

**PENGEMBANGAN *FRONT-END* SISTEM
MANAJEMEN KUIS
DENGAN *SERVICE ORIENTED ARCHITECTURE***



Disusun Oleh:

N a m a : Bayu Dwiyanto

NIM : 18523162

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

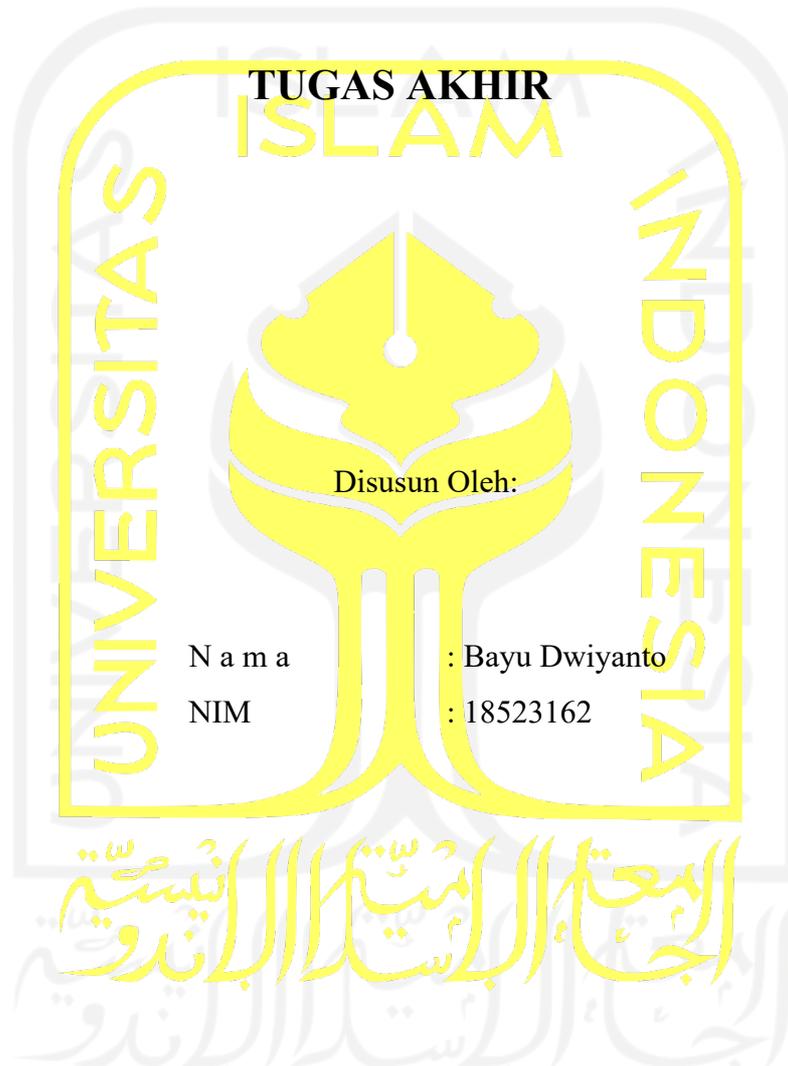
2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN *FRONT-END* SISTEM

MANAJEMEN KUIS

DENGAN *SERVICE ORIENTED ARCHITECTURE*



Yogyakarta, 6 Oktober 2022

Pembimbing,

(Hanson Prihantoro Putro, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN *FRONT-END*

SISTEM MANAJEMEN KUIS

DENGAN *SERVICE ORIENTED ARCHITECTURE*

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 22 Agustus 2022

Tim Penguji

Hanson Prihantoro Putro, S.T., M.T.

Anggota 1

Chanifah Indah Ratnasari, S.Kom.,
M.Kom.

Anggota 2

Elyza Gustri Wahyuni, S.T., M.Cs.

الجمعة الالهية الاندونيسية
Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Bayu Dwiyanto

NIM : 18523162

Tugas akhir dengan judul:

**PENGEMBANGAN *FRONT-END* SISTEM
MANAJEMEN KUIS
DENGAN *SERVICE ORIENTED ARCHITECTURE***

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

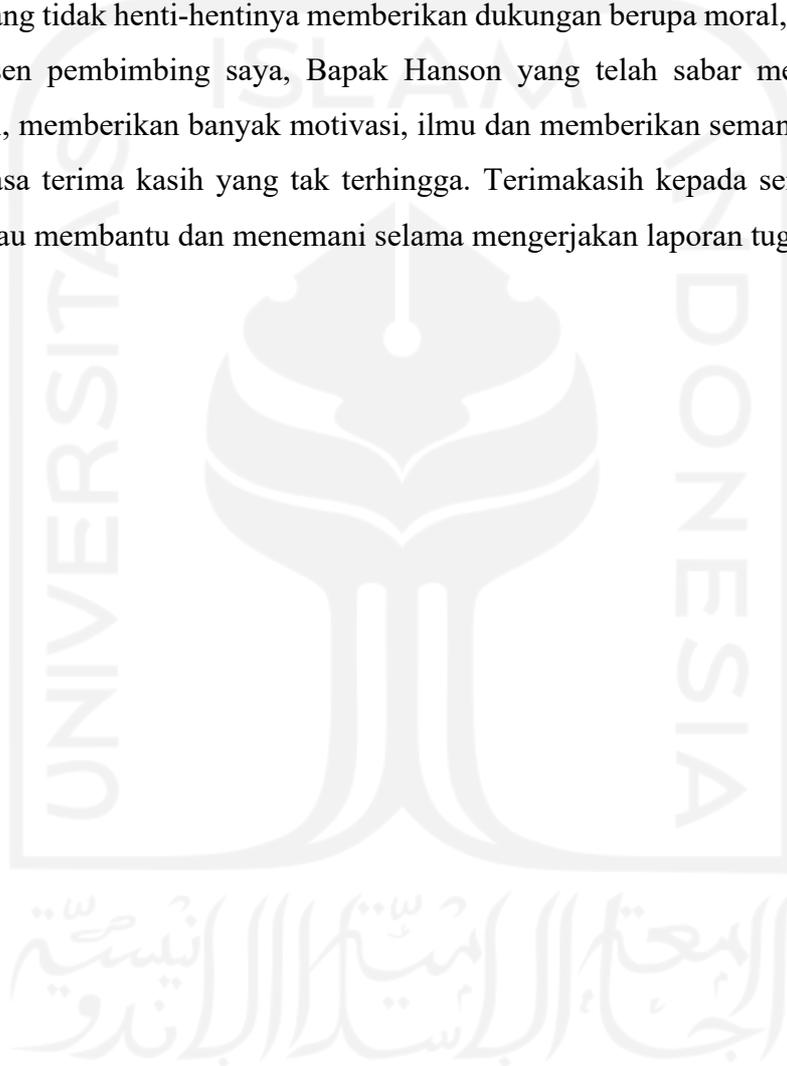
Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.



HALAMAN PERSEMBAHAN

Pertama-tama saya ucapkan Alhamdulillah wa Syukurillah atas karunia Allah SWT yang telah membimbing saya dalam menyelesaikan laporan tugas akhir. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, keluarga, sahabat serta umatnya hingga akhir zaman. Saya persembahkan tugas akhir ini kepada ayahanda tercinta Sudarsiono dan Ibunda tercinta Rubiani Sarinawati, dan juga kakak saya Bagus Rianto dan adik saya Banu Agil Triyanto yang tidak henti-hentinya memberikan dukungan berupa moral, materil dan doa.

Kepada dosen pembimbing saya, Bapak Hanson yang telah sabar membimbing saya sampai tahap ini, memberikan banyak motivasi, ilmu dan memberikan semangat yang positif, saya ucapkan rasa terima kasih yang tak terhingga. Terimakasih kepada semua orang yang terlibat untuk mau membantu dan menemani selama mengerjakan laporan tugas akhir ini.



HALAMAN MOTO

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا

fa inna ma'al-'usri yusrā

Karena sesungguhnya sesudah kesulitan itu ada kemudahan.



KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh

Alhamdulillah rabbil'alamiin, puji syukur dipanjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga laporan tugas akhir yang berjudul "Pengembangan *Front-End* Sistem Manajemen Kuis dengan *Service Oriented Architecture*" dapat diselesaikan dengan baik. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad Shalallahu 'Alayhi Wasallam, yang telah membawa umat manusia dari zaman jahiliah menuju zaman yang terang benderang seperti saat ini.

Keberhasilan laporan tugas akhir ini tidak terlepas dari bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ucapkan terimakasih dan penghargaan sebesar-besarnya kepada:

1. Allah SWT yang telah melimpahkan segala rahmat, hidayah, serta kemudahan bagi penulis untuk menyelesaikan tugas akhir ini.
2. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia.
3. Bapak Prof. Dr. Ir. Hari Purnomo, M.T. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., Ph.D. selaku Ketua Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
5. Bapak DThomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia.
6. Kedua orang tercinta ayahanda Sudarsiono dan Ibunda tercinta Rubiani Sarinawati.
7. Bapak Hanson Prihantoro Putro, S.T., M.T. selaku Dosen Pembimbing yang telah memberikan ilmu dan pencerahan kepada saya.
8. Ibu Arrie Kurniawardhani, S.Si., M.Kom. selaku Dosen Pembimbing Akademik yang telah banyak memberikan bantuan kepada saya.
9. Kakak saya Bagus Rianto dan adik saya Banu Agil Triyanto yang tidak henti-hentinya memberikan dukungan berupa moral, materil dan doa.
10. Teman seperjuangan Informatika angkatan 2018 yang telah banyak membantu saya.

Dengan adanya Tugas Akhir ini, penulis mengharapkan penelitian ini dapat dimanfaatkan sebaik mungkin dan dapat menambah wawasan bagi para pembaca. Laporan

tugas akhir ini disadari masih jauh dari kata sempurna. Maka dari itu penulis mengharapkan kritik dan saran yang membangun agar kedepannya dapat lebih baik lagi. Semoga dengan adanya laporan tugas akhir ini dapat bermanfaat bagi siapapun yang membaca.

Wassalamu'alaikum warahmatullahi wabarakatuh

Samarinda, 18 Juli 2022



(Bayu Dwiyanto)



SARI

Kuis biasanya menjadi penentuan apakah seseorang dapat memenuhi suatu standar yang ditentukan dalam sebuah pembelajaran. Untuk tujuan akademik, kuis digunakan sebagai bahan penilaian hasil belajar pada periode tertentu atau pada akhir suatu materi. Pada saat ini, 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 dapat 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 merupakan arsitektur perangkat lunak yang dibuat berorientasi layanan. SOA memiliki kelebihan yaitu memiliki sifat *loosely coupled*, *reuse* dan *interoperability*. Berdasarkan proses pengembangan sistem manajemen kuis dengan *Service Oriented Architecture*, maka dapat disimpulkan bahwa telah berhasil dirancang sebuah Sistem Manajemen Kuis dengan pendekatan *Service Oriented Architecture*. Hasil dari penelitian ini adalah pengembangan sistem manajemen kuis dilakukan dengan empat tahap sesuai dengan SOA *life-cycle* yaitu *model*, *assemble*, *deploy*, dan *manage*. Sistem manajemen kuis 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. Kemudian pengembangan sistem manajemen kuis dilakukan secara mandiri dan *customized* sehingga dapat memudahkan dalam penambahan fitur-fitur yang diinginkan sesuai dengan kebutuhan.

Kata kunci: Sistem Manajemen Kuis, *Service Oriented Architecture*, Perancangan Service, Pengembangan *Front-End*.

GLOSARIUM

<i>Front-End</i>	<i>Front-End</i> adalah sebuah istilah yang biasa digunakan untuk menggambarkan suatu tampilan dari komputer untuk berinteraksi dengan pengguna.
Kuis	Kuis adalah suatu bahan penilaian hasil belajar pada periode tertentu atau pada akhir suatu materi.
<i>Service</i>	<i>Service</i> atau layanan dalam perangkat lunak adalah sekumpulan fungsi, prosedur dan proses yang akan memberikan respon jika diminta oleh sebuah client.
<i>SOA</i>	<i>SOA</i> adalah gaya desain yang membangun model arsitektur IT yang berfokus pada peningkatan efisiensi, kelincahan dan produktivitas dengan menempatkan layanan sebagai sarana utama.



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan	5
BAB II LANDASAN TEORI	6
2.1 Sistem Manajemen Kuis	6
2.2 Front-End	7
2.3 Macam-Macam Arsitektur Pengembangan Perangkat Lunak	7
2.4 Service Oriented Architecture	8
2.4.1 Service	8
2.4.2 <i>Service Oriented Architecture Lifecycle</i>	9
2.4.3 Prinsip dan Sifat SOA	10
2.5 Kelebihan dan Kekurangan SOA	11
2.6 Kajian Pustaka	12
BAB III PENGEMBANGAN APLIKASI	13
3.1 Model	13
3.1.1 Identifikasi Proses Bisnis	13
3.1.2 Identifikasi Metodologi	14
3.1.3 Analisis Pengguna Perangkat Lunak	14
3.1.4 Identifikasi Kebutuhan Perangkat Lunak	15
3.1.5 Validasi Fitur	16
3.2 Assemble	19
3.2.1 Identifikasi Sistem	19
3.2.2 Identifikasi Service Provider	20
3.2.3 Identifikasi Service Consumer	20
3.2.4 Rancangan Arsitektur Sistem	20
3.2.5 Rancangan Use case Diagram	23
3.2.6 Rancangan Activity Diagram	25
3.2.7 Perancangan Detail Layanan	30
3.2.8 Perancangan Data	32
3.2.9 Tabel Data Sistem Manajemen Kuis	33

3.2.10	Diagram Relasi Basis Data Sistem Manajemen Kuis	39
3.2.1	Perancangan Antarmuka BackEnd.....	40
3.2.2	Perancangan Antarmuka Front-End.....	41
BAB IV HASIL DAN PEMBAHASAN.....		50
4.1	Deploy.....	50
4.1.1	Sitemap.....	50
4.1.2	Hasil Implementasi	52
4.2	Manage.....	64
4.2.1	Pengujian Front-End terkait Fungsionalitas Sistem.....	65
4.2.2	Pengujian Front End.....	66
4.2.3	Pengujian Kelayakan Pengembangan Lanjutan (<i>Back-End</i>)	70
4.2.4	Pengujian Prinsip SOA	71
BAB V KESIMPULAN DAN SARAN		74
5.1	Kesimpulan	74
5.2	Saran.....	74
DAFTAR PUSTAKA.....		75
LAMPIRAN		77



DAFTAR TABEL

Tabel 3.1 Identifikasi kebutuhan perangkat lunak.....	15
Tabel 3.2 Perbandingan fitur antara sistem manajemen kuis yang telah ada	16
Tabel 3.3 Wawancara responden Adita (Mahasiswa Informatika).....	17
Tabel 3.4 Wawancara responden Andri (Mahasiswa Informatika)	17
Tabel 3.5 Wawancara responden Andy (Mahasiswa Informatika).....	18
Tabel 3.6 Wawancara responden Syifa (Mahasiswa Teknik Kimia).....	18
Tabel 3.7 Wawancara responden Reza (Mahasiswa Informatika).....	18
Tabel 3.8 Wawancara responden Nabila (Mahasiswa Psikologi).....	19
Tabel 3.9 Keterangan <i>use case diagram</i>	24
Tabel 3.10 Perancangan detail layanan sistem pengelolaan kelas.....	30
Tabel 3.11 Perancangan detail layanan sistem pengelolaan kuis	31
Tabel 3.12 Perancangan detail layanan sistem pengelolaan nilai.....	31
Tabel 3.13 Struktur tabel kelas	33
Tabel 3.14 Struktur tabel mahasiswa	33
Tabel 3.15 Struktur tabel dosen	34
Tabel 3.16 Struktur tabel KelasMahasiswa	34
Tabel 3.17 Struktur tabel kelas	34
Tabel 3.18 Struktur tabel mahasiswa	35
Tabel 3.19 Struktur tabel dosen	35
Tabel 3.20 Struktur tabel kuis.....	35
Tabel 3.21 Struktur tabel KelasMahasiswa	36
Tabel 3.22 Struktur tabel pertanyaan.....	36
Tabel 3.23 Struktur tabel kelas	37
Tabel 3.24 Struktur tabel mahasiswa	37
Tabel 3.25 Struktur tabel dosen	37
Tabel 3.26 Struktur tabel kuis.....	38
Tabel 3.27 Struktur tabel KelasMahasiswa	38
Tabel 3.28 Struktur tabel nilai	38
Tabel 3.29 Perancangan Antarmuka BackEnd Sistem Manajemen Kuis	40
Tabel 4.1 Pengujian fungsionalitas sistem.....	65
Tabel 4.2 Nama Penguji.....	66
Tabel 4.3 Daftar aspek dan pertanyaan.....	67

Tabel 4.4 Keterangan warna respon.....	68
Tabel 4.5 Wawancara front-end developer.....	68
Tabel 4.6 Pengujian prinsip SOA	71



DAFTAR GAMBAR

Gambar 2.1 SOA Lifecycle.....	9
Gambar 3.1 Identifikasi Proses Bisnis	14
Gambar 3.2 Rancangan arsitektur sistem.....	21
Gambar 3.3 Service consumer	21
Gambar 3.4 Service provider	22
Gambar 3.5 Service registry	22
Gambar 3.6 Message broker (ESB)	23
Gambar 3.7 Use case diagram sistem pengelolaan kelas.....	23
Gambar 3.8 Use case diagram sistem pengelolaan kuis	24
Gambar 3.9 Use case diagram sistem pengelolaan nilai.....	24
Gambar 3.10 Activity diagram mahasiswa melihat kelas.....	25
Gambar 3.11 Activity diagram mahasiswa mengerjakan kuis.....	26
Gambar 3.12 Activity diagram mahasiswa melihat hasil kuis.....	27
Gambar 3.13 Activity diagram dosen membuat kelas	27
Gambar 3.14 Activity diagram dosen membuat kuis.....	28
Gambar 3.15 Activity diagram dosen mengelola nilai	29
Gambar 3.16 Activity diagram dosen mengundang mahasiswa.....	29
Gambar 3.17 Activity diagram mahasiswa bergabung dengan kelas	30
Gambar 3.18 Diagram relasi basis data sistem manajemen kuis.....	39
Gambar 3.19 Diagram relasi basis data sistem pengelolaan kuis	39
Gambar 3.20 Diagram relasi basis data sistem pengelolaan nilai.....	40
Gambar 3.21 Halaman daftar kelas dosen	42
Gambar 3.22 Halaman modal form buat kelas dan link undangan kelas.....	42
Gambar 3.23 Halaman undang mahasiswa.....	43
Gambar 3.24 Halaman daftar kelas mahasiswa	43
Gambar 3.25 Notifikasi sukses masuk kelas.....	44
Gambar 3.26 Halaman login mahasiswa dan dosen	44
Gambar 3.27 Halaman menampilkan form kuis	45
Gambar 3.28 Halaman notifikasi berhasil menyimpan form kuis	46
Gambar 3.29 Halaman menyerahkan kuis ke kelas	46
Gambar 3.30 Menampilkan kuis yang telah dipilih.....	47
Gambar 3.31 Halaman menyimpan kuis.....	47

Gambar 3.32 Halaman daftar mahasiswa dan kuis	48
Gambar 3.33 Halaman menyimpan ke database kuis	49
Gambar 3.34 Halaman hasil kuis	49
Gambar 4.1 Sitemap sistem pengelolaan kelas	51
Gambar 4.2 Sitemap sistem pengelolaan kuis	51
Gambar 4.3 Sitemap sistem pengelolaan nilai	52
Gambar 4.4 Halaman login	53
Gambar 4.5 Halaman daftar kelas dosen	54
Gambar 4.6 Halaman dosen membuat kelas	55
Gambar 4.7 Halaman dosen mengundang mahasiswa	56
Gambar 4.8 Halaman daftar kelas mahasiswa	57
Gambar 4.9 Halaman mahasiswa masuk kelas	58
Gambar 4.10 Halaman dosen membuat kuis	59
Gambar 4.11 Halaman dosen menyimpan kuis	60
Gambar 4.12 Halaman dosen menyerahkan kuis ke kelas	60
Gambar 4.13 Halaman mahasiswa mengerjakan kuis	61
Gambar 4.14 Halaman mahasiswa menyimpan kuis	62
Gambar 4.15 Halaman Dosen Mengelola Nilai	63
Gambar 4.16 Halaman Dosen Menyimpan Nilai	63
Gambar 4.17 Halaman melihat hasil kuis	64

BAB I PENDAHULUAN

1.1 Latar Belakang

Kuis biasanya digunakan sebagai bahan penilaian hasil belajar pada periode tertentu atau pada akhir suatu materi (Siregar, 2018). Dalam hal ini, kuis berfungsi sebagai bahan evaluasi pembelajaran bagi pengajar atas keefektifan dalam proses pembelajaran. Untuk tujuan yang berbeda kuis juga dapat menjadi penentuan apakah seseorang memenuhi standar yang telah ditetapkan atau tidak memenuhi standar yang telah ditetapkan. Kuis dapat 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 dapat menjadi bahan koreksi bagi masing-masing mahasiswa.

Sistem manajemen kuis yang memiliki sistem yang cukup besar mencakup berbagai fungsionalitas didalamnya membutuhkan arsitektur perangkat lunak yang cocok untuk diterapkan. *Service Oriented Architecture* merupakan arsitektur perangkat lunak yang berorientasi layanan. SOA cocok diterapkan pada sistem manajemen kuis karena SOA dapat membagi banyak fungsionalitas sistem ke masing-masing *service* atau layanan yang ada sehingga sistem tidak bekerja sendiri.

Berdasarkan permasalahan di atas akan dilakukan pengembangan sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*. Secara umum, pengembangan aplikasi memiliki dua metode yaitu *back end* dan *front end*. *Back-end* adalah tempat di mana suatu informasi diproses seperti menambah, menghapus dan mengubah data. *Back end* biasanya berhubungan langsung dengan server dan database. *Front-end* memegang peranan penting dalam pembangunan sebuah aplikasi, karena sebuah aplikasi harus memiliki fitur-fitur, tampilan dan berbagai komponen lainnya yang sesuai dengan kebutuhan pengguna.

Service Oriented Architecture atau SOA merupakan arsitektur perangkat lunak yang dibuat berorientasi layanan (Erl, 2005). 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 dapat 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 dapat digunakan di segala jenis *device* (Erl, 2005).

Dengan mengembangkan sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*, diharapkan dapat membantu para mahasiswa maupun dosen dalam melakukan proses pelaksanaan kuis, manajemen nilai dan menyampaikan konten pembelajaran. Selain itu diharapkan adanya sistem ini proses kuis dapat berjalan lebih efisien dan mengurangi kesalahan dalam perhitungan nilai. Sebagai batasan pada penelitian ini, pengembangan akan dikuatkan pada desain arsitektur perangkat lunaknya sehingga karena keterbatasan waktu, implementasi hanya dilakukan pada sisi *front-end* saja.

1.2 Rumusan Masalah

Berdasarkan beberapa masalah yang terdapat pada latar belakang penelitian, maka rumusan masalah dari penelitian ini adalah bagaimana mengembangkan *front-end* sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang sudah diuraikan di atas, maka permasalahan dalam penelitian ini perlu adanya batasan masalah sebagai berikut:

- a. Sistem yang dikembangkan berbasis web.
- b. Penelitian ini berfokus pada pengembangan *front-end*-nya.
- c. Tampilan *front end* yang dibangun adalah manajemen kuis yang akan digunakan oleh mahasiswa dan dosen.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menghasilkan dasar rancangan sistem manajemen kuis dengan pendekatan *Service Oriented Architecture* yang ditujukan bagi mahasiswa dan dosen untuk memudahkan dalam proses pelaksanaan kuis dan manajemen nilai sehingga proses pelaksanaan kuis dapat berjalan lebih efisien dan mengurangi kesalahan dalam perhitungan

nilai. Kemudian melanjutkan implementasi pengembangan sistem manajemen kuis ini dari sisi *front-end*.

