ID
   - tahun = input tahun
   - pagu = input pagu
   - jenis = input jenis
   - created_at/updated_at = waktu sekarang
   - version = 1

8: Simpan entity ke repository
9: Jika gagal simpan, return error

10: Return ID baru

// EDIT ANGGARAN ORGANISASI
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input:
   - id (required)
   - unit_anggaran_id (required)
   - tahun_anggaran (required)
   - pagu (required)
   - jenis (required)
4: Jika invalid, return error

5: Cek keberadaan data:
   - Query by id
   - Jika tidak ada, return error "Data tidak ditemukan"

6: Update entity:
   - unit_anggaran_id = input baru
   - tahun = input baru
   - pagu = input baru
   - jenis = input baru
   - created_at = tetap
   - updated_at = waktu sekarang
   - version += 1

7: Simpan perubahan ke repository
8: Jika gagal update, return error
```

```
9: Return success

// HAPUS ANGGARAN ORGANISASI
1: Authenticate user (role: admin)
2: If authentication fails, return error

3: Validasi input parameter id (required)
4: Jika invalid, return error

5: Cek keberadaan data:
   - Query by id
   - Jika tidak ada, return error "Data tidak ditemukan"

6: Hapus data dari repository
7: Jika gagal hapus, return error

8: Return success

// GET LIST ANGGARAN ORGANISASI
1: Authenticate user (role: admin/user)
2: If authentication fails, return error

3: Build base query:
   - SELECT ao.*, ua.* (join dengan unit_anggaran)
   - Preload relasi unit_anggaran

4: Apply filters:
   a. Unit Anggaran: Partial match nama unit
   b. Tahun: Exact match
   c. Pagu: Exact match
   d. Jenis: Exact match

5: Execute query
6: Jika error, return error

7: Map hasil ke response:
   - Hitung is_aktif berdasarkan expired_at unit_anggaran
   - Include detail unit_anggaran dalam response

