

Pengembangan *Front-End* Sistem Manajemen Kuis dengan *Service Oriented Architecture*

Bayu Dwiyanto
Jurusan Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
18523162@students.uii.ac.id

Hanson Prihantoro Putro
Jurusan Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
hanson@uui.ac.id

Abstract—Sebuah kuis dibuat untuk dapat menjadi penentu apakah seseorang bisa memenuhi suatu standar. Untuk tujuan akademik kuis digunakan sebagai bahan penilaian hasil belajar pada periode tertentu atau pada akhir suatu materi. Pelaksanaan kuis yang umum dilakukan di kebanyakan universitas masih dilaksanakan secara konvensional dengan menggunakan media kertas. Pelaksanaan kuis tersebut memiliki beberapa kekurangan yaitu membutuhkan media kertas yang banyak, biaya yang tidak sedikit, dan proses pemeriksaan jawaban yang memakan waktu yang lama. Tidak jarang ketika kuis selesai dilaksanakan seringkali para mahasiswa tidak mengetahui nilai yang didapatnya dan tidak bisa menjadi bahan evaluasi bagi masing-masing mahasiswa. Berdasarkan permasalahan di atas akan dilakukan pengembangan sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*. *Service Oriented Architecture* atau SOA adalah arsitektur perangkat lunak yang dibuat berorientasi layanan. SOA memiliki kelebihan yaitu memiliki sifat *loosely coupled*, *reuse* dan *interoperability*. Dengan pengembangan sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*, dapat membantu para mahasiswa maupun dosen dalam melakukan proses pelaksanaan kuis, manajemen nilai dan menyampaikan konten pembelajaran. Adanya sistem ini proses kuis dapat berjalan lebih efisien dan mengurangi kesalahan dalam perhitungan nilai

Keywords—Kuis, *Service Oriented Architecture*, *Front-End*.

I. PENDAHULUAN

Perkembangan penggunaan kuis pada saat ini masih umum ditemukan menggunakan metode konvensional dengan menggunakan media kertas atau lembar jawaban. Seiring berkembangnya teknologi informasi dan komunikasi, media pelaksanaan kuis kini sudah mulai berubah. Banyak universitas yang telah melaksanakan kuis dengan memanfaatkan banyak aplikasi manajemen kuis sebagai media pelaksanaan kuis secara online. Beberapa contoh media pelaksanaan kuis secara online seperti Google Classroom, Kahoot dan Quizizz.

Kuis biasanya digunakan sebagai bahan penilaian hasil belajar pada periode tertentu atau pada akhir suatu materi. Dalam hal ini, kuis berfungsi sebagai bahan evaluasi pembelajaran bagi pengajar atas keefektifan dalam proses pembelajaran. Untuk tujuan yang berbeda kuis juga bisa menjadi penentuan apakah seseorang memenuhi standar yang telah ditetapkan atau tidak memenuhi standar yang telah ditetapkan. Kuis bisa berarti pula sebagai sarana penentu kemenangan dalam suatu ajang kompetensi.

Saat ini pelaksanaan kuis yang umum diterapkan di universitas masih dilaksanakan secara konvensional dengan menggunakan media kertas. Pelaksanaan kuis tersebut memiliki beberapa kekurangan yaitu membutuhkan media kertas yang banyak, biaya yang tidak sedikit, dan proses pemeriksaan jawaban yang memakan waktu yang lama. Tidak jarang lembar jawaban juga tidak terbaca, sobek maupun hilang. Ketika kuis selesai dilaksanakan seringkali para mahasiswa tidak mengetahui nilai yang didapatnya dan tidak bisa menjadi bahan koreksi bagi masing-masing mahasiswa.

Berdasarkan permasalahan di atas akan dilakukan pengembangan sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*. *Service Oriented Architecture* atau SOA adalah arsitektur perangkat lunak yang dibuat berorientasi layanan [1]. SOA memiliki komponen yang memiliki sifat *loosely coupled* yang mempertemukan kebutuhan proses bisnis dan pengguna. Kelebihan dari arsitektur SOA dalam penerapan pada perangkat lunak adalah *loosely coupled*, *reuse* dan *interoperability*. *loosely coupled* adalah memungkinkan layanan bisa berjalan dengan baik walaupun layanan lainnya sedang tidak berjalan. *Reuse* memungkinkan terlaksananya pembagian kerja atau desentralisasi dari layanan yang ada. *Interoperability* adalah kelebihan dari SOA yang bisa digunakan di segala jenis *device*.

II. KAJIAN PUSTAKA

A. Sistem Manajemen Kuis

Sistem manajemen berbasis web dapat diartikan sebagai sebuah sistem informasi yang menggunakan teknologi web atau internet untuk memberikan informasi dan layanan kepada pengguna [2]. Oleh karena itu, sistem manajemen kuis adalah sebuah sistem yang berfokus pada pengembangan media kuis yang memanfaatkan teknologi web dan internet sebagai media pelaksanaan dan pengelolaan kuis. dengan memanfaatkan sistem ini, pengguna dapat membuat kuis, mengerjakan kuis, melihat nilai, mengelola nilai dan membuat kelas.

Google Classroom sebagai contoh penerapan sistem manajemen kuis telah menyediakan berbagai kebutuhan dalam manajemen kuis seperti membuat kelas online, membuat tugas, dan melakukan penilaian secara online [3].Selanjutnya juga ada *Quizizz* yang meruplan platform kuis bagis siswa bersifat *gamified* yang menawarkan banyak fitur untuk membuat kelas menjadi lebih menyenangkan, interaktif, dan menarik dengan menambahkan gambar, video, dan audio ke dalam setiap kuis [4].