1.5 Manfaat Penelitian

Setelah tujuan penelitian tercapai, manfaat yang diharapkan yaitu dapat menjadi dasar pengembangan sebuah sistem manajemen kuis dengan pendekatan *Service Oriented Architecture* yang utuh. *Front-end* yang dihasilkan juga diharapkan dapat memberikan gambaran bagaimana nantinya *back-end* dikembangkan dengan *Service Oriented Architecture*. Manfaat dari pengembangan sistem manajemen dengan menggunakan SOA yaitu arsitektur yang bersifat *service oriented*, yaitu arsitektur yang membagi suatu masalah ke dalam berbagai *service* kecil yang saling bekerja sama. Dengan menggunakan SOA suatu aplikasi tidak lagi dipandang dari sisi teknologi, data, lingkungan implementasi melainkan dipandang dari *service* yang disediakan. Setelah itu dengan adanya sistem manajemen kuis dengan pendekatan *Service Oriented Architecture*, maka dapat membantu mahasiswa dan dosen dalam melakukan proses manajemen kuis.

1.6 Metodologi Penelitian

a. Pengumpulan Data

Metode Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan berdasarkan jenis data dan sumbernya. Data objektif dan relevan dengan pokok permasalahan penelitian merupakan indikator keberhasilan suatu penelitian.

1. Studi literatur

Tahap studi literatur dilakukan dengan cara meneliti dan memahami buku-buku, dokumen atau sumber tertulis lainnya yang berkaitan dengan penelitian. F Kapojos, H.F. Wowor, A.M. Rumagit, A.P.R Wowor (2012) dalam penelitiannya menjelaskan penerapan SOA dalam aplikasi informasi akademik memudahkan penggunaan data-data mahasiswa/dosen/matakuliah dan data lainnya yang digunakan kembali oleh aplikasi lain atau sistem informasi lain dalam organisasi. Kurniawan, D., Andrian, R., & Ertina, L. (2016) dalam penelitiannya menjelaskan bahwa sistem ujian online dapat mempermudah kerja pihak sekolah dalam pengoreksian hasil ujian online dan mendapatkan nilai yang akurat.

2. Observasi Aplikasi sejenis

Tahap pengumpulan data dengan melakukan observasi aplikasi sejenis dilakukan untuk membandingkan dan mengeksplorasi fitur yang telah ada pada aplikasi Kahoot, Quizizz dan Google Classroom.

b. Identifikasi Metodologi

Tahap ini dilakukan untuk mengetahui dan mempelajari metode SOA yaitu dengan melakukan studi literatur seperti membaca penelitian terdahulu. Dengan membaca penelitian yang telah dilakukan terdahulu, dapat diketahui apa itu SOA, prinsip SOA dan sifat-sifat SOA. Dari studi literatur juga dapat dipahami bagaimana SOA dapat diterapkan dalam berbagai proses bisnis pada perangkat lunak.

c. Pengembangan Aplikasi Berbasis SOA

Daur hidup SOA adalah sebuah proses iterative dalam proses pengembangan arsitektur perangkat lunak (McBride, 2007). Tahapan daur hidup SOA yaitu: 1) Model; 2) Assemble; 3) Deploy; dan 4) Manage. Hasil akhir dari pengembangan ini adalah sebuah implementasi halaman front-end sistem manajemen kuis yang selanjutnya dibutuhkan implementasi pada sisi back-end.

d. Evaluasi

Untuk memastikan proses SOA berjalan dengan tepat dan menghasilkan produk yang benar, penulis harus dapat membaca dan memahami kaidah, prinsip dan sifat SOA yang telah banyak ditemukan pada berbagai literature. Selain membaca dan memahami, penulis juga perlu melakukan pengujian perangkat lunak dengan metode black box testing atau juga dapat menggunakan test case testing. Pengujian ini bertujuan untuk mengetahui apabila perangkat lunak mempunyai kesalahan dan juga menjaga kualitas perangkat lunak sesuai dengan fitur yang telah direncanakan.

Pengujian *front-end* dapat dilakukan dengan *Black Box testing*. *Black Box Testing* dilakukan sebagai pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. *Black Box testing* ini lebih menguji ke tampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh pengguna.

1.7 Sistematika Penulisan

Sistematika Penulisan ini dalam menyelesaikan penelitian ini adalah:

BAB I PENDAHULUAN

Pada bab ini berisi tentang latar belakang pembuatan laporan, perumusan masalah, ruang lingkup, tujuan penulisan, manfaat penulisan, metodologi penulisan dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini memuat tentang landasan teori sebagai konsep dasar yang mendukung penelitian yang akan dilakukan penulis.

BAB III PENGEMBANGAN APLIKASI

Pada bab ini menjelaskan tentang analisis kebutuhan, *entity relational database*, *use case diagram*, dan *activity diagram* sebagai alur dari website serta ada desain interface sebagai tampilan dari website.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini menyajikan hasil tampilan program, penjelasan dan pembahasan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan dan saran tentang hasil yang telah diperoleh dan saran-saran yang memungkinkan untuk pengembangan sistem lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Sistem Manajemen Kuis

Sistem manajemen adalah suatu kumpulan dari banyak komponen yang saling terhubung dan bekerja sama untuk mencapai suatu tujuan dengan menggunakan input dan menghasilkan sebuah output (O'brien, 2002). Dalam definisi yang lain, sistem manajemen adalah sebuah sistem yang melakukan pengolahan transaksi yang dibutuhkan oleh organisasi dan juga memberikan dukungan informasi dan pengolahan untuk menjalankan tujuan organisasi (T Sutabri, 2016). Sistem manajemen berbasis web dapat diartikan sebagai sebuah sistem informasi yang menggunakan teknologi web atau internet untuk memberikan informasi dan layanan kepada pengguna. Sebuah sistem manajemen web biasanya terdiri dari satu atau lebih aplikasi web yang masing-masing komponen mempunyai fungsi tertentu. Dimana masing-masing komponen tersebut saling mendukung satu sama lain guna mencapai fungsi sistem informasi secara umum.

Kuis adalah suatu metode evaluasi pembelajaran untuk mengambil nilai pada periode tertentu atau selesainya suatu materi. Kuis memiliki fungsi untuk mengukur kemampuan mahasiswa dalam memahami suatu pengetahuan atau materi dan memberikan umpan balik bagi pengajar. Saat ini pelaksanaan kuis yang umum dilaksanakan pada kebanyakan universitas masih menggunakan cara konvensional yaitu dengan menggunakan kertas sebagai media kuis. Pelaksanaan kuis dengan cara konvensional memiliki banyak kekurangan seperti memerlukan media kertas yang banyak, memerlukan waktu pemeriksaan yang lama dan mengeluarkan biaya yang tidak sedikit. Tidak jarang setelah selesai pelaksanaan kuis mahasiswa tidak mengetahui nilai yang didapatnya, menyebabkan mahasiswa menjadi susah melakukan evaluasi terhadap kuis yang dikerjakannya.

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 (Google Classroom, 2021). Kahoot adalah sebuah sistem respon materi yang mengajak siswa menggunakan sebuah website seperti game (Kahoot, 2021). Para siswa tidak perlu membuat akun untuk menggunakan kahoot, siswa hanya membutuhkan kode dari setiap kuis yang dibuat dan dapat mengaksesnya dari berbagai device seperti handphone, *ipad*, dan laptop. Terdapat banyak pilihan dalam membuat kuis di kahoot

seperti mengunggah video, gambar, dan musik untuk mendorong siswa untuk mempercepat pemikiran.

Quizizz adalah platform kuis bagi 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 (Quizizz, 2021). Sebagai seorang guru, anda dapat melakukan penilaian formatif, memberikan pekerjaan rumah, dan melakukan interaksi dengan siswa. Quizizz dapat diakses melalui berbagai perangkat seperti, laptop, handphone, *tablet*, dan komputer.

2.2 Front-End

Front-End adalah sebuah istilah yang biasa digunakan untuk menggambarkan suatu tampilan dari komputer untuk berinteraksi dengan pengguna. *Front-End* adalah bagian dari antarmuka suatu sistem yang berfokus pada keindahan sebuah tampilan, *layout*, pemilihan warna, typography dan hal lainnya. Pembangunan *Front-End* biasanya menggunakan komponen dari *css*, *javascript*, dan *html* (Morris, 2021).

Cascading Style Sheets atau *css*, yaitu bahasa yang digunakan untuk memberikan konten gaya penampilan yang bagus. Penggunaan *css* akan mempermudah kerja karena *css* dapat mengontrol layout multiple web pages hanya dengan satu pengaturan. *Css* digunakan untuk mendefinisikan style halaman web termasuk design, layout, dan variasi tampilan pada berbagai perangkat dan ukuran layer. *Hyper Text Markup Language* atau *HTML* merupakan bahasa pemrograman dasar dalam membuat website. *HTML* belum dapat dikatakan sebagai bahasa pemrograman karena *html* tidak memiliki hal-hal yang dibutuhkan oleh bahasa pemrograman yaitu logika, *html* hanya memberikan output, maka dari itu *html* di ibaratkan sebagai pondasi atau struktur dari web. *Javascript* adalah bahasa pemrograman untuk sisi client atau client side. *Javascript* sendiri tujuannya di buat untuk memperkaya fitur pada website agar lebih dinamis dan interaktif.

2.3 Macam-Macam Arsitektur Pengembangan Perangkat Lunak

Terdapat berbagai macam arsitektur dalam proses pengembangan perangkat lunak. Masing-masing arsitektur memiliki fungsi yang berbeda tergantung dengan penggunaannya. Salah satu arsitektur dalam dalam pengembangan perangkat lunak adalah *model*, *view*, dan *controller* (MVC). MVC didefinisikan sebagai arsitektur dalam pengembangan perangkat lunak yang memisahkan logika bisnis dari input dan presentasi logika yang terkait dengan

tampilan antarmuka suatu aplikasi. MVC mengikuti pendekatan yang paling umum dari layering yaitu sebuah logika yang membagi kode ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode.

MVC sebagai arsitektur pengembangan perangkat lunak belum cocok digunakan dalam pengembangan sistem manajemen kuis. MVC belum bisa diterapkan dalam sistem manajemen kuis karena belum dapat menampung berbagai layanan atau *service* yang akan dimiliki oleh sistem. Sehingga dalam pengembangan sistem manajemen kuis akan digunakan *Service Oriented Architecture*. SOA digunakan karena memiliki arsitektur yang bersifat *service oriented*, yaitu arsitektur yang membagi suatu masalah ke dalam berbagai *service* kecil yang saling bekerja sama. Dengan menggunakan SOA suatu aplikasi tidak lagi dipandang dari sisi teknologi, data, lingkungan implementasi melainkan dipandang dari *service* yang disediakan. Dengan cara ini suatu aplikasi dapat berkomunikasi dan bekerja sama dengan aplikasi yang lain tanpa memandang teknologi, data, lingkungan implementasi dari aplikasi tersebut. Sehingga SOA cocok digunakan dalam pengembangan sistem manajemen kuis yang membutuhkan banyak *service* atau layanan.

2.4 Service Oriented Architecture

Service Oriented Architecture atau dapat disebut dengan SOA adalah gaya desain yang membangun model arsitektur IT yang berfokus pada peningkatan efisiensi, kelincahan dan produktivitas dengan menempatkan layanan sebagai sarana utama (Erl, 2007).

Pada dasarnya SOA adalah sebuah arsitektur aplikasi yang mengatur setiap layanan dan fungsi yang dideskripsikan dalam suatu Bahasa agar dapat 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.

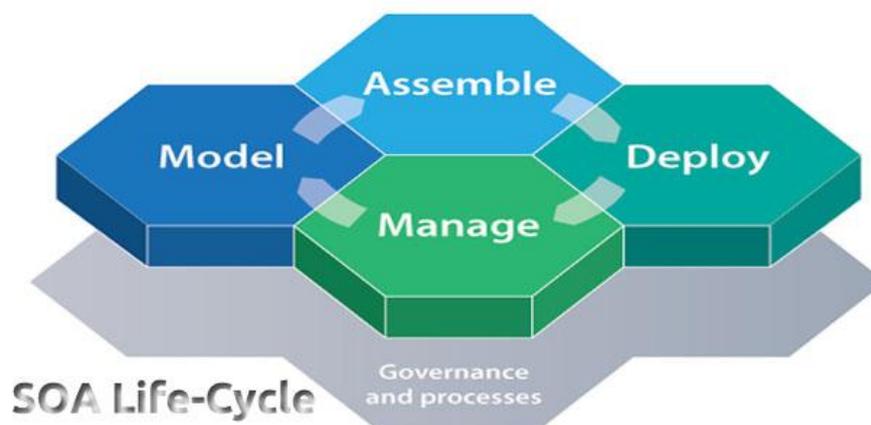
2.4.1 Service

Service atau layanan dalam perangkat lunak adalah sekumpulan fungsi, prosedur, dan proses yang akan memberikan respon jika diminta oleh sebuah *client*. Dalam definisi yang lain layanan adalah unit fungsionalitas perangkat lunak yang berdiri sendiri, atau serangkaian fungsi, yang dirancang untuk menyelesaikan tugas tertentu seperti mengambil informasi tertentu atau menjalankan operasi (Redhat, 2020).

SOA adalah salah satu contoh arsitektur perangkat lunak yang berorientasi *service* atau layanan. *Service* pada SOA memiliki arti setiap fungsi bisnis berjalan terpisah (IBM, 2021), Sebagai contoh dalam sebuah sistem manajemen keuangan bank terdapat beberapa *service* yang saling terpisah seperti *service* kredit pelanggan, *service* pembayaran pinjaman atau *service* hipotek, *service* tersebut menggunakan prinsip *loose coupling*, yang memiliki arti bahwa setiap layanan yang dipanggil tidak memerlukan layanan lainnya untuk mendapatkan data atau informasi, prinsip *service* tersebut berguna untuk mengurangi ketergantungan antar *service*.

2.4.2 *Service Oriented Architecture Lifecycle*

SOA mempunyai tahapan-tahapan atau siklus hidup yang menjadi petunjuk dalam dalam merancang sebuah arsitektur aplikasi. Siklus hidup SOA dapat dilihat pada Gambar 2.1.



Gambar 2.1 SOA Lifecycle

1. *Model*: Mengumpulkan kebutuhan bisnis, merancang, mensimulasikan dan mengoptimalkan proses bisnis yang diinginkan pengguna.
2. *Assemble*: Menerapkan proses bisnis yang dioptimalkan dengan menggabungkan layanan yang baru dibuat dan yang sudah ada untuk membentuk aplikasi
3. *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.
4. *Manage*: Setelah aplikasi diterapkan, pengguna dapat mengelola dan memantau aplikasi dan sumber daya dari perspektif IT 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.

2.4.3 Prinsip dan Sifat SOA

Terdapat beberapa aspek kunci pada prinsip SOA (Thomas, 2005), yaitu:

1. *Loose coupling*, yaitu bahwa *service* tersebut mempertahankan sebuah hubungan yang meminimalisir ketergantungan dan mereka hanya perlu menjaga kesadaran antar satu sama lain.
2. *Service contract*, *service* melekat dan taat pada sebuah kesepakatan komunikasi, yang didefinisikan secara kolektif oleh satu atau lebih deskripsi *service* dan dokumen yang berhubungan.
3. *Autonomy*, bahwa *service* mempunyai kendali berdasarkan logika yang dienkapsulasi.
4. *Abstraction*, di luar apa yang dideskripsikan pada *service contract*, *services* menyembunyikan logika dari dunia luar.
5. *Reusability*, logika terbagi menjadi *service* dengan tujuan untuk digunakan kembali.
6. *Composability*, kumpulan dari *service* dapat dikoordinasikan dan dihimpun untuk membentuk *services* yang berbeda.
7. *Statelessness*, *service* meminimalisasi sifat berpegang teguh pada informasi tertentu untuk sebuah aktivitas.
8. *Discoverability*, *services* dirancang dengan sifat yang deskriptif sehingga mereka dapat ditemukan dan ditentukan dengan menggunakan mekanisme-mekanisme penemuan yang ada.

SOA memiliki sifat-sifat sebagai berikut (Tarigan, 2010):

1. SOA bersifat *loosely coupled* (tingkat ketergantungannya rendah), *high interoperable* (mudah dioperasikan), *reusable* (dapat digunakan kembali), dan *interoperability* (antar platform).
2. Tidak semua kebutuhan komputasi dapat dipenuhi sendiri, terlalu berat dari sisi biaya dan usaha.
3. Independensi diperlukan karena sering kali sebagian fungsionalitas yang diperlukan berada di luar domain kendali pemakai.
4. SOA menggunakan definisi antar muka yang mengenkapsulasi semua implementasi yang ada.

5. SOA bersifat *behind the scene*, SOA tidak terlihat secara langsung oleh client, SOA dihadapkan pada client melalui antar muka client.
6. SOA merupakan suatu *service* yang “hanya menunggu” (*listen*) secara terus-menerus untuk digunakan.

Berikut adalah contoh penerapan SOA yang digunakan dalam teknologi web *service*:

1. *SOAP (Simple Object Access Protocol)*

SOAP atau lebih dikenal juga *Service Oriented Architecture Protocol*, adalah sebuah protokol yang digunakan untuk melakukan pertukaran dokumen *XML* melalui jaringan komputer, biasanya teknologi SOAP ditemukan pada web *service* (Curbera, 2002).

2. *CORBA (Common object request broker architecture)*.

CORBA adalah sebuah standar yang dikeluarkan oleh *Object Management Group* (OMG) yang memungkinkan komponen-komponen perangkat lunak ditulis dalam bahasa II-15 pemrograman dan dijalankan di atas platform yang beragam untuk saling berkomunikasi (*interoperate*) .

3. *REST (Representational State transfer)*

REST adalah gaya arsitektur perangkat lunak untuk sistem *hypermedia* terdistribusi seperti *world wide web*. REST memiliki tujuan untuk memberikan gambaran bagaimana sebuah aplikasi web berperilaku: sebuah jaringan dari halaman-halaman web, tujuan ketika user memilih link (*state transitions*), hasil dari halaman selanjutnya (mewakili status berikutnya) yang akan ditransfer kepada user (*Fielding*).

2.5 Kelebihan dan Kekurangan SOA

Sebagai suatu arsitektur perangkat lunak SOA memiliki beberapa kelebihan dan kekurangan sebagai berikut:

Kelebihan:

1. Lokasi mandiri. Tidak masalah di mana layanan itu berada. Layanan dapat dipublikasikan di satu server atau beberapa server yang berbeda. Permintaan pengguna akan tetap berfungsi dengan baik.
2. Dapat digunakan kembali. Layanan dapat digunakan kembali terlepas dari interaksi sebelumnya dengan layanan lain.
3. Peningkatan skalabilitas. SOA dapat dengan mudah meningkatkan skalabilitas sistem karena beberapa lapisan dari satu layanan dapat berjalan secara bersamaan di server yang berbeda.

4. Peluang pengembangan paralel. Berkat arsitektur berbasis lapisan, pengembang dapat bekerja pada layanan independen. Pengembangan secara paralel akan meningkatkan produktivitas dan juga memungkinkan bisnis untuk mengurangi biaya.

Kekurangan:

1. Beban yang lebih besar dan peningkatan waktu respon sistem. Dalam SOA, setiap interaksi antar layanan diikuti dengan validasi dari semua parameter input. Akibatnya, beban respon sistem menjadi sangat berat.
2. Ukuran software. SOA membagi layanan menjadi layanan yang lebih kecil akan mengakibatkan ukuran software menjadi besar.

2.6 Kajian Pustaka

Penelitian yang pertama Kapojos, (2012) yang mengimplementasikan *Service-oriented architecture* dengan *web service* untuk aplikasi informasi akademik, 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 yang kedua oleh Pakusadewa, (2015) yang memiliki tujuan untuk mengetahui hasil implementasi SOA pada sistem pengelolaan data asuransi BPJS". 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 ketiga yang dilakukan oleh Gumelar, (2017) yang bertujuan untuk merancang sistem informasi berbasis SOA dengan menerapkan kriteria *interoperability* dan *reusable*. Penelitian ini menghasilkan bahwa sifat *reusable* SOA sangat berguna untuk *service* karena dapat digunakan kembali pada sistem yang lain. Kemudian sifat *interoperability* yang dimiliki SOA terbukti dapat digunakan dalam berbagai *device pc* maupun *mobile*.

Terakhir, penelitian yang dilakukan oleh Trichahyo, (2017) yang menerangkan bahwa penggunaan SOA terbukti mampu mengintegrasikan layanan *e-learning* dengan sistem informasi sekolah berbasis ERP. 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.

BAB III

PENGEMBANGAN APLIKASI

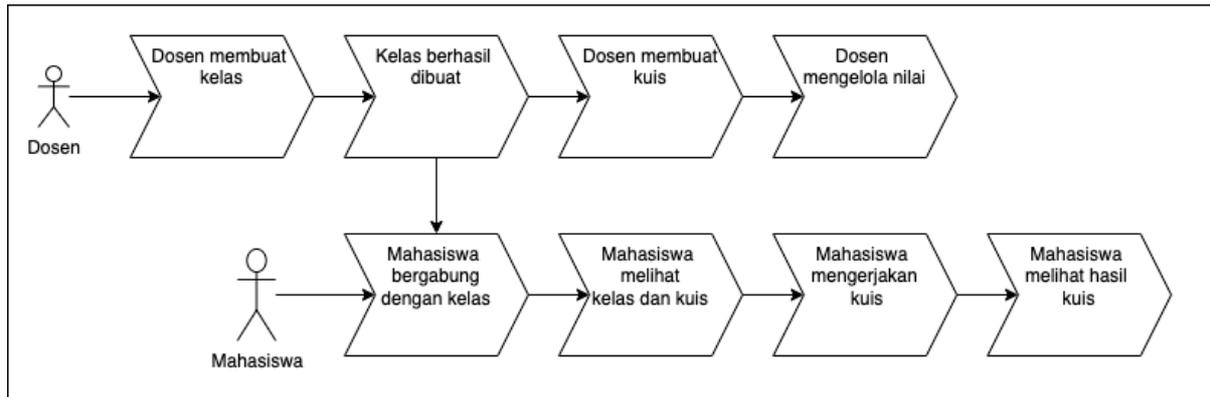
3.1 Model

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

3.1.1 Identifikasi Proses Bisnis

Kuis merupakan suatu metode evaluasi pembelajaran untuk mengambil nilai pada periode tertentu atau selesainya suatu materi. Kuis dibutuhkan untuk mengukur kemampuan mahasiswa dalam memahami suatu pengetahuan atau materi dan memberikan umpan balik bagi pengajar. Saat ini pelaksanaan kuis yang umum dilaksanakan pada kebanyakan universitas masih menggunakan cara konvensional yaitu dengan menggunakan kertas sebagai media kuis.

Proses pertama yaitu dosen dapat membuat kelas sesuai dengan kelas yang akan diampunya, selanjutnya setelah kelas berhasil dibuat, mahasiswa dapat bergabung dengan kelas tersebut, di dalam kelas tersebut mahasiswa dapat melihat kuis yang tersedia, selanjutnya dosen dapat 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 dapat melakukan pengelolaan nilai terhadap setiap kuis yang telah selesai dikerjakan dan yang terakhir adalah para mahasiswa dapat melihat hasil kuis yang telah dinilai oleh dosen. Untuk lebih jelas terkait identifikasi proses bisnis sistem manajemen kuis dapat dilihat pada Gambar 3.1.



Gambar 3.1 Identifikasi Proses Bisnis

3.1.2 Identifikasi Metodologi

Service Oriented Architecture (SOA) adalah suatu arsitektur dalam pengembangan perangkat lunak dengan melakukan pendekatan layanan. SOA memungkinkan hubungan dan pertukaran data atau informasi antar layanan menjadi mudah. Sifatnya yang *loosely coupled* (tingkat ketergantungan antar komponen rendah), *interoperable* (mudah dioperasikan), *reusable* (dapat digunakan kembali), dan *interoperability* (dapat berkomunikasi antar platform) menjadikan SOA handal dalam pengembangan dan pengintegrasian data dan informasi.

SOA melakukan pendekatan dengan membagi masalah besar menjadi sekumpulan service kecil yang bertujuan untuk menyelesaikan permasalahan tertentu. Fungsi-fungsi pada sistem manajemen kuis harus dapat mengakses basis data dan melayani proses bisnis manajemen kuis sehingga dapat berinteraksi satu dengan lainnya tanpa perlu memperhatikan lokasi dan platform apa yang digunakan oleh *service* lainnya pada saat berjalan. Dengan demikian sistem manajemen kuis yang memiliki banyak layanan dapat dengan mudah menjalankan layanan untuk memenuhi kebutuhan bisnis.