8: Return list anggaran
```

Kode Sumber 5.9. Pseudo code API CRUD Anggaran Organisasi

[Halaman ini sengaja dikosongkan]

BAB VI
PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap Backend API modul MKU pada myITS Academics dan myITS Ormawa. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan analisis dan perancangan arsitektur yang sudah dirancangkan.

6.1 Tujuan Pengujian

Pengujian dilakukan terhadap Backend API modul MKU pada myITS Academics dan myITS Ormawa guna menguji kemampuan arsitektur dalam melayani permintaan sistem aplikasi.

6.2 Kriteria Pengujian

Pencapaian tujuan pengujian diukur dengan menilai hasil dari beberapa kriteria berikut:

a. Kemampuan backend API dalam merespons setiap permintaan.
b. Kemampuan backend API untuk menyimpan data yang diinput melalui aplikasi ke dalam database.
c. Kemampuan backend API dalam menampilkan data sesuai dengan permintaan pengguna.

6.3 Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai user yang akan menjalankan fitur-fitur. Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut :

1. Admin Akademik dapat membuka daftar kurikulum MKU (Mata Kuliah Umum) untuk melihat mata kuliah yang tersedia beserta detailnya (kode, SKS, deskripsi).

2. Admin Akademik dapat membuat, mengubah, atau menghapus kelompok prodi dan dapat menambahkan atau menghapus prodi dari kelompok prodi tertentu.
3. Dosen/Admin dapat melihat daftar peserta kelas berdasarkan kode kelas atau filter prodi/semester.
4. Admin Akademik dapat membuat kelas secara otomatis berdasarkan rencana peserta yang telah diassign.
5. Admin Akademik dapat memfinalisasi kelas menjadi FRS.
6. Admin Akademik dapat membatalkan seluruh progres rencana (asign peserta, generate kelas, FRS.
7. Ormawa: Admin dapat melakukan CRUD Ormawa
8. Ormawa: Admin dapat melakukan CRUD Unit Anggaran
9. Ormawa: Admin dapat melakukan CRUD Unit Anggaran

6.4 Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi PPDB terhadap kasus skenario uji coba. Tabel 6.1 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

Tabel 6.1. Hasil Evaluasi Pengujian

Kriteria Pengujian	**Hasil Pengujian**
Admin Akademik dapat membuka daftar kurikulum MKU (Mata Kuliah Umum)	Terpenuhi

untuk melihat mata kuliah yang tersedia beserta detailnya (kode, SKS, deskripsi).	
Admin Akademik dapat membuat, mengubah, atau menghapus kelompok prodi dan dapat menambahkan atau menghapus prodi dari kelompok prodi tertentu.	Terpenuhi
Dosen/Admin dapat melihat daftar peserta kelas berdasarkan kode kelas atau filter prodi/semester	Terpenuhi
Admin Akademik dapat membuat kelas secara otomatis berdasarkan rencana peserta yang telah diassign	Terpenuhi
Admin Akademik dapat memfinalisasi kelas menjadi FRS	Terpenuhi
Admin Akademik dapat membatalkan seluruh progres rencana (asign peserta, generate kelas, FRS.	Terpenuhi
Ormawa: Admin dapat melakukan CRUD Ormawa	Terpenuhi
Ormawa: Admin dapat melakukan CRUD Unit	Terpenuhi

Anggaran	
Ormawa: Admin dapat melakukan CRUD Anggaran Organisasi	Terpenuhi

BAB VII
KESIMPULAN DAN SARAN

7.1 Kesimpulan

Kesimpulan yang didapat setelah melakukan pengembangan terhadap Backend API modul MKU pada myITS Academics dan myITS Ormawa adalah sebagai berikut :

a. Arsitektur sistem yang dibangun berdasarkan arsitektur yang telah didesign oleh tim pengembang di DPTSI.
b. Aplikasi myITS Academics Modul MKU dikembangkan untuk membantu Modul FRS yang merupakan mata kuliah umum memploting mahasiswa dari berbagai bidang studi secara otomatis.
c. Aplikasi myITS Ormawa dikembangkan untuk menjadi platform bagi pengurus organisasi dan Staff direktorat kemahasiswaan ITS dalam mengorganisir organisasi-organisasi dalam maupun luar ITS.

7.2 Saran

Saran untuk perancangan arsitektur sistem Backend API modul MKU pada myITS Academics dan myITS Ormawa adalah sebagai berikut :

a. Untuk meningkatkan kualitas aplikasi, disarankan mengoptimalkan pengembangan backend dengan menerapkan unit testing pada setiap lapisan kode (handler, service, repository) menggunakan framework seperti testify di Golang. Hal ini memastikan validasi logika bisnis, integrasi data, dan respons API berjalan sesuai ekspektasi, termasuk penanganan skenario batas (edge cases) atau kegagalan sistem. Integrasi testing ke pipeline

CI/CD (contoh: GitHub Actions) juga diperlukan untuk automasi proses validasi tiap perubahan kode.

b. Lakukan indexing pada kolom-kolom yang sering digunakan sebagai parameter where untuk menaikkan performa query.

DAFTAR PUSTAKA

Microsoft. (2023). SQL Server Documentation. Diakses dari: https://docs.microsoft.com/en-us/sql/sql-server/
Sumber resmi untuk fitur, arsitektur, dan implementasi Microsoft SQL Server.

Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.
Referensi utama prinsip Clean Architecture dan pemisahan tanggung jawab lapisan.

Young, G. (2010). CQRS Documents. Diakses dari: https://cqrs.files.wordpress.com/2010/11/cqrs_documents.pdf
Dokumen konseptual Command Query Responsibility Segregation (CQRS) oleh pencipta pola.

The Go Team. (2023). The Go Programming Language Documentation. Diakses dari: https://golang.org/doc/
Panduan resmi sintaksis, fitur, dan best practices Golang.

Gin Web Framework. (2023). Gin Documentation. Diakses dari: https://gin-gonic.com/docs/
Dokumentasi penggunaan framework Gin untuk routing dan HTTP handling di Golang.

GORM. (2023). GORM Guide. Diakses dari: https://gorm.io/docs/
Panduan implementasi ORM GORM untuk interaksi database di Golang.

Cockroach Labs. (2023). CockroachDB Documentation. Diakses dari: https://www.cockroachlabs.com/docs/

Penjelasan fitur clustering, konsistensi data, dan arsitektur terdistribusi CockroachDB.

k6.io. (2023). k6 Load Testing Documentation. Diakses dari: https://k6.io/docs/
Panduan simulasi beban pengguna dan analisis performa aplikasi.

Apache JMeter. (2023). JMeter User Manual. Diakses dari: https://jmeter.apache.org/usermanual/
Referensi pengujian skalabilitas dan tekanan sistem.

GitHub. (2023). GitHub Actions Documentation. Diakses dari: https://docs.github.com/en/actions
Integrasi CI/CD untuk automasi testing dan deployment.

Fowler, M. (2014). Continuous Integration. Diakses dari: https://martinfowler.com/articles/continuousIntegration.html
Konsep pengembangan berkelanjutan dan integrasi otomatis.

Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.
Perbandingan arsitektur monolitik modular vs. microservices.

Khorikov, V. (2020). Unit Testing Principles, Practices, and Patterns. Manning Publications.
Prinsip unit testing dan implementasi di berbagai lapisan aplikasi.

Kleppmann, M. (2017). Designing Data-Intensive Applications. O'Reilly Media.
Best practices manajemen data terdistribusi dan optimasi query.

BIODATA PENULIS I

Nama	: Yoel Mountanus Sitorus
Tempat, Tanggal Lahir	: Pematangsiantar, 15 November 2003
Jenis Kelamin	: Laki-laki
Telepon	: +6282213989336
Email	: yoelsit@gmail.com

AKADEMIS

Kuliah	: Departemen Teknik Informatika – FTEIC , ITS
Angkatan	: 2021
Semester	: 7 (Tujuh)