Kemudian dalam sebuah penelitian yang berjudul Pengembangan Sistem Informasi Pemantauan Kuis dan Tugas pada Learning Management Systems Moodle yang mengangkat masalah tentang manajemen kuis dan tugas yang susah untuk dikelola secara teratur. Penelitian tersebut bertujuan untuk memudahkan pengelolaan tugas dan kuis yang diupload pada learning management systems Moodle. Metode yang digunakan dalam penelitian tersebut menggunakan cara pengembangan sistem informasi secara langsung. Penelitian ini menghasilkan bahwa dengan mengembangkan learning management systems Moodle dengan menambahkan fungsi filter validasi kuis dan tugas dapat memudahkan pengguna untuk mengelola kuis dan tugas yang diupload pada learning management systems [5].

B. Front-End

Front-end adalah sebuah istilah yang biasa digunakan untuk menggambarkan suatu tampilan dari komputer untuk berinteraksi dengan pengguna [6]. *Front-end* adalah bagian dari antarmuka suatu sistem yang berfokus pada keindahan sebuah tampilan, layout, pemilihan warna dan typography. Sebuah *front-end* biasanya dikembangkan untuk melengkapi bagian dari suatu sistem. Umumnya dalam proses pengembangan sebuah sistem terbagi menjadi dua yaitu pengembangan *back-end* dan *front-end*.

Front-end juga memiliki pengertian sebagai segala sesuatu yang dapat menghubungkan antara pengguna dengan *back-end*. Seorang *front-end developer* hanya berfokus pada pengembangan tampilan suatu aplikasi dengan memperhatikan keindahan tampilan (*user interface*) dan kemudahan dalam penggunaan sistem (*user experience*). Pengembangan suatu sistem dengan membedakan pengembangan *back-end* dan *front-end* akan dapat memaksimalkan proses pengembangan suatu sistem [7].

C. Service Oriented Architecture

Service Oriented Architecture atau bisa disebut dengan SOA adalah gaya desain yang membangun model arsitektur IT yang berfokus pada peningkatan efisiensi, kelincihan dan produktivitas dengan menempatkan layanan sebagai sarana utama. Pada dasarnya SOA adalah sebuah arsitektur aplikasi yang mengatur setiap layanan dan fungsi yang dideskripsikan dalam suatu Bahasa agar bisa mengakses antarmuka yang dipanggil yang digunakan untuk melakukan proses bisnis. Antarmuka service SOA yang independen menjadikan setiap pengguna dari berbagai *device* dapat menggunakan layanan yang disediakan [1].

Terdapat sebuah penelitian yang mengimplementasikan *Service Oriented Architecture* dengan web service untuk aplikasi informasi akademik [8]. Penelitian ini bertujuan untuk mengetahui implementasi SOA dalam menghadapi permasalahan informasi akademik menggunakan teknologi web service. Penelitian ini menerangkan bahwa penerapan SOA dalam aplikasi informasi akademik memudahkan penggunaan data mahasiswa/dosen/matakuliah dan data lainnya yang digunakan kembali oleh aplikasi lain atau sistem informasi lain dalam organisasi.

Kemudian dalam penelitian selanjutnya yang bertujuan untuk mengetahui hasil implementasi SOA pada sistem pengelolaan data asuransi BPJS [9]. Penelitian ini menghasilkan bahwa metode SOA berhasil diterapkan dalam sistem asuransi BPJS karena telah berhasil dalam

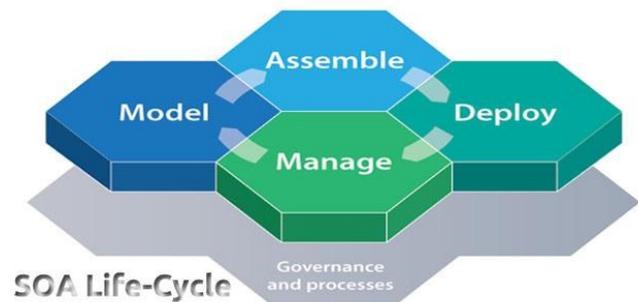
memudahkan komunikasi antar data perusahaan-perusahaan dan mempercepat proses rujukan asuransi karyawan.

Selanjutnya dalam penelitian yang bertujuan untuk merancang sistem informasi berbasis SOA dengan menerapkan kriteria *interoperability* dan reusable [10]. Penelitian ini menghasilkan bahwa sifat reusable SOA sangat berguna untuk service karena bisa digunakan Kembali pada sistem yang lain. Kemudian sifat *interoperability* yang dimiliki SOA terbukti bisa digunakan dalam berbagai *device pc* maupun *mobile*.

Penelitian berikutnya menerangkan bahwa penggunaan SOA terbukti mampu mengintegrasikan layanan e-learning dengan sistem informasi sekolah berbasis ERP [11]. Dalam hasil uji sistem menunjukkan service yang dimiliki setiap sistem dapat saling berkomunikasi secara lancar. Hal tersebut disebabkan oleh penggunaan infrastruktur ESB (*enterprise service bus*) pada sistem.

III. METODOLOGI

Penelitian ini dimulai dari proses pengembangan perangkat lunak yaitu sistem manajemen kuis. Seperti disampaikan sebelumnya bahwa penelitian ini melakukan pengembangan perangkat lunak berbasis layanan, maka pekerjaan yang dilakukan berdasarkan *Service Oriented Architecture Lifecycle*. SOA mempunyai tahapan-tahapan atau siklus hidup yang menjadi petunjuk dalam merancang sebuah arsitektur aplikasi. Siklus hidup SOA bisa dilihat pada gambar 1.



Gambar 1. SOA Life-Cycle [12]

Tahap pertama dalam siklus tersebut yaitu tahap Model di mana dilakukan pengumpulan kebutuhan bisnis, merancang, mensimulasikan dan mengoptimalkan proses bisnis yang diinginkan pengguna. Yang dilakukan di laporan. Pada tahap ini juga telah dilakukan identifikasi proses bisnis yang akan diterapkan pada sistem manajemen kuis. Kemudian juga telah didapatkan identifikasi pengguna yang akan menggunakan sistem manajemen kuis.