3.1.3 Analisis Pengguna Perangkat Lunak

Diidentifikasi bahwa calon pengguna sistem ini terdiri dari dua peran, yaitu Mahasiswa dan Dosen, dengan penjelasan sebagai berikut:

1. Mahasiswa

Peran mahasiswa pada sistem manajemen kuis adalah sebagai pengguna sistem yang dapat melakukan login, mengerjakan kuis, dan melihat hasil dari setiap kuis yang telah dikerjakan.

2. Dosen

Dosen merupakan aktor yang berperan penting dalam jalannya sistem manajemen kuis. Peran dosen nantinya akan meliputi membuat kelas, membuat kuis, mengelola nilai kuis, dan menyerahkan nilai kepada setiap mahasiswa.

3.1.4 Identifikasi Kebutuhan Perangkat Lunak

Kebutuhan fungsionalitas pada Tabel 3.1 didapatkan dari kajian yang telah dilakukan pada bab dua. Tiga sistem manajemen kuis yang telah ada seperti Kahoot, Quizizz, dan Google Classroom menjadi referensi dalam pengembangan kebutuhan fungsional pada sistem manajemen kuis yang akan dikembangkan. Untuk lebih jelas dapat dilihat pada Tabel 3.1

Tabel 3.1 Identifikasi kebutuhan perangkat lunak

No	Fungsionalitas	Kahoot	Quizizz	Classroom
1	Login	Ada	Ada	Ada
2	Membuat kelas	Ada	Ada	Ada
3	Melihat kelas	Tidak ada	Tidak ada	Ada
4	Membuat kuis	Ada	Ada	Ada
5	Mengerjakan kuis	Ada	Ada	Ada
6	Menilai kuis	Ada	Ada	Ada
7	Melihat hasil kuis	Ada	Ada	Ada

Dari fungsionalitas pada Tabel 3.1, perangkat lunak masih perlu fungsionalitas pendukung yaitu registrasi akun mahasiswa. Registrasi akun mahasiswa akan dirancang menggunakan layanan *google identity*. *Google identity* merupakan sebuah layanan lintas platform yang dapat digunakan diaplikasi web dan Android, layanan milik google ini mendukung dan menyederhanakan proses berbagai jenis *kredensial* (Google, 2021).

Tabel 3.2 adalah perbandingan fitur antara sistem manajemen kuis yang telah ada. Fitur yang telah diidentifikasi pada sistem manajemen kuis Kahoot, Quizizz, dan Google Classroom akan dikembangkan dan ditambahkan pada sistem manajemen kuis.

Tabel 3.2 Perbandingan fitur antara sistem manajemen kuis yang telah ada

No	Fitur	Kahoot	Quizizz	Classroom
1	Tampilan soal dan jawaban	Menggunakan layar proyektor atau layar komputer	Ditampilkan pada masing-masing perangkat pengguna	Ditampilkan pada masing-masing perangkat pengguna
2	Progres menjawab setiap soal	Semua peserta harus menyelesaikan soal selanjutnya setelah peserta menjawab soal sebelumnya atau sesuai waktu yang telah diizinkan	Setiap peserta dapat menjawab soal selanjutnya ketika peserta sudah atau belum menyelesaikan soal sebelumnya	Setiap peserta dapat menjawab soal selanjutnya ketika peserta sudah atau belum menyelesaikan soal sebelumnya
3	Informasi dalam pertanyaan	Maksimal 95 karakter dapat memasukan gambar, audio, dan video	Tidak ada batasan karakter dapat memasukan gambar, audio, dan video	Tidak ada batasan karakter dapat memasukan gambar, audio, dan video
4	Kebutuhan teknis	Aplikasi membutuhkan layar besar di mana semua peserta dapat membaca pertanyaan, layar proyektor, smartboard, dan perangkat yang terhubung ke internet seperti smartphone, tablet, laptop atau komputer.	Perangkat yang terhubung ke Internet seperti smartphone, tablet, laptop atau komputer.	Perangkat yang terhubung ke Internet seperti smartphone, tablet, laptop atau komputer.

Setelah dilakukan perbandingan fitur antara sistem manajemen kuis yang telah ada, akan dilakukan perancangan sistem manajemen yang akan memiliki perbedaan dari sistem yang telah ada. Sistem manajemen kuis yang dirancang akan memiliki fitur pengembangan sistem secara mandiri dan *custom*. Pengembangan sistem manajemen kuis secara *custom* dapat memudahkan dalam penambahan fitur-fitur yang diinginkan sesuai dengan kebutuhan. Kemudian juga pengembangan secara mandiri dapat menghindarkan komersialisasi data dari pihak penyedia layanan sistem manajemen kuis.

3.1.5 Validasi Fitur

Telah dilakukan survei untuk mendapatkan data dasar yang digunakan untuk mendapatkan gambaran umum yang berguna untuk proses pengembangan sistem manajemen kuis. Proses survei dimulai dengan mempersiapkan pertanyaan yang telah dibuat berdasarkan

identifikasi kebutuhan fungsional dan non fungsional. Survei dilakukan dengan wawancara secara lisan dengan responden melalui platform *discord*.

Survei kebutuhan fitur fungsionalitas aplikasi disebarakan kepada 6 responden dari tanggal 27-Oktober-2021 sampai dengan tanggal 4-November-2021. Survei tersebut menyatakan bahwa keseluruhan responden atau 100% responden setuju dan puas dengan fitur fungsionalitas yang ditawarkan pada survei, fitur yang ditawarkan pada survei yaitu fitur login, melihat kelas dan kuis, mengerjakan kuis dan melihat hasil kuis. Script/rangkuman wawancara dapat dilihat di Tabel 3.3 sampai dengan Tabel 3.8.

Tabel 3.3 Wawancara responden Adita (Mahasiswa Informatika, UII)

No	Pertanyaan	Jawaban
1	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat kelas dan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
2	Sebagai mahasiswa apakah anda setuju terhadap fitur mengerjakan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
3	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat hasil kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
4	Apakah ada fitur lain yang ingin ditambahkan?	Tidak ada

Tabel 3.4 Wawancara responden Andri (Mahasiswa Informatika, UII)

No	Pertanyaan	Jawaban
1	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat kelas dan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
2	Sebagai mahasiswa apakah anda setuju terhadap fitur mengerjakan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
3	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat hasil kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
4	Apakah ada fitur lain yang ingin ditambahkan?	Tidak ada

Tabel 3.5 Wawancara responden Andy (Mahasiswa Informatika, UII)

No	Pertanyaan	Jawaban
1	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat kelas dan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
2	Sebagai mahasiswa apakah anda setuju terhadap fitur mengerjakan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
3	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat hasil kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
4	Apakah ada fitur lain yang ingin ditambahkan?	Tidak ada

Tabel 3.6 Wawancara responden Syifa (Mahasiswa Teknik Kimia, Universitas Brawijaya)

No	Pertanyaan	Jawaban
1	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat kelas dan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
2	Sebagai mahasiswa apakah anda setuju terhadap fitur mengerjakan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
3	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat hasil kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
4	Apakah ada fitur lain yang ingin ditambahkan?	Tidak ada

Tabel 3.7 Wawancara responden Reza (Mahasiswa Informatika, UII)

No	Pertanyaan	Jawaban
1	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat kelas dan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
2	Sebagai mahasiswa apakah anda setuju terhadap fitur mengerjakan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
3	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat hasil kuis yang akan dirancang pada sistem manajemen kuis?	Setuju

4	Apakah ada fitur lain yang ingin ditambahkan?	Tidak ada
---	---	-----------

Tabel 3.8 Wawancara responden Nabila (Mahasiswa Psikologi, UII)

No	Pertanyaan	Jawaban
1	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat kelas dan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
2	Sebagai mahasiswa apakah anda setuju terhadap fitur mengerjakan kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
3	Sebagai mahasiswa apakah anda setuju terhadap fitur melihat hasil kuis yang akan dirancang pada sistem manajemen kuis?	Setuju
4	Apakah ada fitur lain yang ingin ditambahkan?	Tidak ada

3.2 Assemble

Assemble adalah tahap menggabungkan layanan yang baru dibuat dan yang sudah ada untuk membentuk aplikasi. Pada tahap ini berisi Activity Diagram, DB Diagram, Antarmuka, dan *Use case diagram*.

3.2.1 Identifikasi Sistem

Dari kebutuhan fungsional yang telah diidentifikasi, 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.

3.2.2 Identifikasi Service Provider

Sistem manajemen kuis akan terbagi menjadi tiga. Sistem pertama yaitu sistem pengelolaan kelas yang mempunyai nama *service* kelas, *service* kelas akan digunakan oleh mahasiswa dan dosen untuk mengelola kelas. Dosen dapat membuat kelas dan mengundang mahasiswa sedangkan mahasiswa dapat bergabung dan melihat kelas.

Sistem kedua yaitu sistem pengelolaan kuis yang memiliki nama *service* kuis, *service* kuis digunakan oleh mahasiswa dan dosen dengan masing-masing fungsi yang berbeda. Dosen menggunakan *service* kuis sebagai media pembuatan kuis sedangkan mahasiswa menggunakan *service* kuis sebagai media untuk mengerjakan kuis.

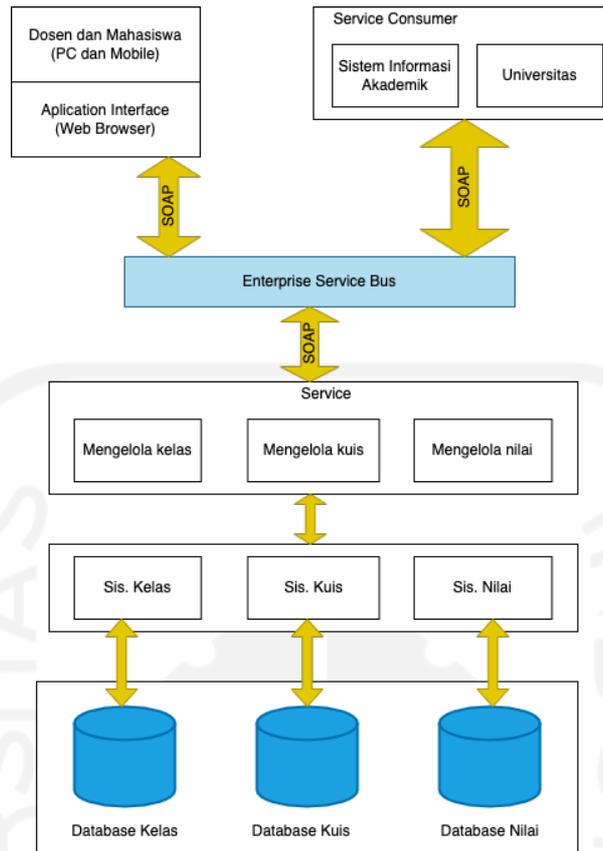
Sistem terakhir yang digunakan adalah sistem pengelolaan nilai, sistem ini mempunyai *service* nilai, *service* nilai akan digunakan oleh mahasiswa dan dosen. *service* kuis memiliki fungsionalitas menilai kuis yang digunakan oleh dosen dan melihat nilai yang akan digunakan oleh mahasiswa.

3.2.3 Identifikasi Service Consumer

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*.

3.2.4 Rancangan Arsitektur Sistem

Rancangan arsitektur sistem manajemen kuis terdiri dari empat lapisan yaitu *service consumer* yang mempunyai beberapa aplikasi. Kemudian message broker menggunakan ESB berfungsi untuk menyampaikan *service*. *Service registry* dan *service provider* digunakan untuk menjalankan sistem. Untuk lebih jelasnya dapat dilihat pada Gambar 3.2 rancangan arsitektur sistem manajemen kuis.

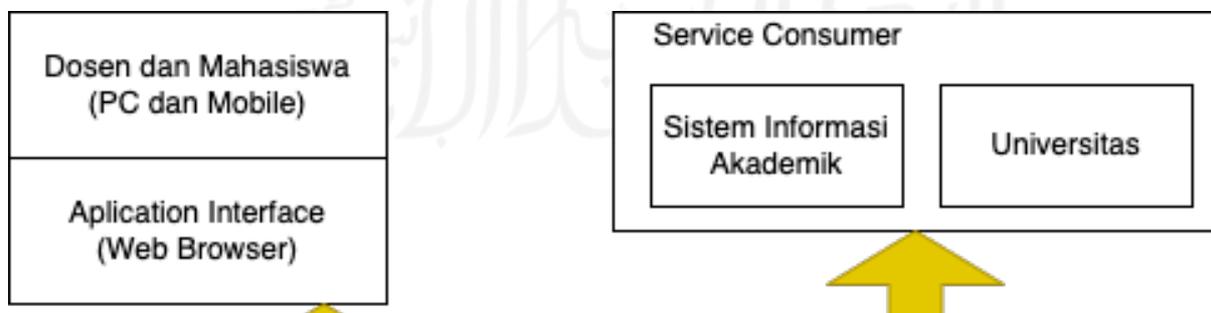


Gambar 3.2 Rancangan arsitektur sistem

Penjelasan rancangan arsitektur sistem Gambar 3.2:

A. Service Consumer

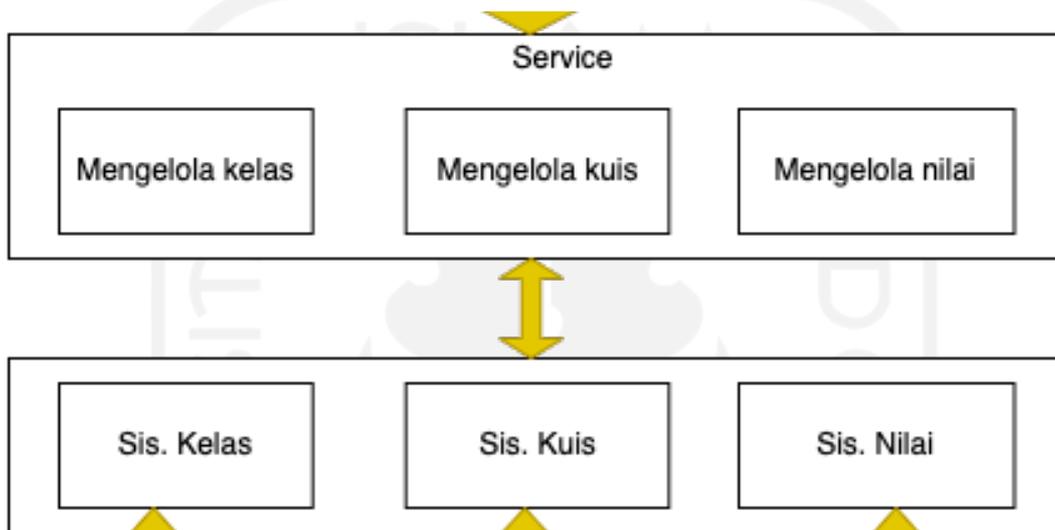
Service Consumer merupakan user yang mempunyai akses untuk menggunakan *service* yang tersedia untuk mengakses segala informasi yang dibutuhkan yang telah disediakan oleh *service provider*. Selain user, *service consumer* juga terdapat pada platform yang akan digunakan oleh user untuk mengakses *service* seperti pada platform *PC* dan *Mobile*. Untuk lebih jelas dapat dilihat pada Gambar 3.3.



Gambar 3.3 Service consumer

B. Service Provider

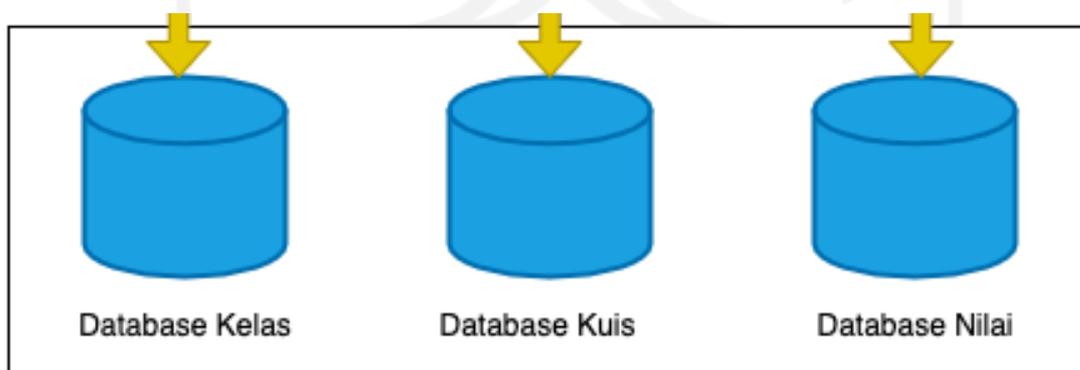
Service provider atau penyedia *service* adalah sebagai penyedia *service* yang akan dibutuhkan pada tingkat produksi. Pada Gambar 3.4 terdapat tiga *service* yang tersedia yaitu mengelola kelas, mengelola kuis, dan mengelola nilai. Tiga *service* yang telah tersedia nantinya akan digunakan oleh *service 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.



Gambar 3.4 Service provider

C. Service Registry

Service Registry merupakan tempat untuk menyimpan data *service-service* yang diperlukan. Data *service* disimpan seperti pada Gambar 3.5.



Gambar 3.5 Service registry

D. Message Broker (ESB)

Message Broker (ESB) merupakan teknologi yang memiliki peran untuk menyampaikan pesan atau *service*. *Message broker* memiliki bermacam teknologi, dalam perancangan

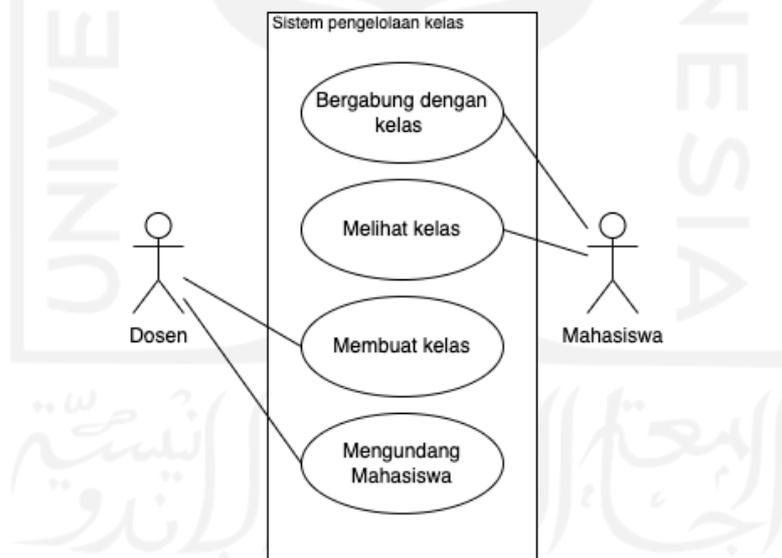
arsitektur ini menggunakan ESB (*enterprise service bus*). Selain itu untuk menyampaikan komunikasi *service* kepada user menggunakan *simple object acces protocol* (SOAP) antara ESB dan user. Untuk lebih jelas dapat dilihat pada Gambar 3.6



Gambar 3.6 Message broker (ESB)

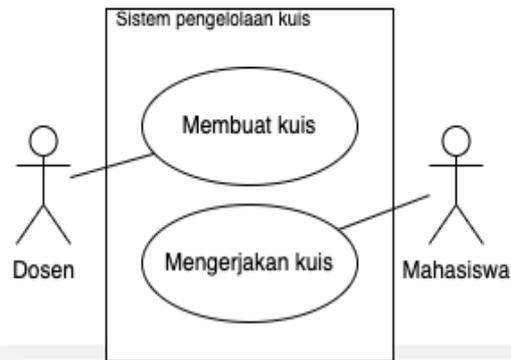
3.2.5 Rancangan Use Case Diagram

Use case diagram adalah gambaran urutan interaksi antara satu atau lebih aktor dengan sistem. *Use case* menggambarkan proses sistem dari sudut pandang pengguna. *Use case* untuk sistem utama manajemen kuis ini memiliki tiga diagram sesuai rancangan identifikasi sistem sebelumnya yang juga menghasilkan tiga sistem. Gambar 3.7 merupakan *use case diagram* pada sistem pengelolaan kelas. sistem pengelolaan kuis terdiri empat *use case* dan dua aktor. Aktor mahasiswa mempunyai *use case* bergabung dengan kelas dan melihat kelas, aktor dosen memiliki *use case* membuat kelas dan mengundang mahasiswa.



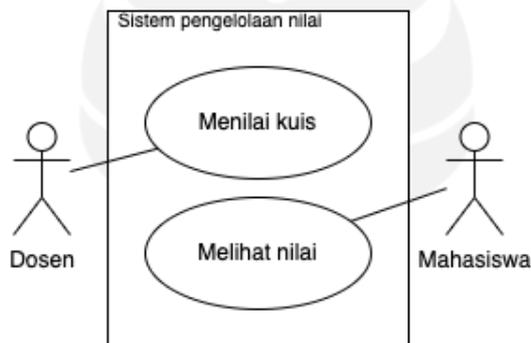
Gambar 3.7 Use case diagram sistem pengelolaan kelas

Use case diagram selanjutnya adalah *use case* dari sistem pengelolaan kuis. Sistem pengelolaan kuis terdiri dari dua *use case* dan dua aktor. Aktor mahasiswa memiliki *use case* mengerjakan kuis dan aktor dosen memiliki *use case* membuat kuis. Untuk lebih jelas dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Use case diagram* sistem pengelolaan kuis

Use case diagram yang terakhir adalah *use case* dari sistem pengelolaan nilai. Sistem pengelolaan nilai terdiri dari dua *use case* dan dua aktor. Aktor mahasiswa memiliki *use case* melihat nilai kuis dan aktor dosen memiliki *use case* menilai kuis. Untuk lebih jelas dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Use case diagram* sistem pengelolaan nilai

Untuk lebih jelas mengenai keterangan *use case diagram* sistem manajemen kuis, dapat dilihat pada Tabel 3.9

Tabel 3.9 Keterangan *use case diagram*

No	Sistem	Use case	Aktor	Keterangan
S1	Sistem pengelolaan kelas	UC1. Bergabung dengan kelas	Mahasiswa	Proses untuk mahasiswa melihat kelas yang diikuti
		UC2. Melihat kelas	Mahasiswa	Proses untuk mahasiswa melihat kelas yang diikuti
		UC3. Mengundang Mahasiswa	Dosen	Proses untuk dosen mengundang mahasiswa ke kelas
		UC4. Membuat kelas	Dosen	Proses untuk dosen membuat kelas

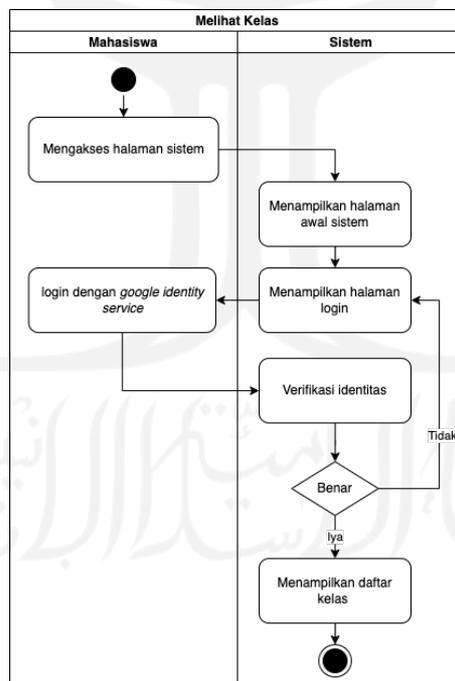
S2	Sistem pengelolaan kuis	UC5. Mengerjakan kuis	Mahasiswa	Proses untuk mahasiswa mengerjakan kuis yang tersedia
		UC6. Membuat kuis	Dosen	Proses untuk dosen membuat kuis bagi mahasiswa
S3	Sistem pengelolaan nilai	UC7. Menilai kuis	Dosen	Proses untuk dosen menilai kuis setiap mahasiswa
		UC8. Melihat nilai	Mahasiswa	Proses untuk mahasiswa melihat nilai kuis yang telah dikerjakan

3.2.6 Rancangan Activity 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.

1. Activity Diagram Mahasiswa Melihat Kelas

Activity diagram dibawah ini menggambarkan alur ketika mahasiswa ingin melihat kelas yang sedang ditempuh. Struktur lengkap diagram dapat dilihat pada Gambar 3.10

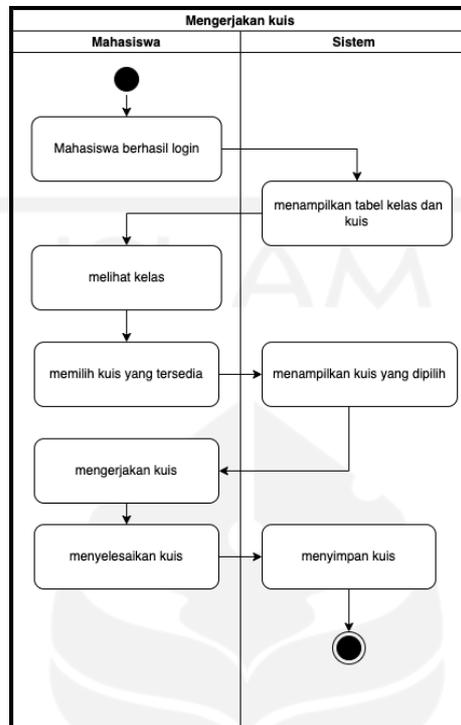


Gambar 3.10 Activity diagram mahasiswa melihat kelas

2. Activity Diagram Mahasiswa Mengerjakan Kuis

Activity diagram dibawah ini menggambarkan alur mengerjakan kuis untuk mahasiswa. Mahasiswa mengakses halaman kelas yang selanjutnya mahasiswa dapat melihat jadwal kuis

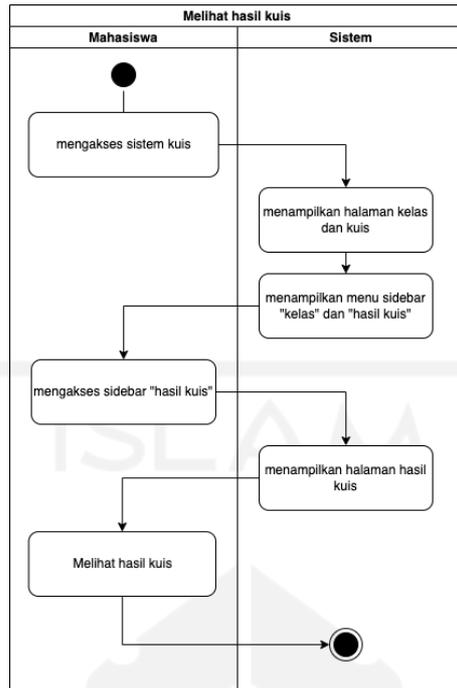
dari setiap kelas yang diikutinya. Setelah mahasiswa melihat jadwal, mahasiswa dapat langsung mengakses kuis yang sudah tersedia pada jadwal kuis, selanjutnya mahasiswa dapat mengerjakan kuis. Struktur lengkap diagram dapat dilihat pada Gambar 3.11.



Gambar 3.11 Activity diagram mahasiswa mengerjakan kuis

3. Activity Diagram Mahasiswa Melihat Hasil Kuis

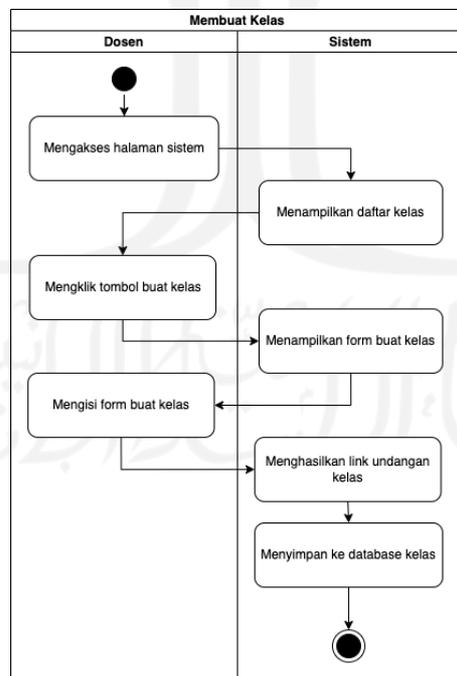
Activity diagram dibawah ini menggambarkan alur melihat hasil kuis untuk mahasiswa. Mahasiswa melakukan akses ke halaman kelas selanjutnya mahasiswa dapat mengakses halaman lihat hasil kuis. Pada halaman lihat hasil kuis mahasiswa dapat melihat nilai kuis dari setiap kuis yang telah diikuti. Struktur lengkap diagram dapat dilihat pada Gambar 3.11



Gambar 3.12 Activity diagram mahasiswa melihat hasil kuis

4. Activity Diagram Dosen Membuat Kelas

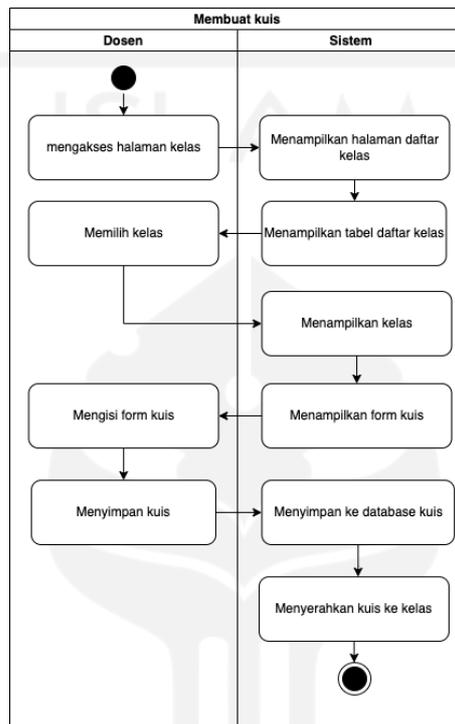
Activity diagram dibawah ini menggambar alur dosen untuk membuat kelas pada sistem manajemen kuis. untuk lebih jelas dapat dilihat pada gambar 3.13.



Gambar 3.13 Activity diagram dosen membuat kelas

5. Activity Diagram Dosen Membuat Kuis

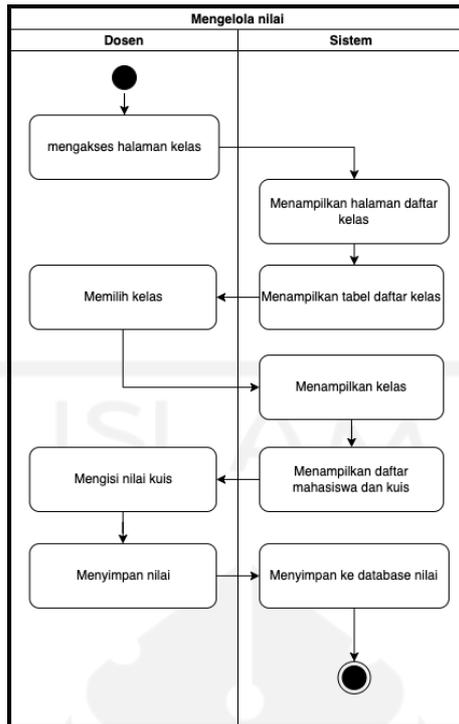
Activity diagram dibawah ini menggambarkan alur membuat kuis untuk dosen. Dosen melakukan akses ke halaman kelas dan memilih kelas mana yang akan melakukan kuis, selanjutnya dosen mengakses halaman buat kuis, setelah selesai dosen dapat menyimpan kuis tersebut untuk ditampilkan di kelas. Struktur lengkap diagram dapat dilihat pada Gambar 3.14.



Gambar 3.14 Activity diagram dosen membuat kuis

6. Activity Diagram Dosen Mengelola Nilai

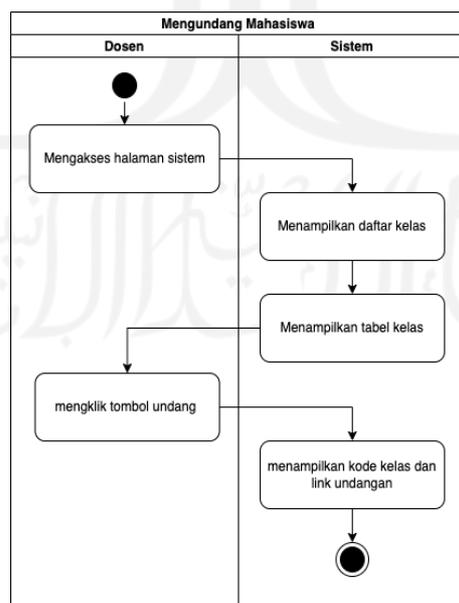
Activity diagram dibawah ini menggambarkan alur mengelola nilai kuis untuk dosen. Dosen melakukan akses ke halaman kelas dan memilih kelas mana yang akan dikelola nilainya, selanjutnya dosen dapat mengakses data setiap mahasiswa, selanjutnya dosen dapat memasukkan nilai kuis ke masing-masing mahasiswa, dan tahap terakhir adalah menyerahkannya nilai kuis kepada masing-masing mahasiswa. Struktur lengkap diagram dapat dilihat pada Gambar 3.15.



Gambar 3.15 Activity diagram dosen mengelola nilai

7. Activity Diagram Dosen Mengundang Mahasiswa

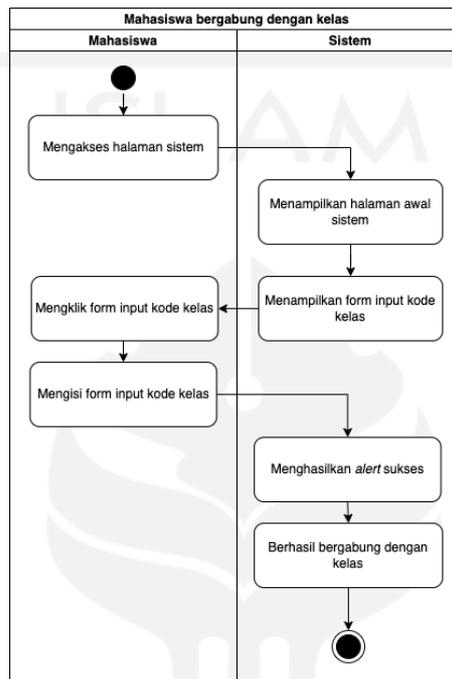
Activity diagram pada Gambar 3.16 menggambarkan alur ketika dosen akan mengundang mahasiswa untuk masuk ke kelas. Dosen mengakses halaman sistem selanjutnya dosen memilih kelas yang akan diundang dan sistem akan menampilkan kode kelas dan link undangan kelas untuk dibagikan. Untuk lebih jelas dapat dilihat pada Gambar 3.16.



Gambar 3.16 Activity diagram dosen mengundang mahasiswa

8. Activity Diagram Mahasiswa Bergabung Dengan Kelas

Activity diagram pada Gambar 3.16 menggambarkan alur ketika mahasiswa bergabung dengan kelas. Mahasiswa yang telah mendapatkan kode kelas atau link undangan kelas dapat menginputkan melalui form input kode, setelah berhasil mahasiswa sudah dapat bergabung dengan kelas. Untuk lebih jelas dapat dilihat pada Gambar 3.16



Gambar 3.17 Activity diagram mahasiswa bergabung dengan kelas

3.2.7 Perancangan Detail Layanan

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 dapat dilihat pada Tabel 3.10

Tabel 3.10 Perancangan detail layanan sistem pengelolaan kelas

<i>Use case</i>	Service	Front-End	BackEnd	Method
UC1 Membuat kelas	Kelas	[FE1.1] Halaman Daftar Kelas Dosen [FE1.2] Halaman modal form buat kelas dan link undangan kelas	Controller Kelas	Method POST Add
UC2 Mengundang mahasiswa		[FE2.1] Menampilkan daftar kelas [FE2.2] Halaman kode kelas dan link undangan		Method POST undang mahasiswa

UC3 Bergabung dengan kelas		[FE3.1]Menampilkan halaman awal sistem dan form input kode kelas [FE3.2]menghasilkan alert sukses		Method GET
UC4 Melihat kelas		[FE4.1]Menampilkan halaman login [FE4.2]Menampilkan daftar kelas		Method GET

Perancangan layanan sistem pengelolaan kuis dapat dilihat pada Tabel 3.11.

Tabel 3.11 Perancangan detail layanan sistem pengelolaan kuis

Use case	Service	Front-End	BackEnd	Method
UC5 Membuat kuis	Kuis	[FE5.1]Menampilkan halaman daftar kelas dosen [FE5.2]Menampilkan form kuis [FE5.3]Menyimpan kuis [FE5.4]menyerahkan kuis ke kelas	Controller kuis	Method GET kuis
UC6 Mengerjakan kuis		[FE6.1]Menampilkan tabel kelas dan kuis mahasiswa [FE6.2]Menampilkan kuis yang dipilih [FE6.3]Menyimpan kuis		Method GET kuis

Perancangan layanan sistem pengelolaan nilai dapat dilihat pada Tabel 3.12.

Tabel 3.12 Perancangan detail layanan sistem pengelolaan nilai.

Use case	Service	Front-End	BackEnd	Method
UC7 Menilai kuis	Nilai	[FE7.1]Menampilkan halaman daftar kelas dan sidebar kelola nilai [FE7.2]Menampilkan daftar mahasiswa dan kuis [FE7.3]Menyimpan ke database kuis	Controller nilai	Method GET nilai
UC8 Melihat nilai		[FE8.1]Menampilkan halaman kelas dan kuis dan menampilkan menu <i>sidebar</i> hasil kuis [FE8.2]Menampilkan halaman hasil kuis		Method POST nilai

3.2.8 Perancangan Data

Terdapat tiga data utama yang diperlukan dalam perancangan sistem manajemen kuis, yang pertama adalah data kelas, data kelas memiliki peran penting dalam penyimpanan data kelas yang akan digunakan pada *use case* yang memiliki *service* yang sama seperti membuat kelas dan bergabung dengan kelas. Kemudian data kuis, data kuis memiliki peran utama dalam penyimpanan data pembuatan kuis dan pengerjaan kuis. Terakhir data nilai, data nilai digunakan untuk menyimpan data nilai hasil kuis dan juga sebagai data untuk mengelola nilai.

Tiga data utama di atas akan didukung oleh berbagai data pendukung seperti data dosen dan data mahasiswa yang berguna untuk menyimpan data diri masing-masing aktor. Kemudian data kelasMahasiswa sebagai penyimpanan data kelas yang diambil oleh mahasiswa. Terakhir adalah data pertanyaan, data pertanyaan mempunyai fungsi untuk menyimpan data pertanyaan yang telah dibuat oleh dosen.

1. Data dosen

Data dosen merupakan data yang digunakan untuk menyimpan data dosen pada sistem manajemen kuis. data dosen yang diperlukan adalah NIK, Nama, Email

2. Data mahasiswa

Data Mahasiswa merupakan data yang digunakan untuk menyimpan data mahasiswa pada sistem manajemen kuis. Data mahasiswa yang diperlukan adalah NIM, Nama, Email

3. Data kelas

Data kelas merupakan data yang digunakan untuk menyimpan data kelas pada sistem manajemen kuis. Data kelas yang diperlukan adalah KodeMataKuliah, NamaMataKuliah, Dosen

4. Data KelasMahasiswa

Data Kelas-Mahasiswa merupakan data yang digunakan untuk menyimpan Kelas-Mahasiswa pada sistem manajemen kuis. Data Kelas-Mahasiswa yang diperlukan adalah NIM, KodeMataKuliah.

5. Data kuis

Data Kuis merupakan data yang digunakan untuk menyimpan data Kuis setelah dosen membuat kuis pada sistem manajemen kuis. Data Kuis yang diperlukan adalah KodeKuis, Nama, KodeMataKuliah.

6. Data pertanyaan

Data Pertanyaan merupakan data yang digunakan untuk menyimpan data pertanyaan kuis yang dibuat dosen pada sistem manajemen kuis. data Pertanyaan yang diperlukan adalah

KodePertanyaan, Pertanyaan, Jawaban1, Jawaban2, Jawaban3, Jawaban4, Jawaban5, JawabanBenar, KodeKuis.

7. Data nilai

Data Nilai merupakan data yang digunakan untuk menyimpan data Nilai hasil Kuis pada sistem manajemen kuis. Data Kuis yang diperlukan adalah NIM, KodeKuis, Nilai

3.2.9 Tabel Data Sistem Manajemen Kuis

Sistem manajemen kuis memiliki tiga sistem yaitu, sistem pengelolaan kelas, sistem pengelolaan kuis dan sistem pengelolaan nilai. Masing-masing sistem akan memiliki database yang terpisah. Oleh karena itu terdapat tiga database yang masing-masing memiliki beberapa tabel sebagai berikut. Tabel ini dirancang dari perancangan data sebelumnya.

1. Database Sistem Pengelolaan Kelas

Database sistem pengelolaan kelas terdiri dari tabel kelas, tabel mahasiswa, tabel dosen, dan tabel kelasMahasiswa. Database ini berfungsi untuk membuat kelas dan melihat kelas.

a. Tabel Kelas

Tabel kelas digunakan untuk menyimpan data kelas yang telah dibuat oleh dosen. Tabel kelas berisi KodeMataKuliah, Nama, dan dosen. Tabel kelas dapat dilihat pada Tabel 3.13.

Tabel 3.13 Struktur tabel kelas

field	type	Keterangan
KodeMataKuliah	int(4)	Untuk mengisi kode
NamaMataKuliah	varchar(100)	Untuk mengisi nama kelas
Dosen	varchar(100)	Untuk mengisi nama dosen

b. Tabel Mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa. Tabel mahasiswa berisi NIM, Email, dan password. Tabel mahasiswa dapat dilihat pada Tabel 3.14.

Tabel 3.14 Struktur tabel mahasiswa

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM mahasiswa
Email	varchar(320)	Untuk mengisi email ke sistem

password	char(60)	Untuk mengisi password ke sistem
----------	----------	----------------------------------

c. Tabel Dosen

Tabel dosen digunakan untuk menyimpan data dosen. Tabel dosen berisi data NIK, Nama, dan Email. Untuk lebih jelas dapat dilihat pada Tabel 3.15.

Tabel 3.15 Struktur tabel dosen

field	type	Keterangan
NIK	int(16)	Untuk mengisi NIK dosen
Nama	varchar(100)	Untuk mengisi Nama dosen
Email	varchar(320)	Untuk mengisi Email dosen ke sistem

d. Tabel KelasMahasiswa

Tabel KelasMahasiswa digunakan untuk menyimpan data kelas yang telah diambil oleh mahasiswa. Tabel KelasMahasiswa berisi data NIM, dan KodeMataKuliah. Untuk lebih jelas dapat dilihat pada Tabel 3.16.

Tabel 3.16 Struktur tabel KelasMahasiswa

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM mahasiswa
KodeMataKuliah	int(4)	Untuk mengisi kode kelas sistem

2. Database Pengelolaan Kuis

Database pengelolaan kuis terdiri dari tabel kelas, tabel mahasiswa, tabel dosen, tabel kuis, tabel kelasMahasiswa, tabel pertanyaan. Database ini berfungsi untuk membuat dan mengerjakan kuis.

a. Tabel kelas

Tabel kelas digunakan untuk menyimpan data kelas yang telah dibuat oleh dosen. Tabel kelas berisi KodeMataKuliah, Nama, dan dosen. Tabel kelas dapat dilihat pada Tabel 3.17.

Tabel 3.17 Struktur tabel kelas

field	type	Keterangan
KodeMataKuliah	int(4)	Untuk mengisi kode
NamaMataKuliah	varchar(100)	Untuk mengisi nama kelas

Dosen	varchar(100)	Untuk mengisi nama dosen
-------	--------------	--------------------------

b. Tabel Mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa. Tabel mahasiswa berisi NIM, Email, dan password. Tabel mahasiswa dapat dilihat pada Tabel 3.18.

Tabel 3.18 Struktur tabel mahasiswa

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM mahasiswa
Email	varchar(320)	Untuk mengisi email ke sistem
password	char(60)	Untuk mengisi password ke sistem

c. Tabel Dosen

Tabel dosen digunakan untuk menyimpan data dosen. Tabel dosen berisi data NIK, Nama, dan Email. Untuk lebih jelas dapat dilihat pada Tabel 3.19.

Tabel 3.19 Struktur tabel dosen

field	type	Keterangan
NIK	int(16)	Untuk mengisi NIK dosen
Nama	varchar(100)	Untuk mengisi Nama dosen
Email	varchar(320)	Untuk mengisi Email dosen ke sistem

d. Tabel Kuis

Tabel Kuis digunakan untuk menyimpan data kuis yang akan digunakan oleh mahasiswa. Tabel kuis berisi data KodeKuis, Nama, dan KodeMataKuliah. Untuk lebih jelas dapat dilihat pada Tabel 3.20.

Tabel 3.20 Struktur tabel kuis

field	type	Keterangan
KodeKuis	int(5)	Untuk mengisi KodeKuis
Nama	varchar(100)	Untuk mengisi nama kuis
KodeMataKuliah	int(4)	Untuk mengisi KodeMataKuliah

e. Tabel KelasMahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa. Tabel mahasiswa berisi NIM, Email, dan password. Tabel mahasiswa dapat dilihat pada Tabel 3.21.

Tabel 3.21 Struktur tabel KelasMahasiswa

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM
KodeMataKuliah	int(4)	Untuk mengisi KodeMataKuliah

f. Tabel Pertanyaan

Tabel Pertanyaan digunakan untuk menyimpan pertanyaan kuis yang dibuat oleh dosen. Tabel pertanyaan berisi data KodePertanyaan, Pertanyaan, Jawaban1, Jawaban2, Jawaban3, Jawaban4, Jawaban5, JawabanBenar, KodeKuis. Untuk lebih jelas dapat dilihat pada tabel Tabel 3.22.

Tabel 3.22 Struktur tabel pertanyaan

field	type	Keterangan
KodePertanyaan	int(4)	Untuk mengisi KodePertanyaan
Pertanyaan	text	Untuk mengisi Pertanyaan
Jawaban1	text	Untuk mengisi Jawaban1
Jawaban2	text	Untuk mengisi Jawaban2
Jawaban3	text	Untuk mengisi Jawaban3
Jawaban4	text	Untuk mengisi Jawaban4
Jawaban5	text	Untuk mengisi Jawaban5
JawabanBenar	char(255)	Untuk mengisi JawabanBenar
KodeKuis	int(5)	Untuk mengisi KodeKuis

3. Database Sistem Pengelolaan Nilai

Database pengolahan nilai terdiri dari tabel kelas, tabel mahasiswa, tabel dosen, tabel kuis, tabel kelasMahasiswa dan tabel nilai. Database ini berfungsi untuk menilai dan melihat hasil kuis.

a. Tabel kelas

Tabel kelas digunakan untuk menyimpan data kelas yang telah dibuat oleh dosen. Tabel kelas berisi KodeMataKuliah, Nama, dan dosen. Tabel kelas dapat dilihat pada Tabel 3.23.

Tabel 3.23 Struktur tabel kelas

field	type	Keterangan
KodeMataKuliah	int(4)	Untuk mengisi KodeMataKuliah
NamaMataKuliah	varchar(100)	Untuk mengisi nama kelas
Dosen	varchar(100)	Untuk mengisi nama dosen

b. Tabel Mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa. Tabel mahasiswa berisi NIM, Email, dan password. Tabel mahasiswa dapat dilihat pada Tabel 3.24.

Tabel 3.24 Struktur tabel mahasiswa

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM mahasiswa
Email	varchar(320)	Untuk mengisi email ke sistem
password	char(60)	Untuk mengisi password ke sistem

c. Tabel Dosen

Tabel dosen digunakan untuk menyimpan data dosen. Tabel dosen berisi data NIK, Nama, dan Email. Untuk lebih jelas dapat dilihat pada Tabel 3.25.

Tabel 3.25 Struktur tabel dosen

field	type	Keterangan
NIK	int(16)	Untuk mengisi NIK dosen
Nama	varchar(100)	Untuk mengisi Nama dosen
Email	varchar(320)	Untuk mengisi Email dosen ke sistem

d. Tabel Kuis

Tabel Kuis digunakan untuk menyimpan data kuis yang akan digunakan oleh mahasiswa. Tabel kuis berisi data KodeKuis, Nama, dan KodeMataKuliah. Untuk lebih jelas dapat dilihat pada Tabel 3.26.

Tabel 3.26 Struktur tabel kuis

field	type	Keterangan
KodeKuis	int(5)	Untuk mengisi KodeKuis
Nama	varchar(100)	Untuk mengisi Nama kuis
KodeMataKuliah	int(4)	Untuk mengisi KodeMataKuliah

e. Tabel KelasMahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa. Tabel mahasiswa berisi NIM, Email, dan password. Tabel mahasiswa dapat dilihat pada Tabel 3.27.