Tahap berikutnya yaitu Assemble yaitu menerapkan proses bisnis yang dioptimalkan dengan menggabungkan layanan yang baru dibuat dan yang sudah ada untuk membentuk aplikasi. Tahap assemble merupakan tahap untuk menggabungkan layanan yang telah dibuat. Pada tahap ini telah dilakukan perancangan yang berisi activity diagram, data base diagram, Mock up design antar muka, dan use case diagram.

Tahap ketiga yaitu Deploy. Aset kemudian disebarkan ke lingkungan yang aman dan terintegrasi, dengan memanfaatkan layanan dengan dukungan untuk

mengintegrasikan orang, proses, dan informasi. Pada tahap ini juga memastikan bahwa semua elemen penting perusahaan dapat terhubung dan saling bekerja sama. Kemudian juga deploy memastikan sistem bisa diakses oleh setiap pengguna. Pada tahap ini telah dilakukan perancangan desain antar muka yang berdasarkan pada perancangan mock up design sistem. Perancangan desain antar muka telah menghasilkan sepuluh halaman *front-end* sistem manajemen kuis.

Setelah aplikasi diterapkan, pengguna dapat melakukan Manage atau mengelola dan memantau aplikasi dan sumber daya dari perspektif teknologi informasi dan bisnis. Informasi yang dikumpulkan dalam waktu yang sebenarnya akan mendapatkan wawasan tentang proses bisnis, memungkinkan keputusan bisnis yang lebih baik dan memberikan informasi kembali ke siklus hidup untuk penyempurnaan dan perbaikan berkelanjutan. Pada tahap ini telah dilakukan pengujian dengan melibatkan dua *front-end developer*. Proses pengujian yang dilakukan adalah dengan metode wawancara langsung dan memperlihatkan langsung aplikasi kepada penguji.

IV. HASIL DAN PEMBAHASAN

A. Model

Pada tahap ini proses pengembangan aplikasi akan dimulai dengan mengumpulkan kebutuhan bisnis, merancang, mensimulasikan dan mengoptimalkan proses bisnis yang diinginkan oleh pengguna.

a) Identifikasi Proses Bisnis

Proses pertama yaitu dosen bisa membuat kelas sesuai dengan kelas yang akan diampunya, selanjutnya setelah kelas berhasil dibuat, mahasiswa bisa bergabung dengan kelas tersebut, di dalam kelas tersebut mahasiswa bisa melihat kuis yang tersedia, selanjutnya dosen bisa membuat kuis yang selanjutnya akan dikerjakan oleh mahasiswa sesuai dengan hari dan jam yang telah ditentukan, setelah waktu kuis selesai dan semua mahasiswa telah menyelesaikan kuis dosen bisa melakukan penilaian terhadap setiap kuis yang telah selesai dikerjakan dan yang terakhir adalah para mahasiswa bisa melihat hasil kuis yang telah dinilai oleh dosen.

b) Identifikasi Pengguna

Pengguna pertama adalah mahasiswa, mahasiswa berperan sebagai pengguna sistem yang bisa melakukan login, mengerjakan kuis dan melihat hasil dari setiap kuis yang telah dikerjakan. Pengguna kedua adalah dosen, dosen merupakan aktor yang berperan penting dalam jalannya sistem manajemen kuis. Peran dosen disini nantinya akan membuat kelas, membuat kuis, mengelola nilai kuis dan menyerahkan nilai kepada setiap mahasiswa.

B. Assamble

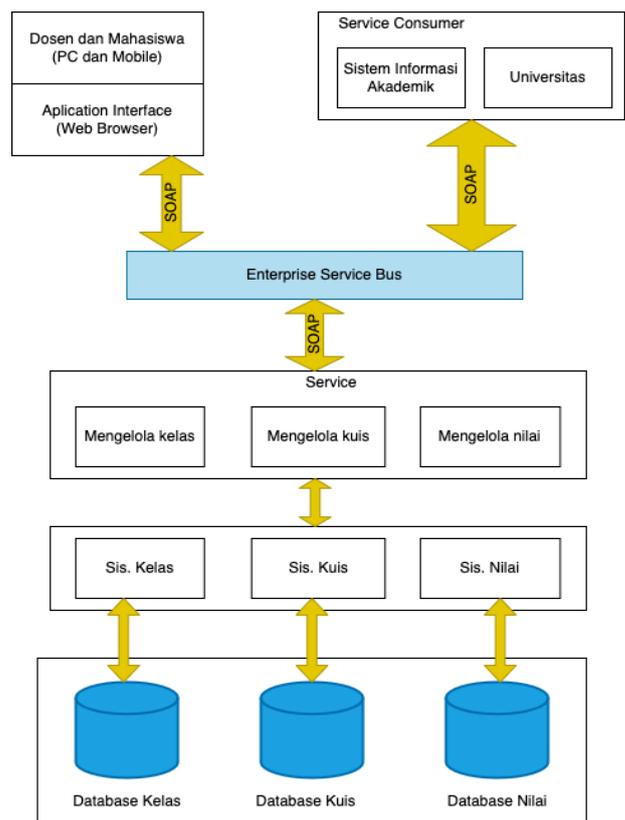
Assamble adalah tahap menggabungkan layanan yang baru dibuat dan yang sudah ada untuk membentuk aplikasi. Pada tahap ini berisi Activity Diagram, DB Diagram, Desain Antarmuka, dan Use Case Diagram.