Tabel 3.27 Struktur tabel KelasMahasiswa

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM
KodeMataKuliah	int(4)	Untuk mengisi KodeMataKuliah

f. Tabel nilai

Tabel nilai digunakan untuk menyimpan nilai yang telah diberikan oleh dosen kepada mahasiswa. Tabel nilai berisi data NIM, KodeKuis, dan Nilai. Untuk lebih jelas dapat dilihat pada Tabel 3.28.

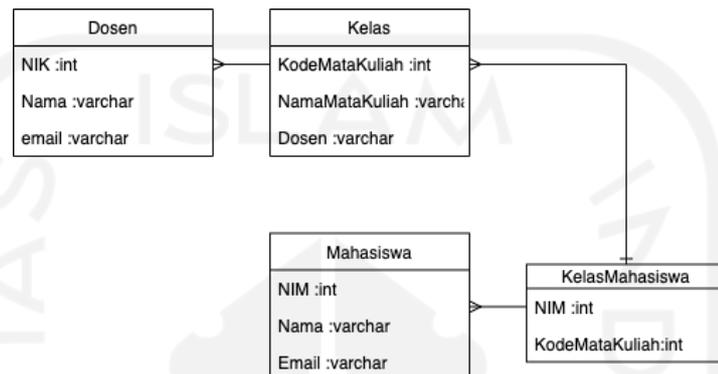
Tabel 3.28 Struktur tabel nilai

field	type	Keterangan
NIM	int(8)	Untuk mengisi NIM
KodeKuis	int(5)	Untuk mengisi KodeKuis
Nilai	int(3)	Untuk mengisi Nilai

3.2.10 Diagram Relasi Basis Data Sistem Manajemen Kuis

1. Diagram Relasi Basis Data Pengelolaan Kelas

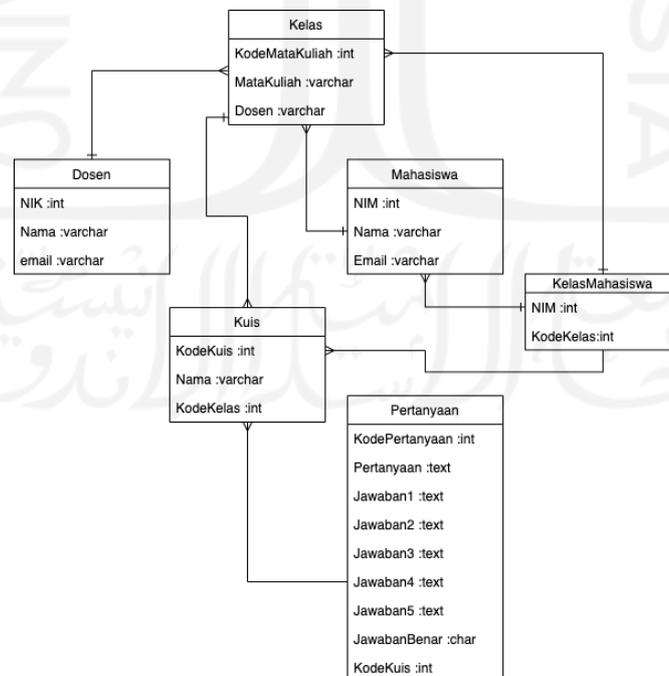
Gambar 3.18 menggambarkan relasi relasi basis data dari sistem pengelolaan kuis. Diagram relasi basis data pengelolaan kuis terdiri dari tabel dosen, tabel kelas, tabel mahasiswa, dan tabel kelasMahasiswa.



Gambar 3.18 Diagram relasi basis data sistem manajemen kuis

2. Diagram Relasi Basis Data Pengelolaan Kuis

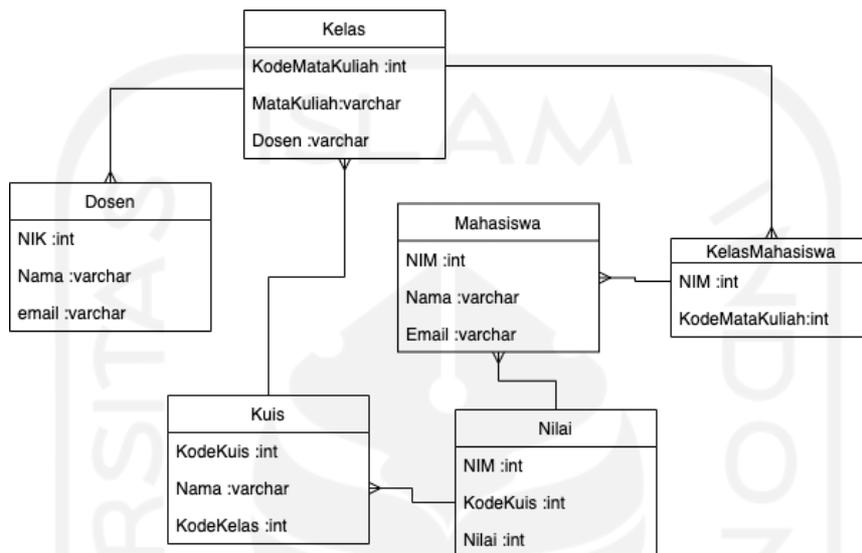
Gambar 3.19 menggambarkan relasi basis data dari sistem pengelolaan kuis. Diagram relasi basis data pengelolaan kuis terdiri dari tabel dosen, tabel kelas, tabel mahasiswa, tabel kelasMahasiswa, tabel kuis, dan tabel pertanyaan



Gambar 3.19 Diagram relasi basis data sistem pengelolaan kuis

3. Diagram Relasi Basis Data Pengelolaan Nilai

Gambar 3.20 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 3.20 Diagram relasi basis data sistem pengelolaan nilai

3.2.1 Perancangan Antarmuka BackEnd

Perancangan antarmuka backend sistem manajemen kuis akan berbasis application programming interface atau API. API akan memungkinkan antarmuka BackEnd digunakan lebih luas karena pembuatan logic dilakukan secara terpisah dengan FrontEnd. Untuk lebih jelas tentang perancangan antarmuka BackEnd dapat dilihat pada Tabel 3.29.

Tabel 3.29 Perancangan Antarmuka BackEnd Sistem Manajemen Kuis

Perancangan Antarmuka BackEnd Pengelolaan Kelas				
No	Use case/Halaman	API		Parameter
		Method	Path	
1	Membuat kelas	POST	/api/tambahkelas	/service/myresource/kelas/{tambahkelasId}
2	Mengundang mahasiswa	GET	/api/undang mahasiswa	/service/myresource/kelas/{undangmahasiswaId}

3	Bergabung dengan kelas	POST	/api/gabungkelas	/service/myresource/kelas/{gabungkelasId}
4	Melihat kelas	POST	/api/lihatkelas/	/service/myresource/kelas/{lihatkelasId}
Perancangan Antarmuka BackEnd Pengelolaan kuis				
No	Use case/Halaman	API		Parameter
		Method	Path	
1	Membuat kuis	POST	/api/buatkuis	/service/myresource/kuis/{buatkuisId}
2	Mengerjakan kuis	POST	/api/mengerjakankuis	/service/myresource/kuis/{mengerjakankuisId}
Perancangan Antarmuka BackEnd Pengelolaan Nilai				
No	Use case/Halaman	API		Parameter
		Method	Path	
1	Mengelola nilai	POST	/api/kelolanilai	/service/myresource/nilai/{kelolanilaiId}
2	Melihat nilai	GET	/api/lihatnilai	/service/myresource/nilai/{lihatnilaiId}

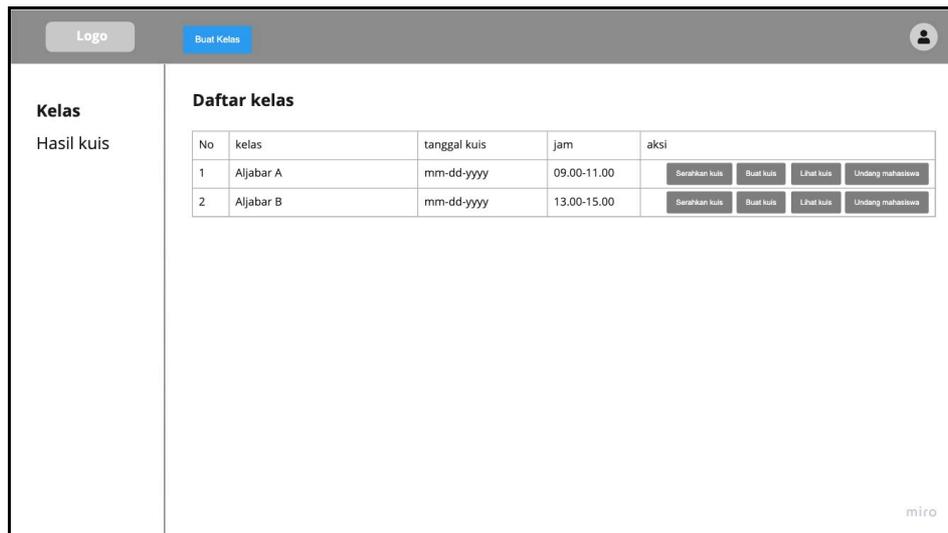
3.2.2 Perancangan Antarmuka Front-End

Seluruh proses rancangan antar muka dibuat menggunakan aplikasi miro. Aplikasi miro digunakan karena memudahkan dalam perancangan perancangan struktur, *user flow* dan navigasi. Rancangan antarmuka sistem manajemen dibuat berdasarkan kebutuhan fungsional dan non fungsional yang telah dianalisis Tabel 3.10.

1. FE 1.1 Halaman Daftar Kelas Dosen

Berdasarkan pada Tabel 3.10 Halaman ini adalah halaman ketika aktor dosen ingin membuat kelas. Halaman dosen buat kelas berisi form nama kelas, undangan kelas dengan email dan link undangan. Langkah pertama untuk membuat kelas dengan mengakses halaman daftar kelas dan klik tombol tambah kelas. Halaman ini dapat lebih jelas dilihat pada

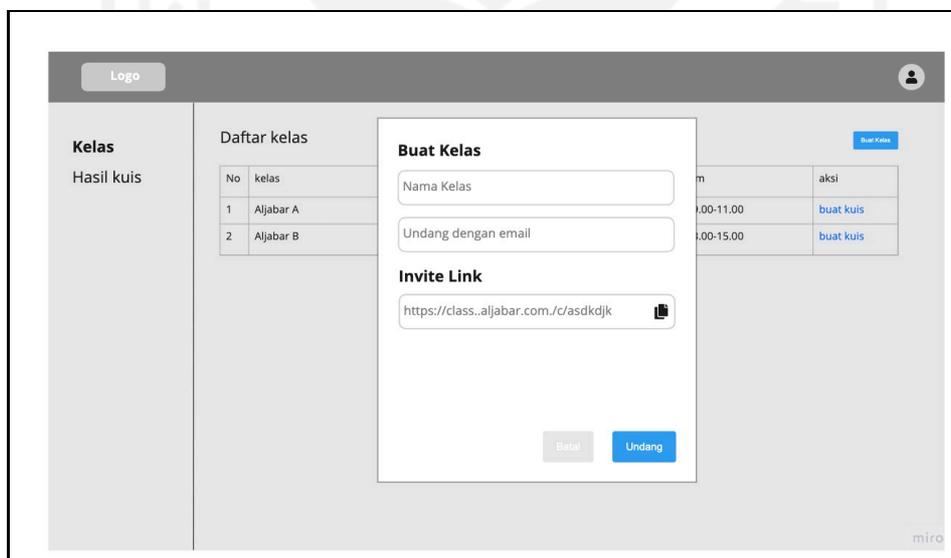
Gambar 3.21.



Gambar 3.21 Halaman daftar kelas dosen

2. FE 1.2 Halaman Modal Form Buat Kelas Dan Link Undangan Kelas

Setelah klik tombol tambah kelas, akan muncul modal yang berisi form nama kelas, undangan dengan email dan link undangan kelas. Halaman ini dapat lebih jelas dilihat pada Gambar 3.21.



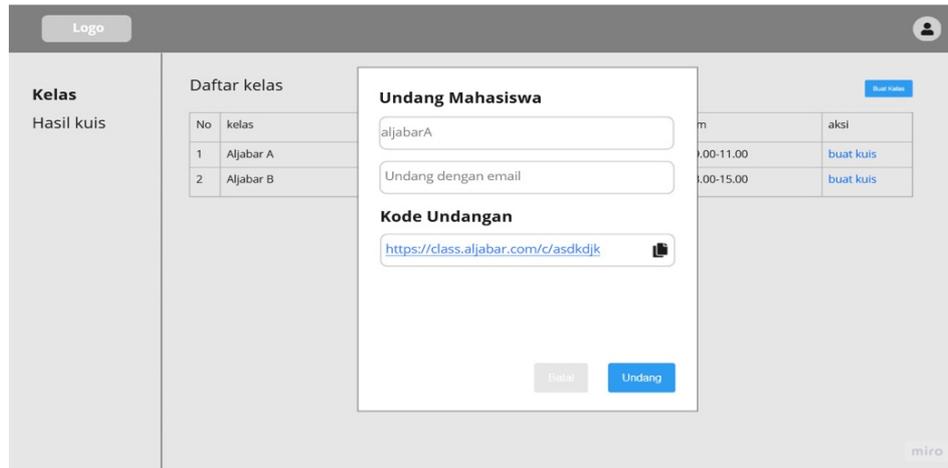
Gambar 3.22 Halaman modal form buat kelas dan link undangan kelas

3. FE 2.1 Menampilkan Daftar Kelas

Halaman ini merupakan halaman ketika dosen ingin mengundang mahasiswa masuk ke kelas. Pertama akan ditampilkan daftar kelas yang telah ada, kemudian dosen dapat memilih kelas yang akan membuka link undangan bagi mahasiswa. Halaman ini dapat lebih jelas dilihat pada Gambar 3.21.

4. FE 2.2 Halaman Undang Mahasiswa

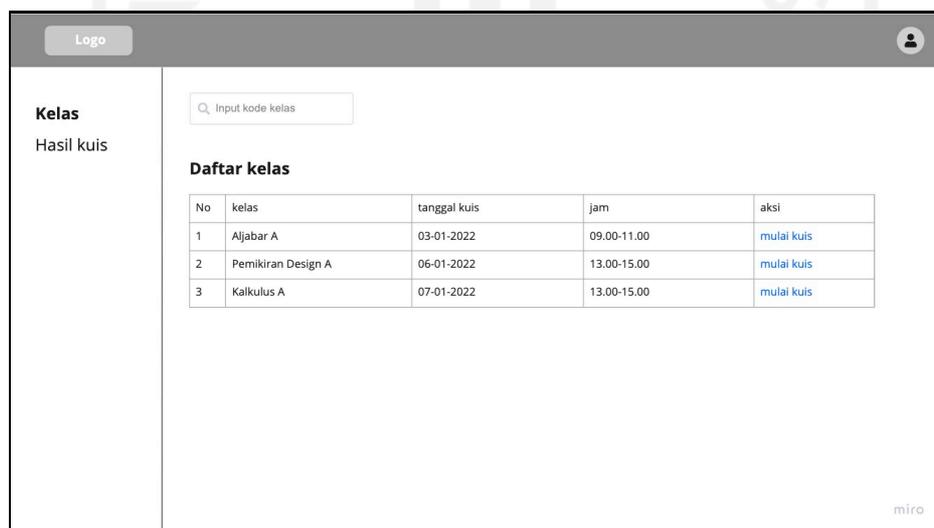
Kemudian setelah dosen memilih kelas akan muncul modal undangan kelas. Halaman dosen mengundang mahasiswa akan menampilkan modal nama kelas dan kode undangan kelas yang dapat di copy. Halaman ini dapat lebih jelas dilihat pada Gambar 3.23.



Gambar 3.23 Halaman undang mahasiswa

5. FE 3.1 Menampilkan Daftar Kelas Mahasiswa

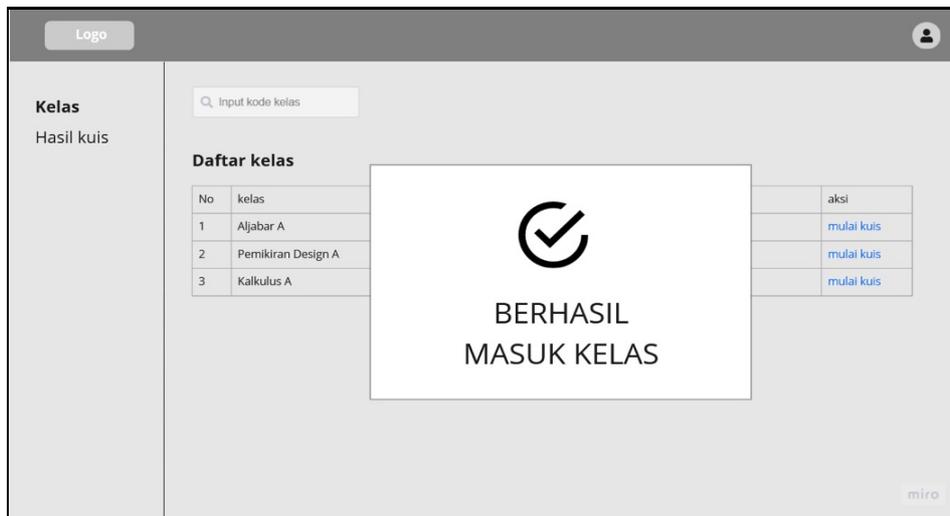
Halaman ini adalah halaman awal. Ketika aktor mahasiswa berhasil melakukan login. Halaman ini akan menampilkan daftar kelas, jadwal kuis untuk mahasiswa dan form input kode kelas. Halaman ini dapat lebih jelas dilihat pada Gambar 3.24.



Gambar 3.24 Halaman daftar kelas mahasiswa

6. FE 3.2 Menghasilkan Alert Sukses

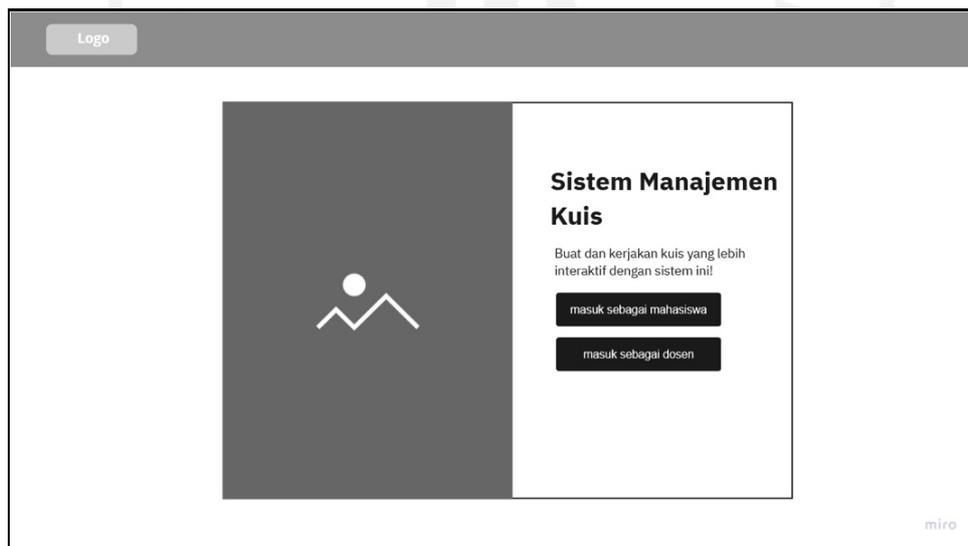
Halaman mahasiswa bergabung dengan kelas merupakan halaman ketika mahasiswa berhasil masuk ke kelas dengan menggunakan metode kelas. Sistem akan menampilkan notifikasi atau alert ketika mahasiswa berhasil masuk ke kelas. Halaman ini dapat lebih jelas dilihat pada Gambar 3.25



Gambar 3.25 Notifikasi sukses masuk kelas

7. FE 4.1 Perancangan Halaman Login Mahasiswa dan Dosen

Halaman ini adalah halaman awal pada saat sistem dijalankan, dimana pemilik pengguna diminta untuk memasukkan username dan password agar dapat melakukan akses ke sistem. Halaman ini dapat lebih jelas dilihat pada Gambar 3.26



Gambar 3.26 Halaman login mahasiswa dan dosen

8. FE 4.2 Perancangan Halaman Daftar Kelas dan Jadwal Kuis

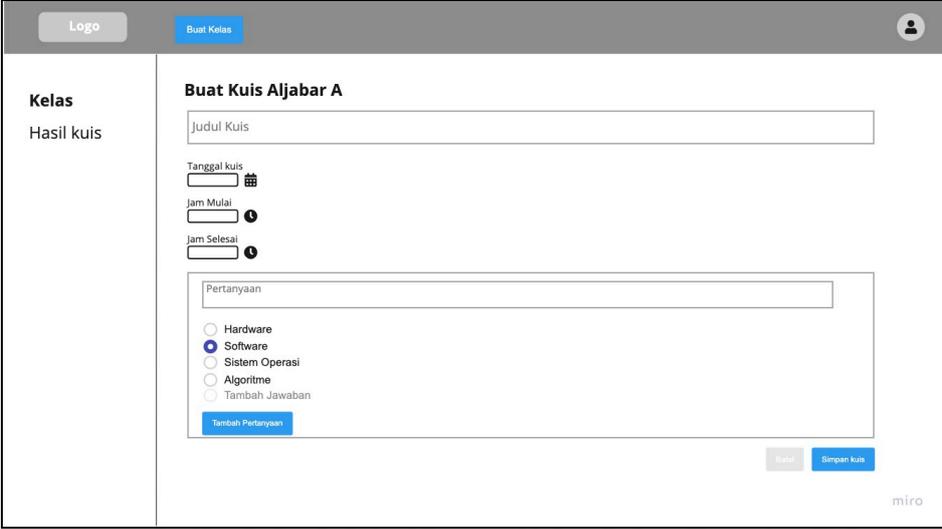
Halaman ini adalah halaman awal Ketika aktor mahasiswa berhasil melakukan login. Halaman ini akan menampilkan daftar kelas dan jadwal kuis untuk mahasiswa. Halaman ini dapat lebih jelas dilihat pada Gambar 3.24.

9. FE 5.1 Menampilkan Halaman Daftar Kelas Dosen

Halaman ini merupakan halaman ketika dosen ingin membuat kuis. Pertama akan ditampilkan daftar kelas yang telah ada, kemudian dosen dapat memilih kelas yang akan dibuat kuis. Kemudian dosen dapat mengklik tombol buat kuis. Halaman ini dapat lebih jelas dilihat pada Gambar 3.21.

10. FE 5.2 Menampilkan Form Kuis

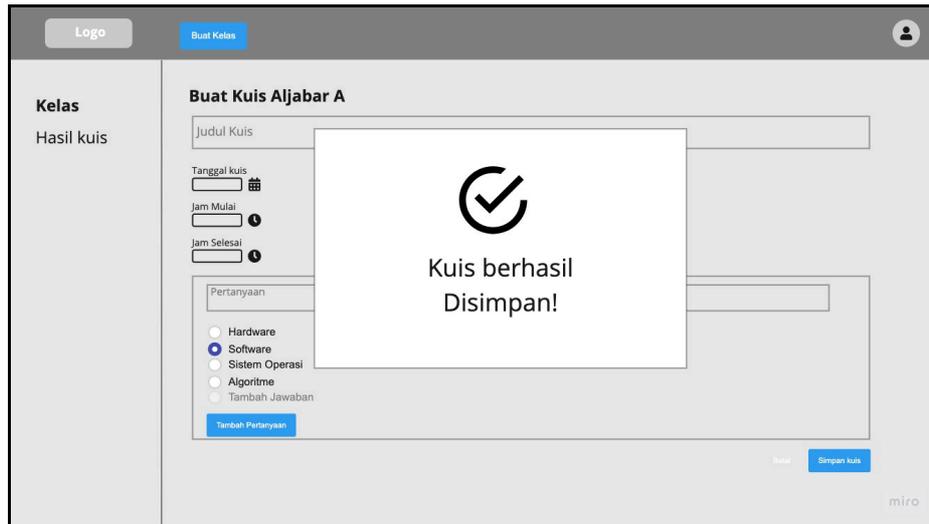
Halaman ini adalah halaman pada saat aktor dosen ingin membuat kuis pada suatu kelas, dimana dosen dapat langsung membuat kuis pada halaman tersebut. Dosen dapat mengisi form kuis yang telah tersedia. Halaman ini dapat lebih jelas dilihat pada Gambar 3.27.