1) Identifikasi Sistem

Sistem manajemen kuis membutuhkan sistem pengelolaan kelas, sistem pengelolaan kuis dan sistem pengelolaan nilai. Sistem pengelolaan kelas terdiri dari fungsionalitas membuat kelas dan melihat kelas. Sistem pengelolaan kuis terdiri dari membuat kuis, mengerjakan kuis, dan yang terakhir sistem pengelolaan nilai yang memiliki fungsionalitas menilai kuis dan melihat hasil kuis. Masing-masing sistem akan mempunyai basis data yang diperlukan untuk menjalankan sistem manajemen kuis. Pembagian sistem menjadi beberapa sistem kecil berfungsi untuk memenuhi prinsip *loose coupled* yang artinya sebuah *service* dapat dipanggil oleh *service* lainnya tanpa perlu pemanggil tersebut memperhatikan di mana lokasi *service* yang dipanggil. Misalnya sistem manajemen kuis memanggil *service* kuis, maka *service* kelas dan *service* nilai tidak perlu ikut terpanggil. Hal ini memiliki tujuan untuk meminimalisir *service* memiliki ketergantungan dengan *service* yang lainnya.

2) Rancangan Arsitektur Sistem

Rancangan arsitektur sistem manajemen kuis terdiri dari empat lapisan yaitu *service consumer*, *message broker*, *Service registry* dan *service provider*. Untuk lebih jelasnya bisa dilihat pada Gambar 2.



Gambar 2. Rancangan arsitektur Sistem Manajemen Kuis

a) Service Consumer

Sistem *Service Consumer* merupakan user yang mempunyai akses untuk menggunakan *service* yang tersedia untuk mengakses segala informasi yang dibutuhkan yang telah disediakan oleh *service provider*. Selain user, *service consumer* juga terdapat pada platform yang akan digunakan

oleh user untuk mengakses *service* seperti pada platform PC dan Mobile.

b) *Service provider*

Service provider atau penyedia *service* adalah sebagai penyedia *service* yang akan dibutuhkan pada tingkat produksi. Pada Gambar 2 terdapat tiga *service* yang tersedia yaitu mengelola kelas, mengelola kuis dan mengelola nilai. Tiga *service* yang telah tersedia nantinya akan digunakan oleh *service consumer*. ketiga *service* ini dibuat sebagai penerapan dari sifat *loosely couple*, yaitu adalah agar setiap *service* tidak memiliki ketergantungan dengan *service* yang lainnya ketika berjalan.

c) *Service Registry*

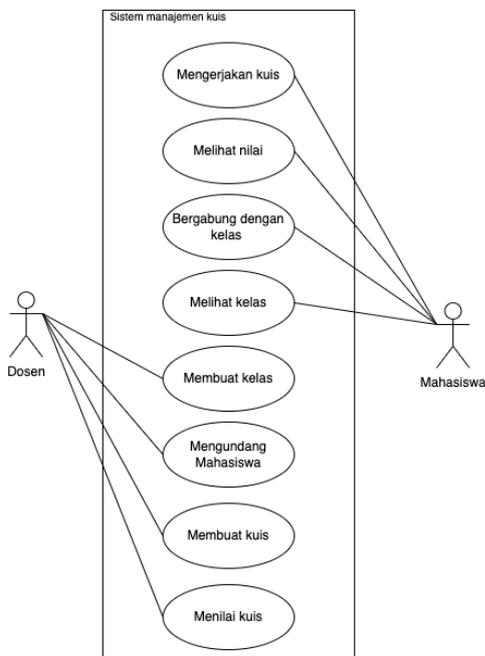
Service Registry merupakan sebuah tempat untuk menyimpan data *service-service* yang diperlukan. *Service Registry* biasanya terdapat pada layer *service orchestration*.

d) *Message Broker (ESB)*

Message Broker (ESB) merupakan teknologi yang memiliki peran untuk menyampaikan pesan atau *service*. *Message broker* memiliki bermacam teknologi, dalam perancangan arsitektur ini menggunakan *ESB (enterprise service bus)*. Selain itu untuk menyampaikan komunikasi *service* kepada user menggunakan *simple object acces protocol (SOAP)* antara *ESB* dan user.

3) *Use Case*

Use case untuk sistem utama Manajemen Kuis ini memiliki 3 diagram sesuai rancangan identifikasi sistem sebelumnya yang juga menghasilkan 3 sistem. Berikut adalah *use case* dari sistem manajemen kuis bisa dilihat pada Gambar 3.



Gambar 3. Use Case Sistem Manajem Kuis

4) *Acitivity diagram*

Activity diagram merupakan diagram yang menjelaskan aliran kerja sistem secara umum. *Activity diagram* dapat menggambarkan aliran kerja secara umum dari sistem yang dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi dan bagaimana proses berakhir.

a) *Activity diagram mahasiswa mengerjakan kuis*

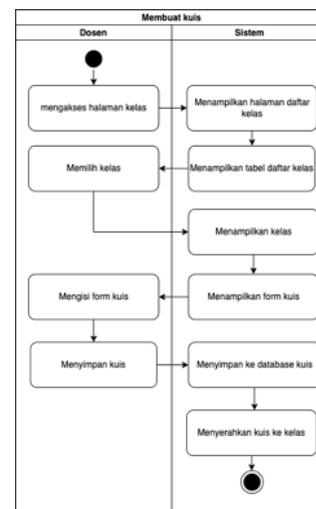
Activity diagram dibawah ini menggambarkan alur mengerjakan kuis untuk mahasiswa. Mahasiswa mengakses halaman kelas yang selanjutnya mahasiswa bisa melihat jadwal kuis dari setiap kelas yang diikutinya. Setelah mahasiswa melihat jadwal, mahasiswa bisa langsung mengakses kuis yang sudah tersedia pada jadwal kuis, selanjutnya mahasiswa bisa mengerjakan kuis. Struktur lengkap diagram dapat dilihat pada Gambar 4.