Gambar 3.27 Halaman menampilkan form kuis

11. FE 5.3 Menyimpan Form Kuis

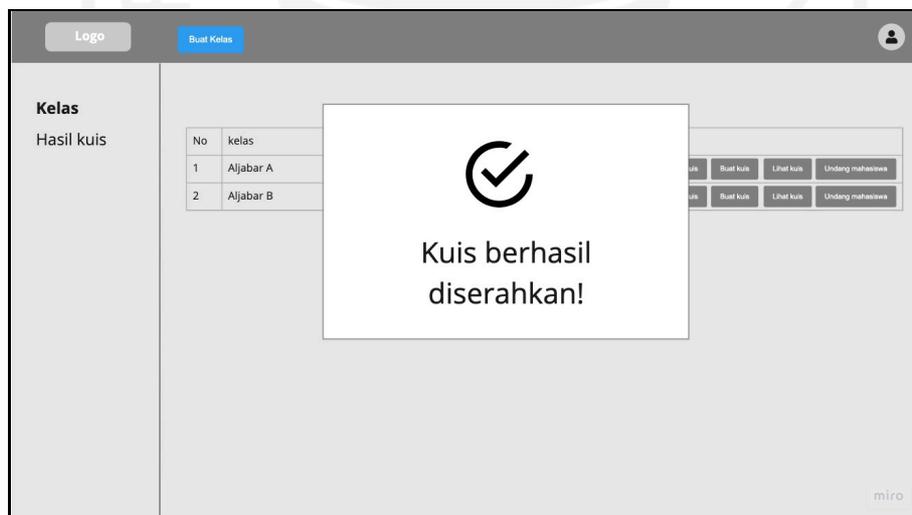
Halaman ini adalah halaman pada saat aktor dosen telah mengisi semua form pembuatan kuis. Setelah semua form telah terisi, dosen dapat menyimpan kuis dan mengklik tombol simpan kuis. Halaman ini dapat lebih jelas dilihat pada Gambar 3.28



Gambar 3.28 Halaman notifikasi berhasil menyimpan form kuis

12. FE 5.4 Menyerahkan Kuis Ke Kelas

Halaman ini adalah halaman pada saat aktor dosen telah menyimpan kuis. Setelah kuis tersimpan dosen dapat menyerahkan ke kelas dengan mengklik tombol serahkan kuis. Halaman ini dapat lebih jelas dilihat pada Gambar 3.29



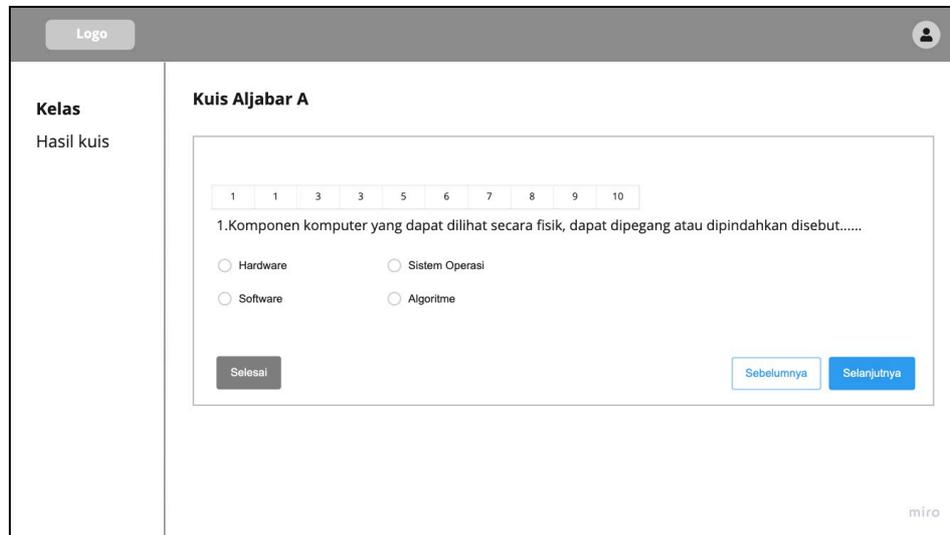
Gambar 3.29 Halaman menyerahkan kuis ke kelas

13. FE 6.1 Menampilkan Kelas Dan Kuis Mahasiswa

Halaman ini adalah halaman Ketika aktor mahasiswa mengakses daftar kelas dan selanjutnya mengakses kuis yang tersedia. Halaman ini dapat lebih jelas dilihat pada Gambar 3.24

14. FE 6.2 Menampilkan Kuis Yang Dipilih

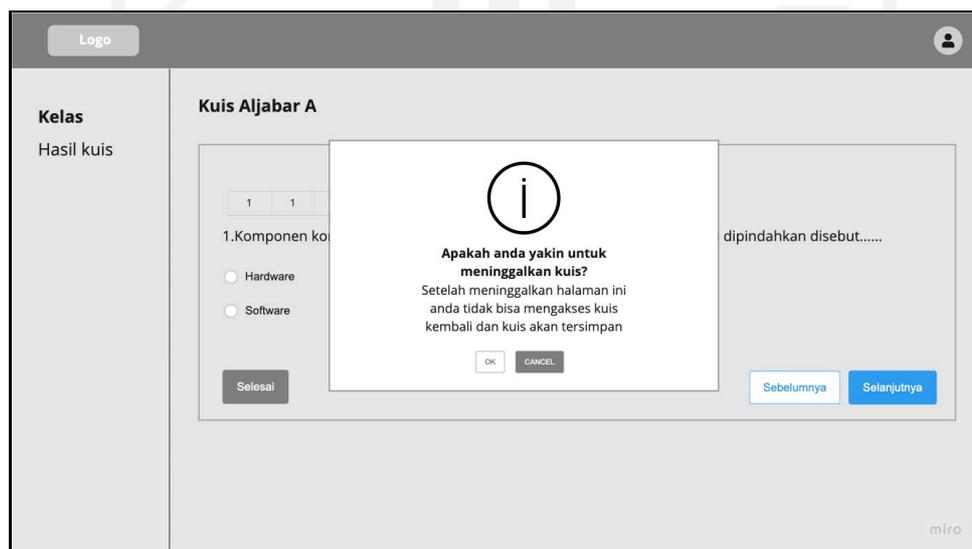
Halaman ini merupakan halaman ketika aktor mahasiswa mengakses form kuis. Mahasiswa yang telah mengklik mulai kuis akan masuk pada kuis dan dapat mulai mengerjakan kuis. Halaman ini dapat lebih jelas dilihat pada Gambar 3.30



Gambar 3.30 Menampilkan kuis yang telah dipilih

15. FE 6.3 Menyimpan Kuis

Halaman ini merupakan halaman ketika mahasiswa telah selesai mengerjakan kuis. Setelah mahasiswa menyelesaikan kuis, mahasiswa dapat mengklik tombol selesai dan kuis akan tersimpan. Halaman ini dapat lebih jelas dilihat pada Gambar 3.31



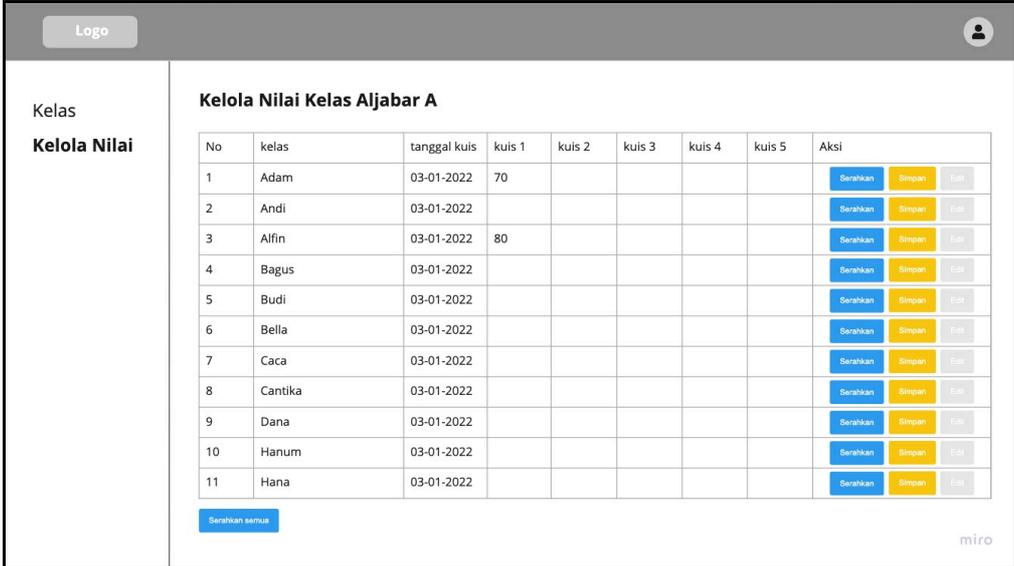
Gambar 3.31 Halaman menyimpan kuis

16. FE 7.1 Menampilkan Halaman Daftar Kelas Dosen

Halaman ini merupakan halaman awal ketika dosen ingin mengelola nilai kuis. Pertama akan ditampilkan daftar kelas yang telah ada, kemudian dosen dapat memilih pada menu sidebar untuk kelola nilai. Halaman ini dapat lebih jelas dilihat pada Gambar 3.21

17. FE 7.2 Menampilkan Daftar Mahasiswa Dan Kuis

Halaman ini adalah halaman ketika aktor dosen ingin memberikan nilai kepada setiap mahasiswa di kelas yang diampunya, dimana dosen dapat langsung mengisi nilai kepada setiap mahasiswa. Halaman ini dapat lebih jelas dilihat pada Gambar 3.32

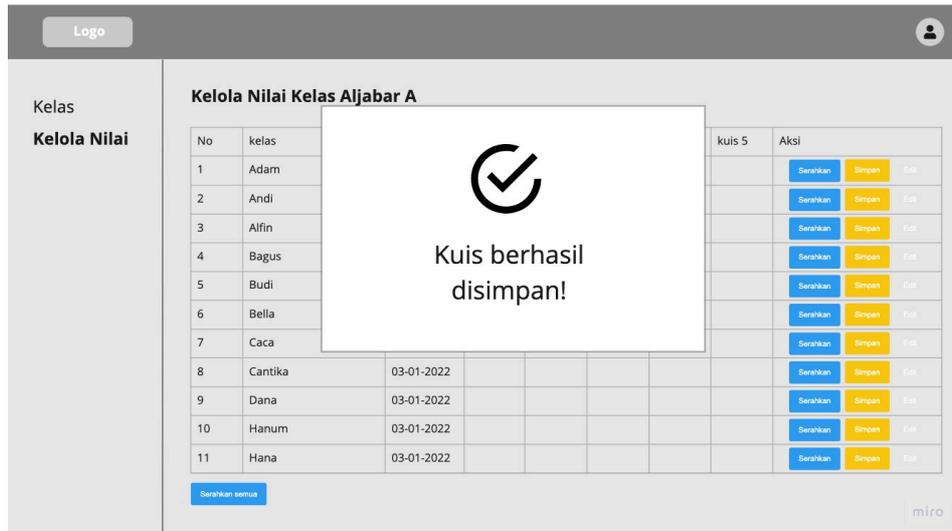


No	kelas	tanggal kuis	kuis 1	kuis 2	kuis 3	kuis 4	kuis 5	Aksi
1	Adam	03-01-2022	70					Serahkan Simpan Edit
2	Andi	03-01-2022						Serahkan Simpan Edit
3	Alfin	03-01-2022	80					Serahkan Simpan Edit
4	Bagus	03-01-2022						Serahkan Simpan Edit
5	Budi	03-01-2022						Serahkan Simpan Edit
6	Bella	03-01-2022						Serahkan Simpan Edit
7	Caca	03-01-2022						Serahkan Simpan Edit
8	Cantika	03-01-2022						Serahkan Simpan Edit
9	Dana	03-01-2022						Serahkan Simpan Edit
10	Hanum	03-01-2022						Serahkan Simpan Edit
11	Hana	03-01-2022						Serahkan Simpan Edit

Gambar 3.32 Halaman daftar mahasiswa dan kuis

18. FE 7.3 Menyimpan ke Database Kuis

Halaman ini adalah halaman ketika aktor dosen ingin menyimpan nilai kuis ke sistem. Setelah mengisi nilai untuk setiap mahasiswa dosen dapat menyimpan nilai dengan mengklik tombol serahkan nilai. Halaman ini dapat lebih jelas dilihat pada Gambar 3.33



Gambar 3.33 Halaman menyimpan ke database kuis

19. FE 8.1 Menampilkan Halaman Kelas dan Menampilkan Menu *Sidebar* Hasil Kuis

Halaman ini adalah awal halaman ketika aktor mahasiswa ingin mengakses nilai kuis. Mahasiswa dapat memilih menu sidebar hasil kuis untuk melihat nilai kuis. Halaman ini dapat lebih jelas dilihat pada Gambar 3.24.

20. FE 8.2 Halaman Menampilkan Hasil Kuis

Halaman ini adalah halaman ketika aktor mahasiswa berhasil mengakses menu melihat nilai kuis. Mahasiswa dapat melihat nilai kuis dari setiap kelas yang telah selesai melaksanakan kuis. Untuk lebih jelasnya dapat dilihat pada Gambar 3.34

The screenshot shows a web application interface displaying quiz results. A sidebar menu on the left is labeled "Hasil kuis". The main content area contains a table with the following data:

No	kelas	tanggal kuis	kuis 1	kuis 2	kuis 3	uts	kuis 4	kuis 5	Kuis 6	kuis 7
1	Aljabar A	03-01-2022	70							
2	Pemikiran Design A	06-01-2022								
3	Kalkulus A	07-01-2022	80							

Gambar 3.34 Halaman hasil kuis

BAB IV HASIL DAN PEMBAHASAN

4.1 Deploy

Deploy dan Implementasi memiliki hubungan erat karena kedua tahap ini merupakan langkah-langkah yang dibutuhkan dalam pengembangan perangkat lunak. *Deploy* sendiri memiliki pengertian mengaktifkan sebuah aplikasi agar dapat diakses secara publik dan implementasi adalah tahap penerapan sistem yang telah dirancang dan analisis. Kedua proses ini menjadi langkah terakhir dalam pengembangan perangkat lunak

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.

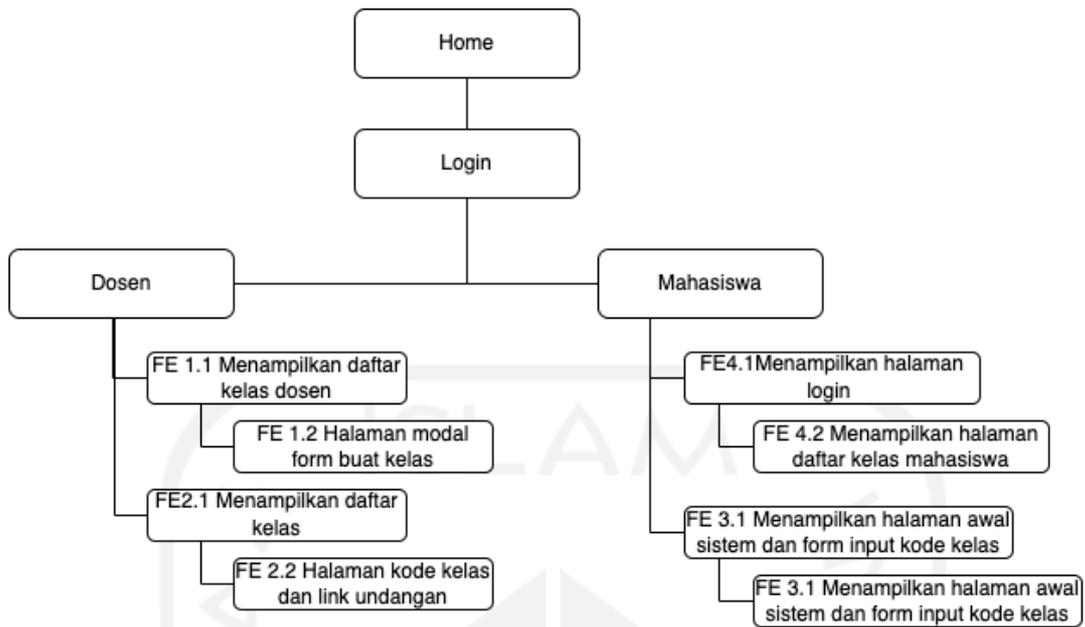
Implementasi sistem manajemen kuis hanya berfokus pada pengembangan pada sisi *front-end* dengan lebih mengembangkan tampilan visual dari sistem manajemen kuis. Sistem ini selanjutnya dapat dikembangkan kembali pada sisi *back-end* dengan menggunakan API.

4.1.1 Sitemap

Sitemap merupakan daftar isi dari sebuah situs website yang memiliki file berbasis *xml*. Sitemap juga sering disebut sebagai peta situs website. Secara umum sitemap berfungsi untuk mempermudah para pengguna untuk menelusuri halaman atau konten-konten web yang sedang dikunjungi. Sitemap juga berfungsi sebagai navigasi bagi pengunjung, menaikkan tingkat *visibilitas* website, membantu mengatur arsitektur website dan mempercepat proses index di mesin pencari.

4.1.1.1 Sitemap Pengelolaan Kelas

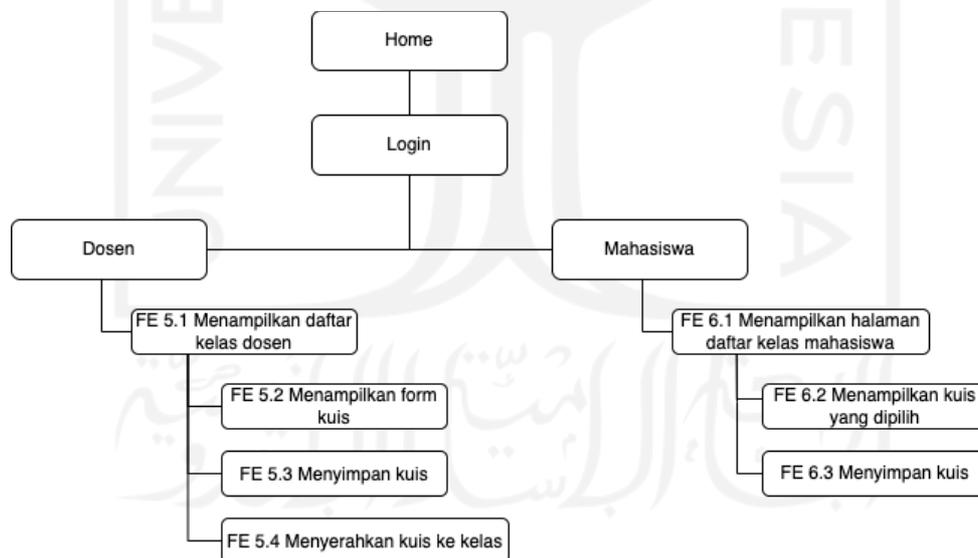
Pada Gambar 4.1, merupakan sitemap dari sistem pengelolaan kelas. Sitemap dimulai dari halaman awal login, jika dosen berhasil login, maka dosen dapat mengakses halaman kelas, buat kelas dan undang mahasiswa. Kemudian ketika mahasiswa berhasil login, maka mahasiswa dapat mengakses halaman daftar kelas dan bergabung dengan kelas.



Gambar 4.1 Sitemap sistem pengelolaan kelas

4.1.1.2 Sitemap sistem pengelolaan kuis

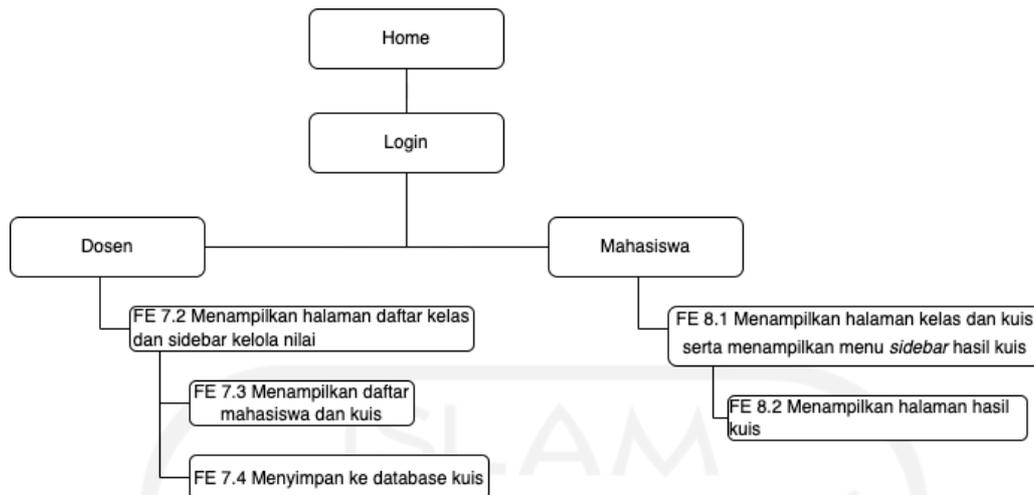
Pada Gambar 4.2, merupakan sitemap sistem pengelolaan kuis. Sitemap dimulai dengan dosen login, selanjutnya dosen dapat mengakses halaman kelas dan buat kuis. Kemudian mahasiswa yang berhasil login akan dapat mengakses halaman daftar kelas dan mulai kuis.



Gambar 4.2 Sitemap sistem pengelolaan kuis

4.1.1.3 Sitemap sistem pengelolaan nilai

Pada Gambar 4.3, merupakan sitemap sistem pengelolaan nilai. Sitemap dimulai dengan dosen melakukan login dan mengakses halaman kelola nilai. Selanjutnya bagi mahasiswa yang berhasil login dapat mengakses halaman melihat nilai.



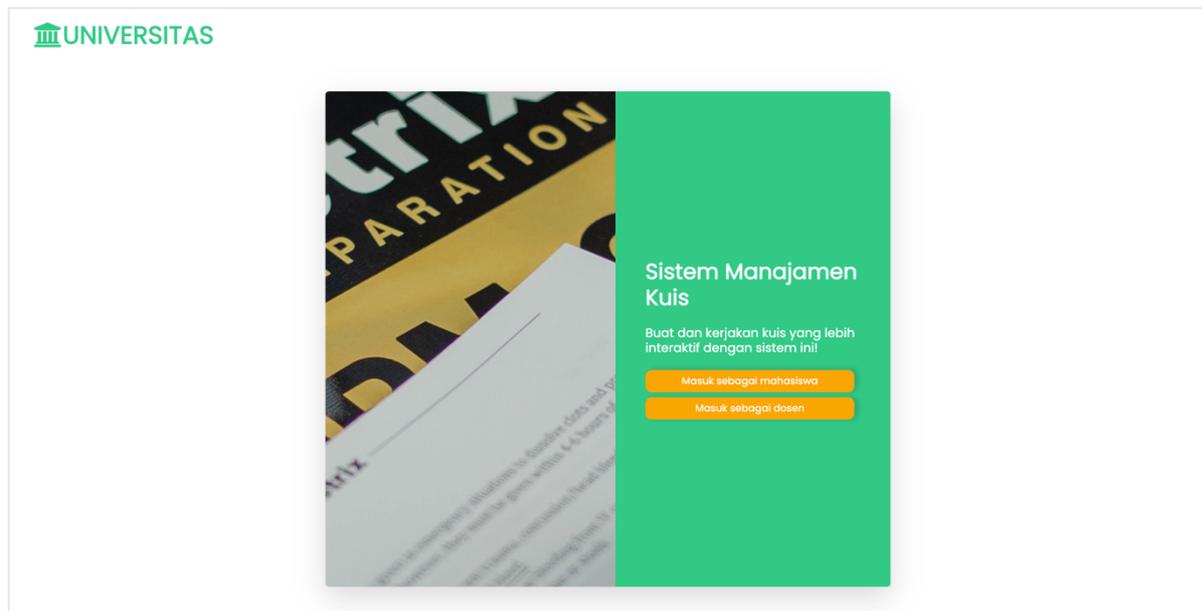
Gambar 4.3 Sitemap sistem pengelolaan nilai

4.1.2 Hasil Implementasi

Hasil implementasi sistem berdasarkan tahapan perancangan sistem yang telah dibuat sebelumnya. Implementasi sistem manajemen kuis dilakukan dengan mengembangkan pada sisi *front-end* atau tampilan sistem. Proses pembuatan *front-end* sistem manajemen kuis menggunakan *css framework* dari *bootstrap*. *Framework bootstrap* digunakan karena akan memperingkas dan mempermudah programmer dalam melakukan pengembangan suatu web. Selanjutnya untuk membuat website lebih responsive digunakan *sweet alert* sebagai pengganti alert bawaan dari *html*.

4.1.2.1 Halaman Login

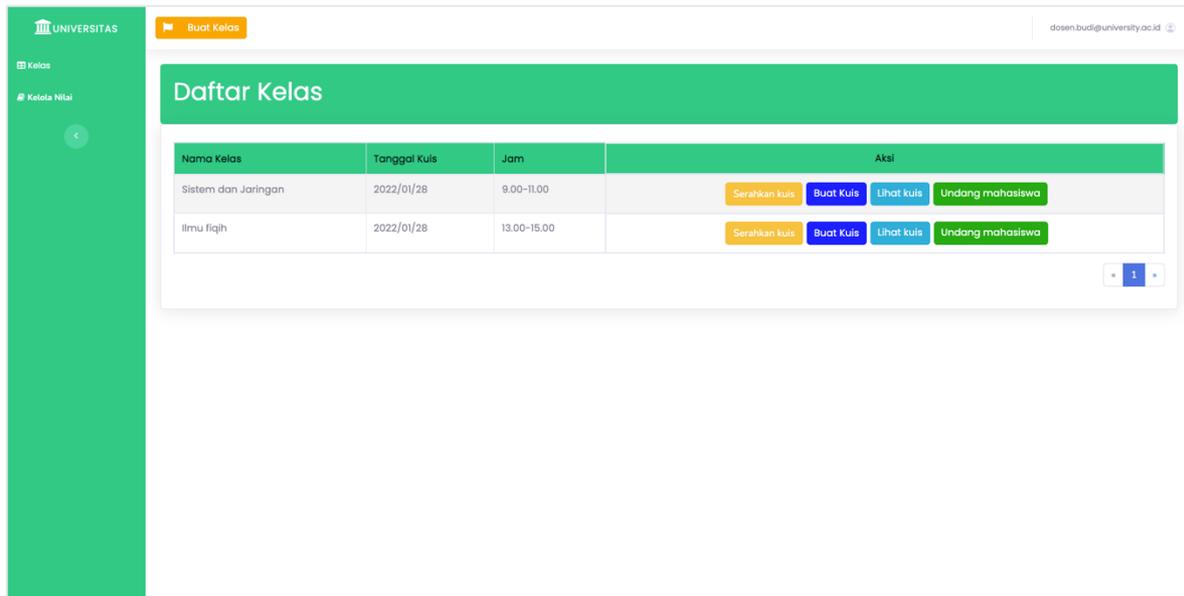
Halaman utama merupakan halaman yang dapat diakses oleh seluruh user. Halaman ini berisi satu tombol masuk yang akan mengarahkan user menuju halaman login. Halaman ini merupakan fungsionalitas pendukung yang menggunakan sistem *google identity* sebagai layanan log in. Halaman login telah sesuai dengan rancangan antar muka FE 4.2. Berikut adalah *screenshot* halaman utama yang ditunjukkan pada Gambar 4.4



Gambar 4.4 Halaman login

4.1.2.2 Halaman Daftar Kelas Dosen

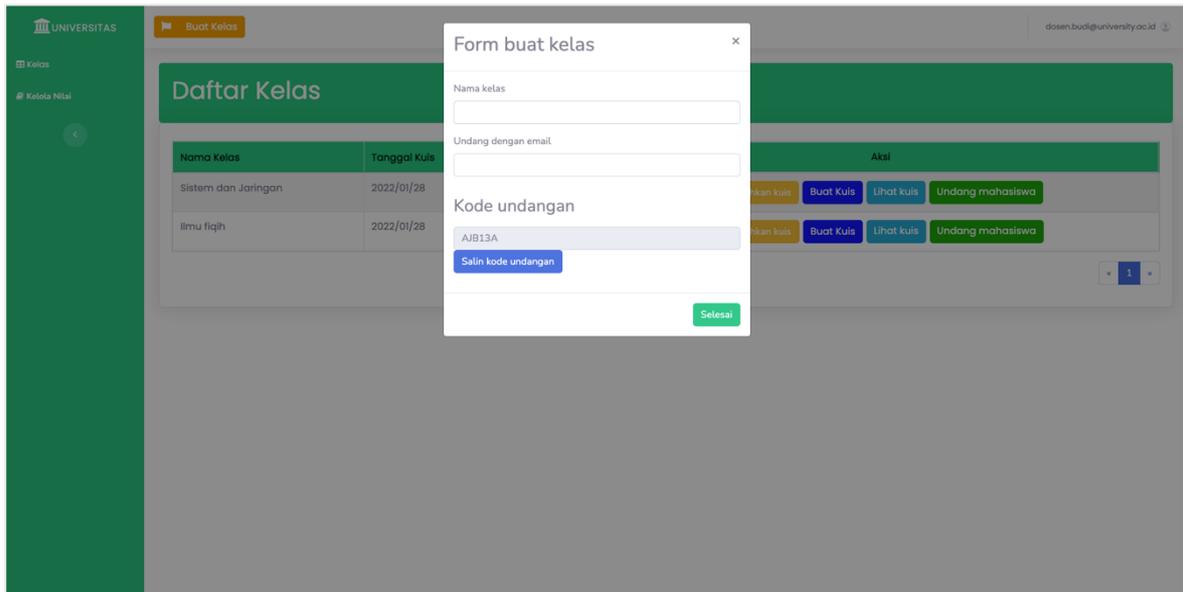
Gambar 4.5 menunjukkan halaman ketika dosen pertama kali masuk ke dalam sistem manajemen kuis. Halaman dosen melihat kelas adalah bagian dari *use case* melihat daftar kelas, halaman ini juga mempunyai tombol yang berfungsi untuk membuat kelas dan membuat kuis. Halaman daftar kelas dosen telah sesuai dengan rancangan antar muka FE 1.1 . Kemudian halaman dosen melihat kelas merupakan bagian dari service kelas dan menggunakan method GET untuk mendapatkan daftar kelas yang dimiliki seorang dosen.



Gambar 4.5 Halaman daftar kelas dosen

4.1.2.3 Halaman Dosen Membuat Kelas

Gambar 4.6 menunjukkan halaman ketika dosen ingin membuat kelas. Halaman dosen membuat membuat kelas mempunyai input nama kelas kuis, undang dengan email dan kode undangan kelas. Halaman dosen membuat kelas telah sesuai dengan rancangan antar muka FE 1.2. Kemudian halaman ini merupakan bagian dari *use case* membuat kuis yang memanfaatkan service kuis dan menggunakan method post tambah kelas. Halaman ini dapat lebih jelas dilihat pada Gambar 4.6.



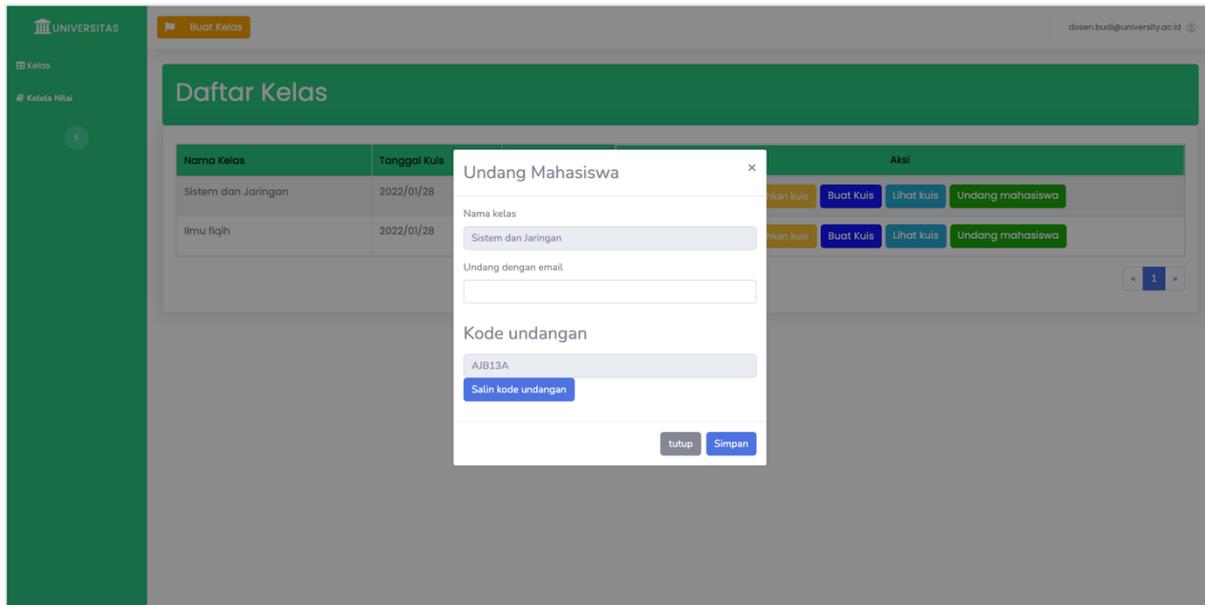
Gambar 4.6 Halaman dosen membuat kelas

4.1.2.4 Menampilkan Daftar Kelas Dosen

Halaman menampilkan daftar kelas dosen merupakan halaman ketika dosen ingin mengundang mahasiswa masuk ke kelas. Pertama akan ditampilkan daftar kelas yang telah ada, kemudian dosen dapat memilih kelas yang akan membuka link undangan bagi mahasiswa. Halaman daftar kelas dosen telah sesuai dengan rancangan antar muka FE 2.1. Halaman ini serupa dengan Gambar 4.5.

4.1.2.5 Halaman Dosen Mengundang Mahasiswa

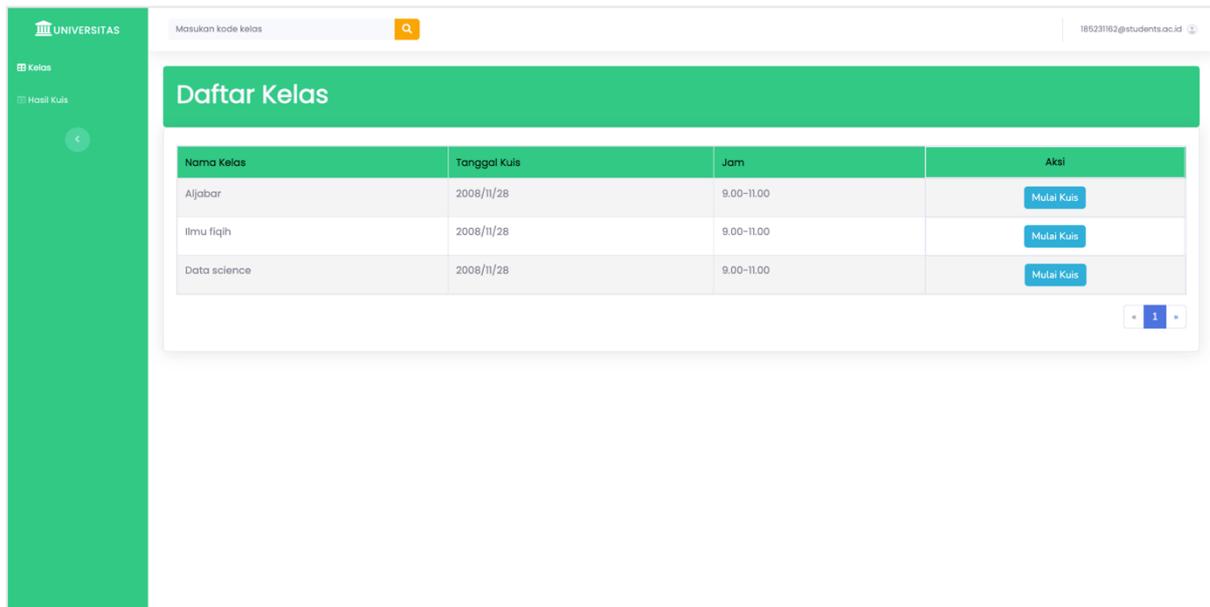
Gambar 4.7 menunjukkan halaman ketika dosen ingin mengundang mahasiswa. Halaman dosen mengundang mahasiswa akan menampilkan nama kelas dan kode undangan kelas yang dapat dicopy. Halaman dosen mengundang mahasiswa telah sesuai dengan rancangan antar muka FE 2.2. Kemudian halaman ini merupakan bagian dari *use case* mengundang mahasiswa yang memanfaatkan service kuis dan menggunakan method post. Halaman ini dapat lebih jelas dilihat pada Gambar 4.7



Gambar 4.7 Halaman dosen mengundang mahasiswa

4.1.2.6 Halaman Melihat Kelas dan Jadwal Kuis Mahasiswa

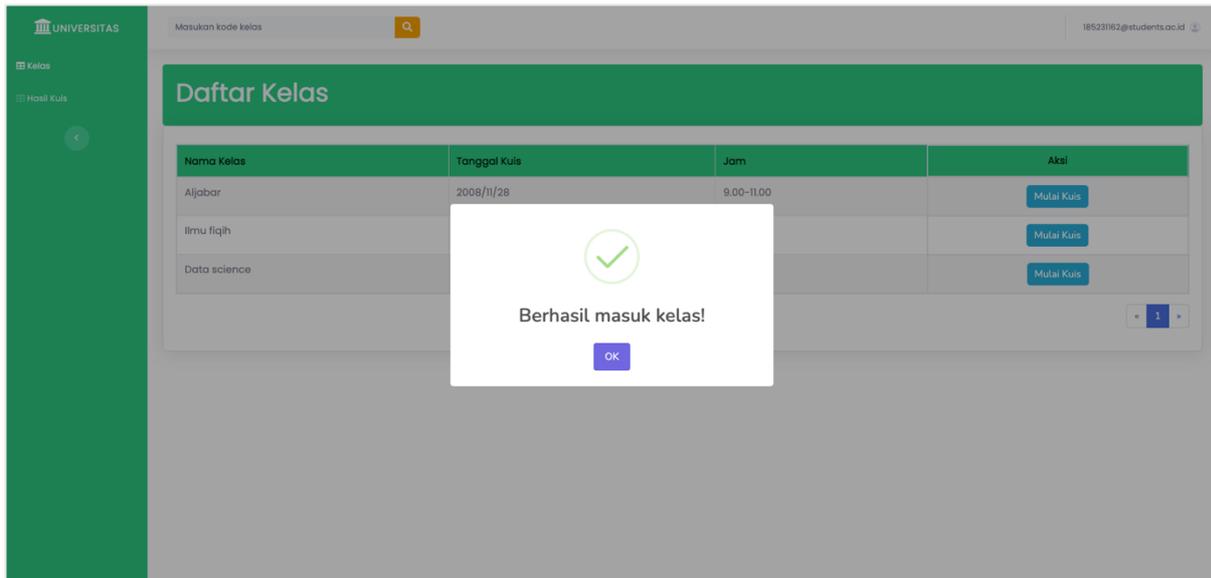
Gambar 4.8 merupakan halaman awal ketika mahasiswa berhasil masuk ke sistem manajemen kuis. Sistem akan menampilkan tabel daftar kelas dan jadwal kuis yang diambil oleh mahasiswa. Halaman melihat kelas dan jadwal kuis mahasiswa telah sesuai dengan rancangan antar muka FE 3.1. Kemudian juga halaman ini merupakan bagian dari *use case* melihat kelas yang memanfaatkan service kelas dan menggunakan methodGet. Halaman ini dapat lebih jelas dilihat pada Gambar 4.8.



Gambar 4.8 Halaman daftar kelas mahasiswa

4.1.2.7 Halaman Mahasiswa Masuk Kelas

Gambar 4.9 merupakan halaman ketika mahasiswa berhasil masuk ke kelas dengan menggunakan kode kelas. Sistem akan menampilkan notifikasi atau alert ketika mahasiswa berhasil masuk ke kelas. Halaman mahasiswa masuk kelas telah sesuai dengan rancangan antar muka FE 3.2. Kemudian juga halaman ini merupakan bagian dari *use case* melihat kelas yang memanfaatkan service kelas dan menggunakan method get. Halaman ini dapat lebih jelas dilihat pada Gambar 4.9.



Gambar 4.9 Halaman mahasiswa masuk kelas

4.1.2.8 Halaman Daftar Kelas dan Jadwal Kuis

Halaman ini merupakan halaman awal Ketika aktor mahasiswa berhasil melakukan login. Halaman ini akan menampilkan daftar kelas dan jadwal kuis untuk mahasiswa. Halaman daftar kelas dan jadwal kuis telah sesuai dengan rancangan antar muka FE 4.2. Halaman ini serupa dengan Gambar 4.8.

4.1.2.9 Halaman Daftar Kelas Dosen

Halaman ini merupakan halaman ketika dosen ingin membuat kuis. Pertama akan ditampilkan daftar kelas yang telah ada, kemudian dosen dapat memilih kelas yang akan dibuat kuis. Halaman daftar kelas dosen telah sesuai dengan rancangan antar muka FE 5.1. Kemudian dosen dapat mengklik tombol buat kuis. Halaman ini serupa dengan Gambar 4.5.

4.1.2.10 Halaman Dosen Membuat Kuis

Gambar 4.10 menunjukkan halaman ketika dosen ingin membuat kuis. Halaman dosen membuat kuis mempunyai text input judul kuis dan pertanyaan sedangkan jawaban kuis menggunakan radiobutton. Halaman dosen membuat kuis telah sesuai dengan rancangan antar muka FE 5.2. Kemudian halaman ini merupakan bagian dari *use case* membuat kuis yang memanfaatkan service kuis dan menggunakan method get. Halaman ini dapat lebih jelas dilihat pada Gambar 4.10.

UNIVERSITAS

Buat Kelas

dosen.budi@university.ac.id

Buat Kuis

Home Buat Kuis Lihat Kuis

Judul Kuis

Tanggal kuis
dd/mm/yyyy

Jam Mulai
--:--

Jam Selesai
--:--

Pertanyaan

Choose file | No file chosen

- Pilihan 2
- Pilihan 2
- Pilihan 2
- Pilihan 2

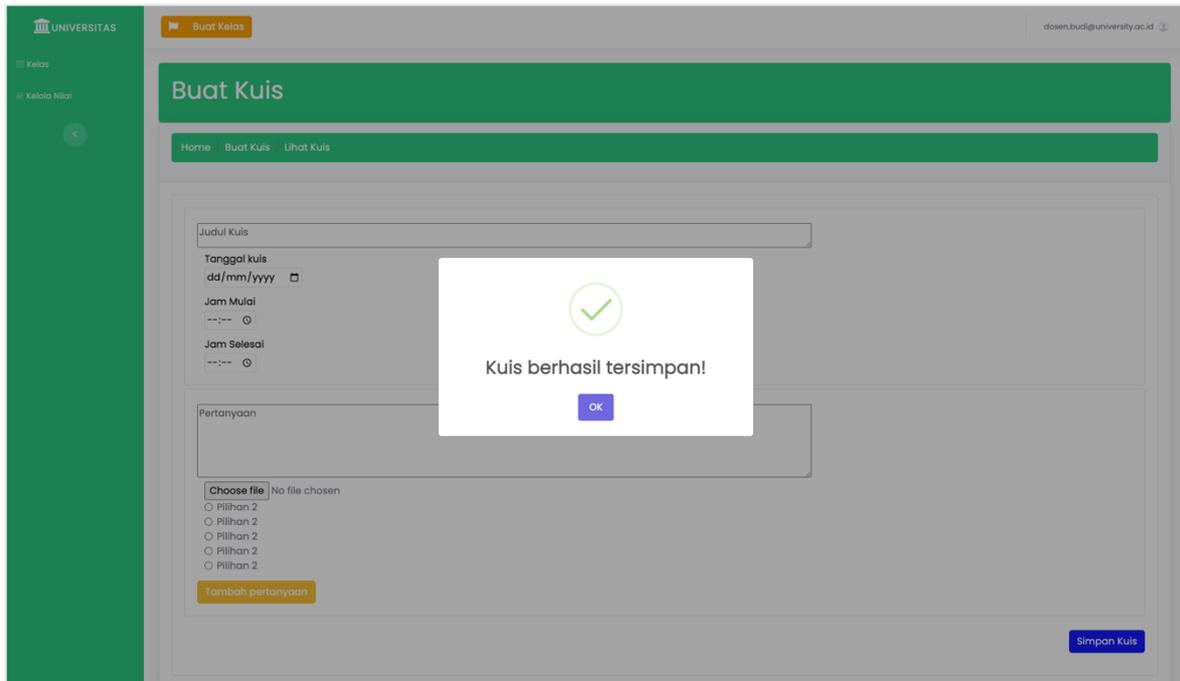
Tambah pertanyaan

Simpan Kuis

Gambar 4.10 Halaman dosen membuat kuis

4.1.2.11 Halaman Dosen Menyimpan Kuis

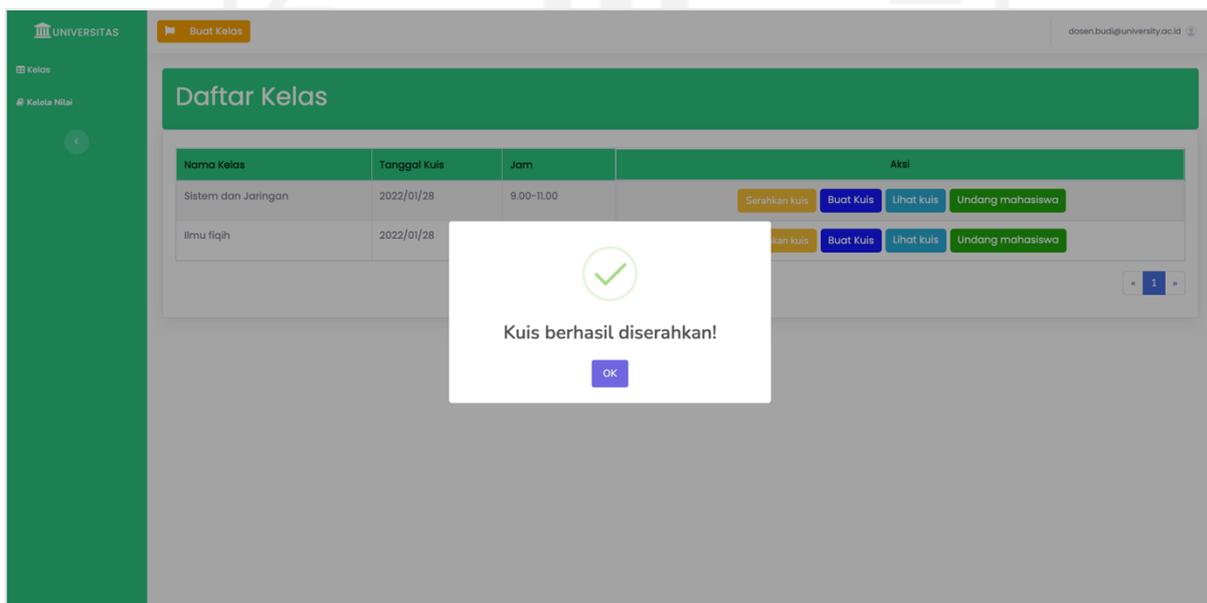
Gambar 4.11 merupakan halaman pada saat aktor dosen telah mengisi semua form pembuatan kuis. Setelah semua form telah terisi, dosen dapat menyimpan kuis dan mengklik tombol simpan kuis. Halaman dosen menyimpan kuis telah sesuai dengan rancangan antarmuka FE 5.3. Halaman ini dapat lebih jelas dilihat pada Gambar 4.11.