Gambar 4. Activity diagram mahasiswa mengerjakan kuis

b) *Activity diagram dosen membuat kuis*

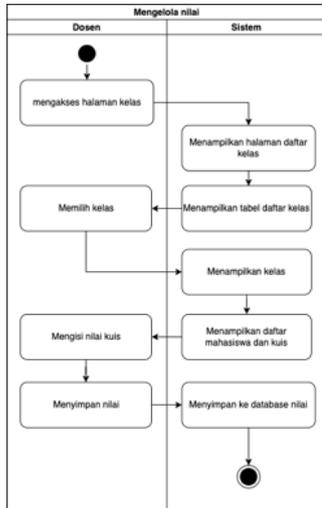
Activity diagram dibawah ini menggambarkan alur membuat kuis untuk dosen. Dosen melakukan akses ke halaman kelas dan memilih kelas mana yang akan melakukan kuis, selanjutnya dosen mengakses halaman buat kuis, setelah selesai dosen dapat menyimpan kuis tersebut untuk ditampilkan di kelas. Struktur lengkap diagram dapat dilihat pada Gambar 5.



Gambar 5. Activity diagram dosen membuat kuis

c) *Activity diagram dosen mengelola nilai*

Activity diagram dibawah ini menggambarkan alur mengelola nilai kuis untuk dosen. Dosen melakukan akses ke halaman kelas dan memilih kelas mana yang akan dikelola nilainya, selanjutnya dosen bisa mengakses data setiap mahasiswa selanjutnya dosen bisa memasukan nilai kuis ke masing-masing mahasiswa dan tahap terakhir adalah menyerahkannya nilai kuis kepada masing-masing mahasiswa. Struktur lengkap diagram dapat dilihat pada Gambar 6.



Gambar 6. Diagram relasi basis data sistem manajemen kuis

a) *Detail layanan sistem pengelolaan kelas*

Detail layanan sistem pengelolaan kelas menghasilkan empat usecase, satu service, empat belas halaman front-end, satu controller back-end dan empat method. Fungsi pembuatan detail layanan sistem adalah sebagai rincian dari sistem pengelolaan kelas seperti membuat kelas, mengundang mahasiswa, bergabung dengan kelas dan melihat kelas. Pembuatan detail layanan dari sistem pengelolaan kelas akan memudahkan dalam proses pengembangan sistem. Untuk lebih jelas bisa dilihat pada Tabel I.

TABEL I. TABEL DETAIL LAYANAN SISTEM PENGELOLAAN KELAS

Use Case	Service	Front-End	Back-End	Method
UC1 Membuat kelas	Kelas	[1] Halaman Daftar Kelas [2] Halaman form buat kelas [3] Halaman Menghasilkan link undangan kelas	Controller Kelas	Method POST Add
UC2 Mengundang mahasiswa		[4] Menampilkan daftar kelas [5] Halaman tabel kelas [6] Halaman kode kelas dan link undangan		Method POST undang mahasiswa
UC3 Bergabung dengan kelas		[7] Menampilkan Halaman awal sistem [8] Menampilkan form input kode kelas [9] menghasilkan alert sukses [10] berhasil bergabung dengan kelas		Method GET
UC4 Melihat kelas		[11] Menampilkan halaman awal sistem [12] Menampilkan halaman login [13] Verifikasi identitas [14] Menampilkan daftar kelas		Method GET

5) *Perancangan detail layanan sistem manajemen kuis*

Perancangan detail layanan sistem menghasilkan delapan use case, tiga service, 23 halaman front-end, tiga controller back-end dan delapan method. Untuk lebih jelas terkait perancangan detail layanan bisa dilihat pada Tabel I, Tabel II dan Tabel III.

b) *Detail layanan sistem pengelolaan kuis*

Detail layanan sistem pengelolaan kuis menghasilkan dua usecase, satu service, sembilan halaman front-end, satu controller back-end dan dua method. Fungsi pembuatan detail layanan sistem adalah sebagai dokumen rincian dari layanan sistem pengelolaan kuis seperti membuat kuis dan mengerjakan kuis. Pembuatan detail layanan dari sistem pengelolaan kuis akan memudahkan dalam proses pengembangan sistem. Untuk lebih jelas bisa dilihat pada Tabel II.

TABEL II. TABEL DETAIL LAYANAN SISTEM PENGELOLAAN KUIS

Use Case	Service	Front-End	Back-End	Method
UC5 Membuat kuis	Kuis	[15]Menampilkan halaman daftar kelas [16]Menampilkan tabel daftar kelas [17]Menampilkan kelas [18]Menampilkan form kuis [19]menyimpan ke database kuis [20]menyerahkan kuis ke kelas	Controller kuis	Method GET kuis
UC6 Mengerjakan kuis		[21]Menampilkan tabel kelas dan kuis [22]Menampilkan kuis yang dipilih [23]Menyimpan kuis		Method GET kuis

c) *Detail layanan sistem pengelolaan nilai*

Detail layanan sistem pengelolaan kuis menghasilkan dua usecase, satu service, delapan halaman front-end, satu controller back-end dan dua method. Fungsi pembuatan detail layanan sistem adalah sebagai dokumen rincian dari layanan sistem pengelolaan nilai seperti menilai kuis dan melihat nilai. Pembuatan detail layanan dari sistem pengelolaan kuis akan memudahkan dalam proses pengembangan sistem. Untuk lebih jelas bisa dilihat pada Tabel III.

TABEL III. TABEL DETAIL LAYANAN SISTEM PENGELOLAAN NILAI

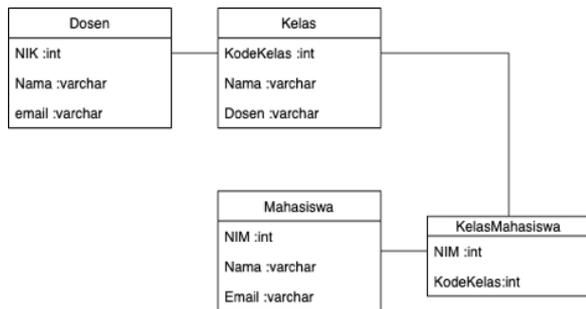
Use Case	Service	Front-End	Back-End	Method
UC7 Menilai kuis	Nilai	[24]Menampilkan halaman daftar kelas [25]Menampilkan daftar kelas [26]Menampilkan kelas [27]menampilkan daftar mahasiswa dan kuis [28]Menyimpan ke database kuis	Controller nilai	Method GET nilai
UC8 Melihat nilai		[29]Menampilkan halaman kelas dan kuis [30]Menampilkan menu <i>sidebar</i> kelas dan hasil kuis [31]Menampilkan halaman hasil kuis		Method POST nilai

6) *Diagram Relasi Basis Data Sistem Manajemen Kuis*

Diagram relasi basis data menunjukkan komponen-komponen yang terlibat dalam sistem manajemen kuis. Perancangan diagram relasi basis data sistem manajemen kuis akan menghasilkan tiga diagram untuk masing-masing *service*.

a) *Diagram relasi basis data sistem manajemen kuis*

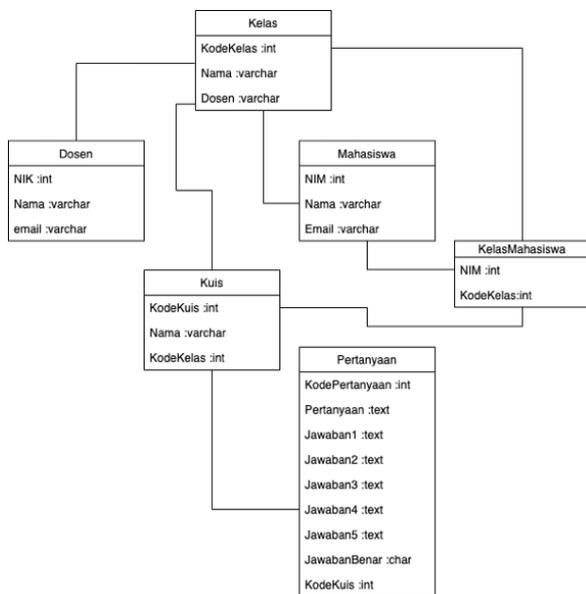
Diagram pada Gambar 7 Menggambarkan relasi-relasi basis data dari sistem pengelolaan kuis. Diagram relasi basis data pengelolaan kuis terdiri dari tabel dosen, tabel kelas, tabel mahasiswa dan tabel kelasMahasiswa.



Gambar 7. Diagram relasi basis data sistem manajemen kuis

b) *Diagram relasi basis data pengelolaan kuis*

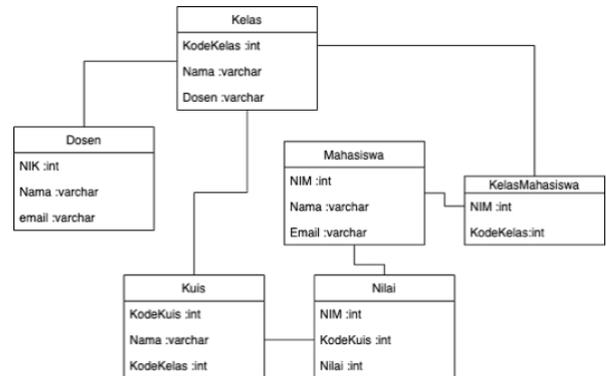
Diagram pada Gambar 8 Menggambarkan relasi basis data dari sistem pengelolaan kuis. Diagram relasi basis data pengelolaan kuis terdiri dari tabel dosen, tabel kelas, tabel mahasiswa, tabel kelasMahasiswa, tabel kuis dan tabel pertanyaan.



Gambar 8. Diagram relasi basis data pengelolaan kuis

c) *Diagram relasi basis data pengelolaan nilai*

Gambar 9 Menggambarkan relasi relasi basis data dari sistem pengelolaan nilai. Diagram relasi basis data pengolahan nilai terdiri dari tabel dosen, tabel kelas, tabel mahasiswa, tabel kelasMahasiswa, tabel kuis dan tabel nilai.



Gambar 9. Diagram relasi basis data pengelolaan nilai

Proses desain antar muka dimulai dengan mencari inspirasi desain dari sistem manajemen kuis yang telah ada. Kemudian dimulai dengan membuat mock up design sistem menggunakan aplikasi *miro*. Pembuatan mock up design sistem menghasilkan sepuluh rancangan desain yang selanjutnya menjadi acuan dalam proses pembuatan desain antar muka.

Kemudian setelah didapatkan sepuluh mock up design dilakukan proses pembuatan *front-end* yang menghasilkan sepuluh halaman *front-end*. Proses pembuatan *front-end* sistem manajemen kuis menggunakan *css framework* dari *bootstrap*. *Framework bootstrap* digunakan karena akan mempermudah dan mempercepat programmer dalam melakukan pengembangan suatu web. Selanjutnya untuk membuat website lebih responsive digunakan *sweet alert* sebagai pengganti alert bawaan dari html.

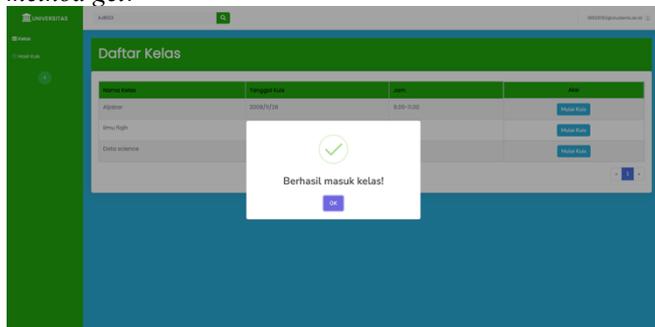
C. *Deploy*

Deploy adalah tahap dari sebuah sistem disebarkan ke lingkungan yang aman dan terintegrasi, dengan memanfaatkan layanan beserta dukungan untuk mengintegrasikan orang, proses, dan informasi. Dalam tahap deployment ini, dilakukan implementasi sistem.

Implementasi sistem merupakan tahapan dari pembuatan sistem informasi yang dibangun berdasarkan tahapan perancangan sistem yang telah dibuat sebelumnya. Pada tahap implementasi digunakan untuk menguji sistem informasi yang telah dibangun supaya dapat berfungsi sesuai dengan kebutuhan. Implementasi yang akan dilakukan mengenai antarmuka dan proses untuk mendapatkan hasil yang diharapkan. Implementasi sistem manajemen kuis hanya berfokus pada pengembangan pada sisi *front-end* dengan lebih mengembangkan tampilan visual dari sistem manajemen kuis.

1) Halaman mahasiswa masuk kelas

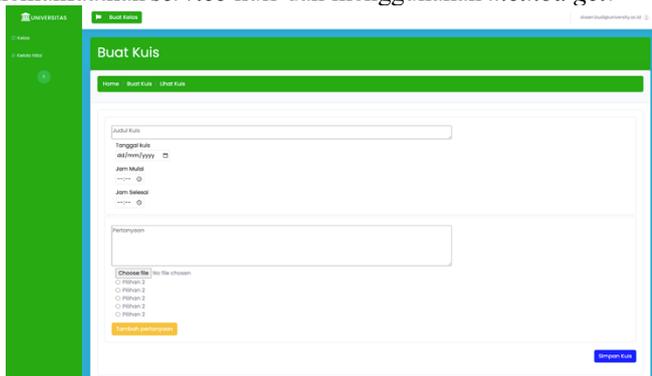
Gambar 10 Merupakan halaman ketika mahasiswa berhasil masuk ke kelas dengan menggunakan kode kelas. manajemen kuis. Sistem akan menampilkan notifikasi atau *alert* ketika mahasiswa berhasil masuk ke kelas. Kemudian juga halaman ini merupakan bagian dari *use case* melihat kelas yang memanfaatkan *service* kelas dan menggunakan *method get*.



Gambar 10. Halaman mahasiswa masuk kelas

2) Halaman dosen membuat kuis

Gambar 11 menunjukan halaman ketika dosen ingin membuat kuis. Halaman dosen membuat kuis mempunyai text input judul kuis dan pertanyaan sedangkan jawaban kuis menggunakan *radiobutton*. Kemudian halaman ini merupakan bagian dari *use case* membuat kuis yang memanfaatkan *service* kuis dan menggunakan *method get*.



Gambar 11. Halaman dosen membuat kuis

3) Halaman dosen mengelola nilai

Gambar 12 adalah halaman ketika dosen mengakses halaman kelola nilai. Halaman kelola nilai merupakan halaman ketika dosen ingin mengelola nilai kelas yang didapatkan dari kuis, uts dan uas. Kemudian halaman ini merupakan bagian dari *use case* nilai yang memanfaatkan *service* nilai dan menggunakan *method get*.



D. Manage

Proses pengujian akan dilakukan dengan metode *usability testing*. *Usability testing* adalah suatu kegiatan yang dilakukan secara iterative untuk mendapatkan respon yang komprehensif dari pengguna. Teknik ini digunakan untuk menentukan fungsi, melakukan analisa data dengan merekam atau mencatat hasil dari *usability testing*, melakukan pengukuran *usability* dengan kuesioner dan melakukan analisis untuk menghasilkan rekomendasi perbaikan sistem ke depannya [13]. Pengujian telah dilakukan dengan melakukan wawancara dengan dua orang *front-end developer*. Pengujian dilakukan melalui media zoom meeting dengan menunjukkan langsung sistem manajemen kuis kepada para penguji. Setelah para penguji melakukan testing, dilakukan wawancara terkait hasil pengujian. Detail hasil wawancara dapat dilihat pada Tabel IV [14].

TABEL IV. TABEL DETAIL HASIL WAWANCARA

No	Pertanyaan	Kesimpulan hasil wawancara
1	Apakah anda setuju bahwa tampilan antar muka sistem manajemen kuis memiliki tampilan yang menarik?	Setuju
2	Apakah setiap komponen seperti warna button dan penggunaan alert pada sistem manajemen kuis sudah tepat?	Setuju
3	Apakah anda setuju bahwa setiap fungsionalitas sistem manajemen kuis telah sesuai dengan perancangan layanan?	Setuju
4	Apakah anda setuju sistem manajemen kuis mudah digunakan?	Setuju

Dari wawancara yang dilakukan dengan dua orang *front-end developer*, dapat disimpulkan bahwa sistem manajemen kuis dapat dengan mudah digunakan. Sistem manajemen kuis juga telah memiliki tampilan yang menarik dan telah memenuhi kaidah penggunaan *css* dan *html*. Kaidah penggunaan *css* dan *html* yang telah diterapkan pada sistem manajemen kuis seperti penggunaan warna pada setiap tombol yang memiliki fungsi yang berbeda.

Kemudian juga telah dilakukan wawancara dengan *back-end developer* untuk menguji apakah *front-end* bisa dikembangkan lebih lanjut pada sisi *back-end*. Pengujian dilakukan dengan melakukan wawancara dan menunjukan langsung sistem manajemen kuis kepada penguji. Pengujian dimulai dengan membaca dokumen penelitian yang berisi detail rancangan sistem manajemen kuis, selanjutnya dilakukan eksplorasi kode pada setiap halaman *front-end*. Berdasarkan pengujian yang telah dilakukan, halaman *front-end* sistem manajemen kuis berpotensi untuk bisa dikembangkan lebih lanjut pada sisi *back-end*.

Dalam pengujian selanjutnya dilakukan untuk mencari tahu prinsip dan sifat SOA yang telah terpenuhi oleh sistem. Sifat SOA yang telah diterapkan dalam pengembangan sistem manajemen kuis adalah *Loosely coupled*. *Loosely coupled* yaitu setiap *service* berdiri sendiri tanpa bergantung dengan *service* lainnya. Sebagai contoh yang telah diterapkan adalah sistem memiliki tiga *service* yaitu *service* kelas, *service* kuis dan *service* nilai, setiap *service* yang berada pada sistem telah berdiri sendiri.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan proses pengembangan *front-end* sistem manajemen kuis dengan *Service Oriented Architecture*, maka dapat disimpulkan bahwa telah berhasil dirancang sebuah Sistem Manajemen Kuis dengan pendekatan *Service Oriented Architecture*. Perangkat lunak ini telah menerapkan sifat SOA yaitu *loosely coupled*. Sebagai contoh yang telah diterapkan adalah sistem memiliki tiga *service* yaitu *service* kelas, *service* kuis dan *service* nilai, setiap *service* yang berada pada sistem telah berdiri sendiri tanpa bergantung dengan *service* lainnya.

Pengembangan sistem manajemen kuis dilakukan dengan empat tahap yaitu *model*, *assamble*, *deploy* dan *manage* yang menghasilkan rancangan *activity diagram*, *db diagram*, desain antarmuka, dan *use case diagram*. Hasil dari pengujian dapat disimpulkan bahwa sistem manajemen kuis mudah digunakan dan telah sesuai dengan kaidah css dan html. Kemudian pengujian dari *back-end developer* menyimpulkan bahwa *fron-end* sistem manajemen kuis bisa dikembangkan lebih lanjut ke sisi *back-end*

B. Saran

Sistem manajemen kuis hanya dikembangkan pada sisi *front-end* saja. Sistem ini belum dilakukan implementasi dari sisi *back-end* sehingga pengembangan sistem ini belum maksimal dalam pemanfaatan arsitektur SOA. Pada tahap selanjutnya sistem lebih banyak mengkaji penerapan sifat-sifat SOA dalam pengembangan sistem manajemen kuis ini. Oleh karena itu, implementasi *back-end* sangat diperlukan pada pengembangan selanjutnya.

REFERENSI

- [1] Erl, T. (2007). *SOA: Principles of Service Design*. Pearson Education. Boston.
- [2] O'brien, James A. (2006). *Introduction to Information Systems*. McGraw Hill. New York.
- [3] Edu Google. (2022). *workspace-for-education*. https://edu.google.com/intl/ALL_id/workspace-for-education/classroom/
- [4] Haripriya. (2022). *What is Quizziz?*. [https://support.quizizz.com/hc/en-us/articles/203610052-What-is-Quizizz- /](https://support.quizizz.com/hc/en-us/articles/203610052-What-is-Quizizz-/)
- [5] Firmansyah, R., Aliim, M. S., & Atmojo, E. M. (2021). Pengembangan Sistem Informasi Pemantauan Kuis dan Tugas pada Learning Management Systems Moodle. *Jurnal Pendidikan Dan Teknologi Indonesia*, 1(3), 97-102.
- [6] Morris, S. (n.d.). *Full Stack, Front End, Back End—What Does It All Mean?* <https://skillcrush.com/blog/front-end-back-end-full-stack/>
- [7] Arhandi, P. P. (2016). Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan dengan Menggunakan Metode Back End dan Front End. *Jurnal Teknologi Informasi: Teori, Konsep, Dan Implementasi*, 7(1).
- [8] Kapoadewa, F., Wowor, H. F., Rumagit, A. M., & Wowor, A. P. R. (2012). Implementasi Service-Oriented Architecture Dengan Web Service Untuk Aplikasi Informasi Akademik. *jurnal Teknik Elektro Dan Komputer*, 1(1), 1-5.
- [9] Pakusadewa, N. A., & Slamet Sudaryanto, N. (2015). Implementasi Service Oriented Architecture Pada Sistem Pengelolaan Data Asuransi Bpjs Karyawan Pt. Transportasi Jakarta. .
- [10] Gumelar, A. S., Saputra, M. C., & Wardani, N. H. (2017). Analisa Kebutuhan dan Perancangan Sistem Informasi Produksi Dinas Peternakan dan Kesehatan Hewan Kabupaten Malang Berbasis Teknologi Service Oriented Architecture. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, 2548, 964X.
- [11] Tricahyo, V. A. (2016). Pengembangan e-learning yang terintegrasi pada enterprise resource planning pondok pesantren tipe D menggunakan service oriented architecture (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).
- [12] McBride, G. (2007). *The Role of {SOA} Quality Management in {SOA} Service Lifecycle Management*. IBM.
- [13] Setiawati, A., Rahim, A., & Kisbianty, D. (2018). Pengembangan dan pengujian aspek usability pada sistem informasi perpustakaan (Studi Kasus: STIKOM Dinamika Bangsa Jambi). *Jurnal Processor*, 13(1), 1173-1188.
- [14] Fikriyya, A., & Dirgahayu, R. T. (2020). Implementasi Prototyping dalam Perancangan Sistem Informasi Sekolah Desa Pendar Foundation Yogyakarta.