Gambar 4.11 Halaman dosen menyimpan kuis

4.1.2.12 Halaman Dosen Menyerahkan Kuis ke Kelas

Halaman ini merupakan halaman pada saat akor dosen telah menyimpan kuis. Setelah kuis tersimpan dosen dapat menyerahkan ke kelas dengan mengklik tombol serahkan kuis. Halaman dosen menyerahkan kuis ke kelas telah sesuai dengan rancangan antar muka FE 5.4. Halaman ini dapat lebih jelas dilihat pada Gambar 4.12



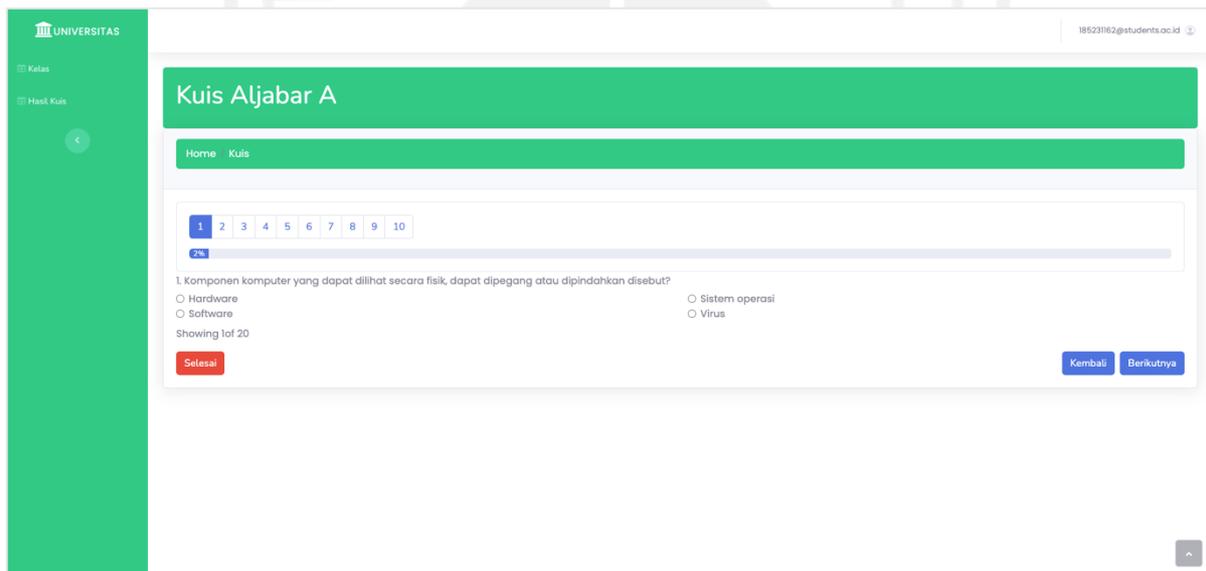
Gambar 4.12 Halaman dosen menyerahkan kuis ke kelas

4.1.2.13 Halaman Kelas dan Kuis Mahasiswa

Halaman ini merupakan halaman Ketika aktor mahasiswa mengakses daftar kelas dan selanjutnya mengakses kuis yang tersedia. Halaman menampilkan kelas dan kuis mahasiswa telah sesuai dengan rancangan antar muka FE 6.1. Halaman ini serupa dengan Gambar 4.8.

4.1.2.14 Halaman Mengerjakan Kuis

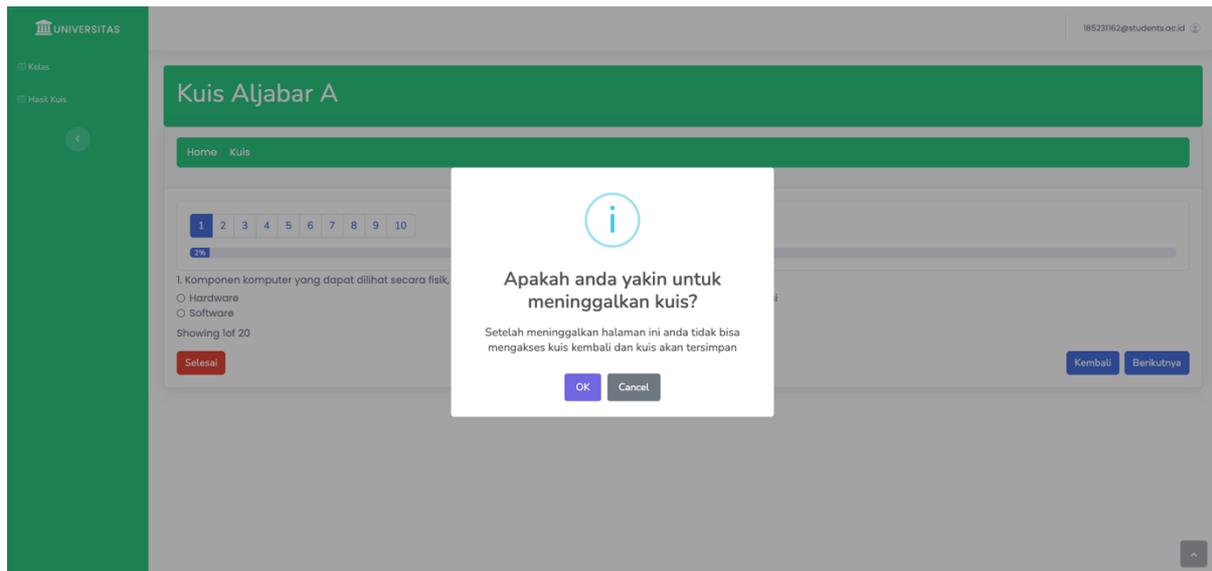
Gambar 4.13 merupakan halaman ketika mahasiswa mengerjakan kuis. Halaman ini merupakan bagian dari *use case* mengerjakan kuis yang menggunakan service kuis dan menggunakan method GET kuis. Halaman menampilkan halaman mengerjakan kuis telah sesuai dengan rancangan antar muka FE 6.2. Halaman ini dapat lebih jelas dilihat pada Gambar 4.13.



Gambar 4.13 Halaman mahasiswa mengerjakan kuis

4.1.2.15 Halaman Menyimpan Kuis

Halaman ini merupakan halaman ketika mahasiswa telah selesai mengerjakan kuis. Setelah mahasiswa menyelesaikan kuis, mahasiswa dapat mengklik tombol selesai dan kuis akan tersimpan. Halaman menyimpan kuis mahasiswa telah sesuai dengan rancangan antar muka FE 6.3. Halaman ini dapat lebih jelas dilihat pada Gambar 4.14



Gambar 4.14 Halaman mahasiswa menyimpan kuis

4.1.2.16 Halaman Daftar Kelas Dosen

Halaman daftar kelas dosen merupakan halaman awal ketika dosen ingin mengelola nilai kuis. Pertama akan ditampilkan daftar kelas yang telah ada, kemudian dosen dapat memilih pada menu sidebar untuk kelola nilai. Halaman menampilkan daftar kelas dosen telah sesuai dengan rancangan antar muka FE 7.1. Halaman daftar kelas dosen serupa dengan Gambar 4.5.

4.1.2.17 Halaman Dosen Mengelola Nilai

Halaman dosen mengelola nilai merupakan 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*. Halaman dosen mengelola nilai telah sesuai dengan rancangan antar muka FE 7.2. Halaman ini dapat lebih jelas dilihat pada Gambar 4.15.

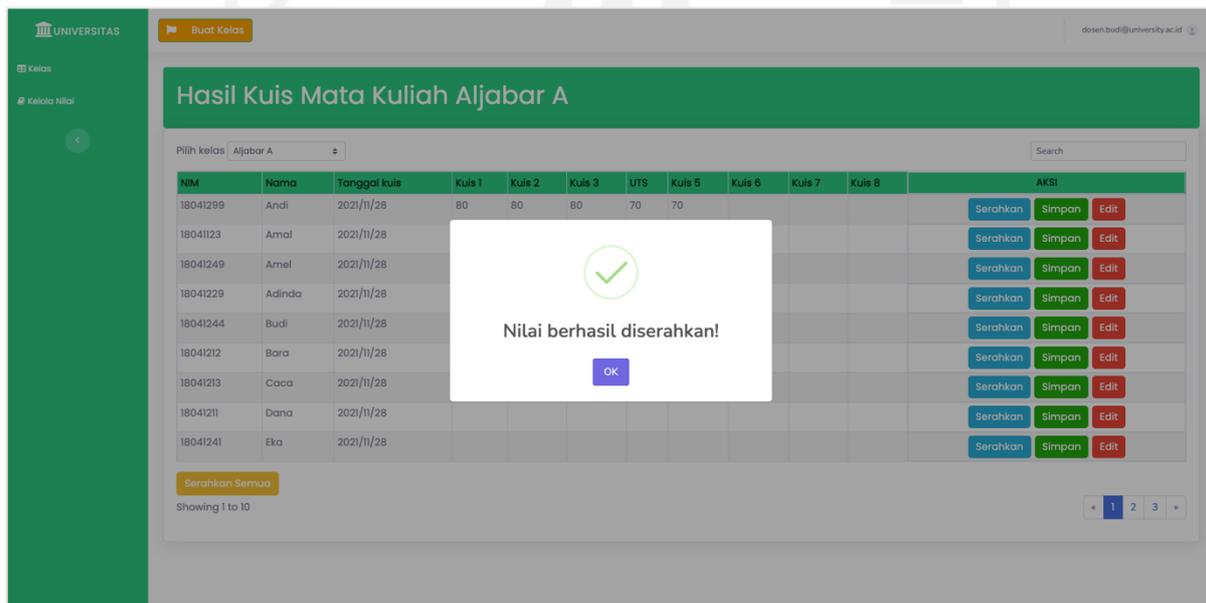
The screenshot shows a web interface for managing exam results. The main heading is 'Hasil Kuis Mata Kuliah Aljabar A'. Below the heading, there is a search bar and a table with columns for NIM, Nama, Tanggal kuis, Kuis 1, Kuis 2, Kuis 3, UTS, Kuis 5, Kuis 6, Kuis 7, Kuis 8, and AKSI. The AKSI column contains three buttons: 'Serahkan' (blue), 'Simpan' (green), and 'Edit' (red). A 'Serahkan Semua' button is located at the bottom left of the table area. The page also includes a sidebar with 'UNIVERSITAS' and 'Kelola Nilai' options, and a top navigation bar with 'Buat Kelas' and a user email 'dosen.budi@university.ac.id'.

NIM	Nama	Tanggal kuis	Kuis 1	Kuis 2	Kuis 3	UTS	Kuis 5	Kuis 6	Kuis 7	Kuis 8	AKSI
18041299	Andi	2021/11/28	80	80	80	70	70				Serahkan Simpan Edit
18041123	Amal	2021/11/28	70	80	80	80					Serahkan Simpan Edit
18041249	Amel	2021/11/28	98	80	80	80					Serahkan Simpan Edit
18041229	Adinda	2021/11/28									Serahkan Simpan Edit
18041244	Budi	2021/11/28									Serahkan Simpan Edit
18041212	Bara	2021/11/28									Serahkan Simpan Edit
18041213	Caca	2021/11/28									Serahkan Simpan Edit
18041211	Dana	2021/11/28									Serahkan Simpan Edit
18041241	Eka	2021/11/28									Serahkan Simpan Edit

Gambar 4.15 Halaman Dosen Mengelola Nilai

4.1.2.18 Halaman Dosen Menyimpan nilai

Halaman dosen menyimpan nilai merupakan halaman ketika aktor dosen ingin menyimpan nilai kuis ke sistem. Setelah mengisi nilai untuk setiap mahasiswa, dosen dapat menyimpan nilai dengan mengklik tombol serahkan nilai. Halaman dosen menyimpan nilai telah sesuai dengan rancangan antar muka FE 7.3. Halaman ini dapat lebih jelas dilihat pada Gambar 4.16



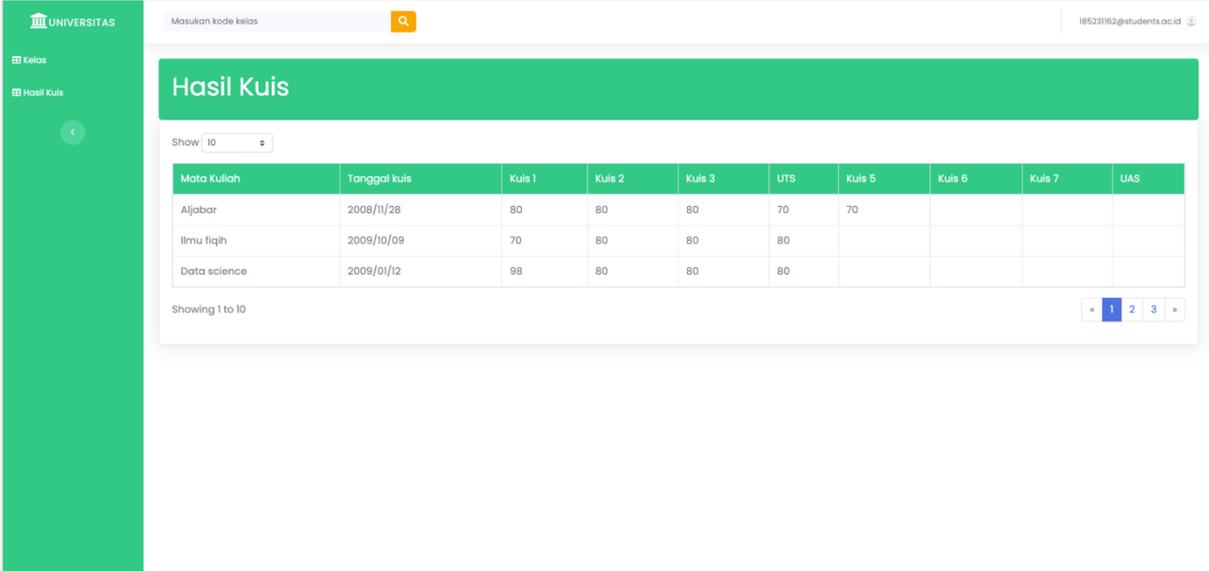
Gambar 4.16 Halaman Dosen Menyimpan Nilai

4.1.2.19 Halaman Kelas dan Menampilkan Menu *sidebar* Hasil Kuis

Halaman ini merupakan awal halaman ketika aktor mahasiswa ingin mengakses nilai kuis. Mahasiswa dapat memilih menu sidebar hasil kuis untuk melihat nilai kuis. Halaman kelas dan menampilkan menu sidebar hasil kuis telah sesuai dengan rancangan antar muka FE 8.1. Halaman ini dapat lebih jelas dilihat pada Gambar 4.8.

4.1.2.20 Halaman Melihat Hasil Kuis

Gambar 4.17 merupakan halaman hasil kuis mahasiswa yang telah selesai mengikuti kuis. Halaman ini adalah bagian dari *use case* melihat nilai yang memanfaatkan service nilai dan menggunakan method POST nilai. Halaman melihat hasil kuis telah sesuai dengan rancangan antar muka FE 8.2. Halaman ini dapat dilihat lebih jelas pada Gambar 4.17.



Mata Kuliah	Tanggal kuis	Kuis 1	Kuis 2	Kuis 3	UTS	Kuis 5	Kuis 6	Kuis 7	UAS
Aljabar	2008/11/28	80	80	80	70	70			
Ilmu fiqih	2009/10/09	70	80	80	80				
Data science	2009/01/12	98	80	80	80				

Gambar 4.17 Halaman melihat hasil kuis

4.2 Manage

Manage atau mengelola adalah 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.

4.2.1 Pengujian Front-End terkait Fungsionalitas Sistem

Pengujian fungsionalitas sistem digunakan untuk menguji komponen-komponen yang ada pada sistem yang telah dibuat. Tahapan pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan tahapan perancangan yang telah dilakukan. Pengujian perancangan sistem dapat dilihat pada Tabel 4.1 Pengujian fungsionalitas sistem.

Tabel 4.1 Pengujian fungsionalitas sistem

No	Use case	Fungsionalitas	Navigasi Halaman	Penilaian
				Sesuai (√) atau Tidak sesuai (X)
1	UC1 Membuat kelas	Melihat Daftar Kelas	Halaman daftar kelas	√
		Mengakses modal form buat kelas dan link undangan kelas	Halaman modal form buat kelas dan link undangan kelas	√
2	UC2 Mengundang mahasiswa	Menampilkan daftar kelas	Halaman daftar kelas	√
		Mengakses Halaman kode kelas dan link undangan	Halaman kode kelas dan link undangan	√
3	UC3 Bergabung dengan kelas	Menampilkan halaman awal sistem dan form input kode kelas	Halaman awal sistem dan form input kode kelas	√
		Menghasilkan alert sukses	Halaman Menghasilkan alert sukses	√
4	UC4 Melihat kelas	Menampilkan halaman login	Halaman login	√
		Menampilkan daftar kelas	Halaman daftar kelas	√
5	UC5 Membuat kuis	Menampilkan halaman daftar kelas dosen	Halaman daftar kelas dosen	√
		Menampilkan form kuis	Halaman form kuis	√
		Menyimpan kuis	Halaman menyimpan kuis	√
		Menyerahkan kuis ke kelas	Halaman menyerahkan kuis ke kelas	√
6	UC6 Mengerjakan kuis	Menampilkan tabel kelas dan kuis mahasiswa	Halaman menampilkan tabel kelas dan kuis mahasiswa	√

		Menampilkan kuis yang dipilih	Halaman menampilkan kuis yang dipilih	√
		Menyimpan kuis	Halaman menyimpan kuis	√
7	UC7 Menilai kuis	Menampilkan halaman daftar kelas dan sidebar kelola nilai	Halaman daftar kelas dan sidebar kelola nilai	√
		Menampilkan daftar mahasiswa dan kuis	Menampilkan daftar mahasiswa dan kuis	√
8	UC8 Melihat nilai	Menampilkan halaman kelas dan kuis dan menampilkan menu sidebar hasil kuis	Halaman kelas dan kuis dan menampilkan menu sidebar hasil kuis	√
		Menampilkan halaman hasil kuis	Halaman menampilkan halaman hasil kuis	√

Dari hasil uji fungsionalitas sistem yang telah dilakukan menghasilkan bahwa semua fungsionalitas sistem telah berjalan dengan baik dan sesuai. Setiap komponen-komponen sistem telah berhasil bekerja sesuai dengan fungsionalitas masing-masing. Setelah dilakukan proses pengujian didapatkan bahwa pengujian fungsionalitas sistem telah berhasil.

4.2.2 Pengujian Front End

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 analisis data dengan merekam atau mencatat hasil dari *usability testing*, melakukan pengukuran *usability* dengan kuesioner dan melakukan analisis untuk menghasilkan rekomendasi perbaikan sistem ke depannya. Pengujian telah dilakukan dengan melakukan wawancara dengan dua orang *front-end developer*. Berikut pada Tabel 4.2 yang berisi nama penguji dan perusahaan tempat bekerja.

Tabel 4.2 Nama Penguji

No	Nama	Perusahaan
1	Azka	<i>scoters</i>
2	Adimas	Telkom University

Dua orang *front-end developer* dihadirkan karena mereka sudah cukup berpengalaman pada bidangnya dengan pengalaman hampir dua tahun menekuni *Javascript* dan *React*. 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 pertanyaan wawancara dapat dilihat pada Tabel 4.3 (Setiawati,2018).

Tabel 4.3 Daftar aspek dan pertanyaan

No	Aspek	Pertanyaan
1	Visibilitas status sistem (<i>visibility of system status</i>)	Ada konsistensi antar tampilan mulai dari warna sampai pada <i>icon-icon</i> yang dipakai.
		Ada <i>feedback</i> yang diberikan sistem setiap kali sebuah aksi.
		Setelah pengguna menyelesaikan suatu tahap ada <i>feedback</i> yang memberikan indikasi bahwa tahap berikutnya dapat dilakukan.
2	Kecocokan antara sistem dengan dunia nyata (<i>match between system and the real world</i>)	Setiap icon mudah dimengerti dan dipahami maksudnya (familiar).
		Menu-menu berada pada posisi yang logis dan sesuai dengan langkah kerja sebenarnya.
		Kata dalam petunjuk cocok dengan nama kunci sebenarnya.
3	Kebebasan dan kendali pengguna terhadap sistem	Pengguna dapat mengedit data pada kata kunci, atau melakukan perintah <i>copy- paste</i> ?
4	Konsisten dan baku (<i>consistency and standard</i>).	Setiap <i>icon</i> diberi label.
		Perintah digunakan dengan cara yang sama, dan apakah itu berarti hal yang sama, dalam semua bagian dari sistem.
5	Pencegahan kesalahan (<i>error prevention</i>).	Sistem memperingatkan pengguna jika aksi yang mereka buat berpotensi menimbulkan kesalahan serius
6	Mudah dikenal dibandingkan mengingat (<i>recognition rather than recall</i>).	Peringatan dan pesan dimunculkan di posisi mata sering melihat layar?
		Terdapat banyak data yang membingungkan bila memungkinkan dieliminasi
		Item menu tidak aktif berwarna abu-abu atau dihilangkan

7	Estetika dan desain minimalis (<i>esthetic and minimalist design</i>).	Hanya informasi penting saja yang ditampilkan di layar
	Membantu pengguna untuk mengenal, mendiagnosa dan menghilangkan kesalahan (<i>help users recognize, diagnose, and recover from error</i>).	Pesan kesalahan dinyatakan jelas akibat dari kesalahan pengguna ? Sehingga pengguna mengerti kesalahan apa yang telah diperbuat ?

Tahap berikutnya adalah tahapan wawancara kepada penguji. Wawancara dilakukan agar penguji mendapatkan data yang lebih spesifik dari pengujian yang telah dilakukan. Hasil dari wawancara tersebut dapat dilihat pada Tabel 4.5. Keterangan masing-masing tester dapat disimbolkan dengan warna pada Tabel 4.4.

Tabel 4.4 Keterangan warna respon

No	Nama warna	Warna	Keterangan
1	Hijau		Sangat mudah, sangat bagus, sangat bermanfaat, sangat cepat, sangat baik, sangat membantu
2	Kuning		Biasa saja
3	Merah		Tidak mudah, tidak bagus, tidak bermanfaat, tidak cepat, tidak baik, tidak membantu

Tabel 4.5 Wawancara front-end developer

Aspek	Pertanyaan	Jawaban Tester 1	Jawaban Tester 2
Visibilitas status sistem (<i>visibility of system status</i>)	Ada konsistensi antar tampilan mulai dari warna sampai pada <i>icon-icon</i> yang dipakai.	Sudah konsisten	Tampilan konsisten, warna dan icon cukup serasi
	Ada <i>feedback</i> yang diberikan sistem setiap kali sebuah aksi.	Ada beberapa menu yang belum ada <i>feedback</i>	Ada menu yang belum menggunakan <i>feedback</i>
	Setelah pengguna menyelesaikan suatu tahap ada <i>feedback</i> yang memberikan indikasi bahwa tahap berikutnya dapat dilakukan.	Terdapat <i>feedback</i> yang telah sesuai	Terdapat <i>feedback</i> yang telah sesuai

Kecocokan antara sistem dengan dunia nyata (<i>match between system and the real world</i>)	Setiap icon mudah dimengerti dan dipahami maksudnya (familiar).	Mudah dipahami	Icon mudah dipahami
	Menu-menu berada pada posisi yang logis dan sesuai dengan langkah kerja sebenarnya.	Menu sudah sesuai	Posisi menu sudah sesuai dan mudah untuk dijangkau
Kebebasan dan kendali pengguna terhadap sistem	Pengguna dapat mengedit data pada kata kunci, atau melakukan perintah <i>copy-paste</i> ?	Dapat	Dapat
Konsisten dan baku (<i>consistency and standard</i>).	Setiap <i>icon</i> diberi label.	Sudah diberi label	Sudah diberi label
	Perintah digunakan dengan cara yang sama, dan apakah itu berarti hal yang sama, dalam semua bagian dari sistem.	Perintah sudah sama	Perintah sudah sama
Pencegahan kesalahan (<i>error prevention</i>).	Sistem memperingatkan pengguna jika aksi yang mereka buat berpotensi menimbulkan kesalahan serius	Sudah ada peringatan	Sudah ada peringatan
Mudah dikenal dibandingkan mengingat (<i>recognition rather than recall</i>).	Peringatan dan pesan dimunculkan di posisi mata sering melihat layar?	Sudah benar	Sudah benar
	Terdapat banyak data yang membingungkan bila memungkinkan dieliminasi	Cukup sederhana, tidak membingungkan	Tampilan data cukup sederhana
	Item menu tidak aktif berwarna abu-abu atau dihilangkan	Ada beberapa menu belum	Ada beberapa menu belum
Estetika dan desain minimalis (<i>esthetic and minimalist design</i>).	Hanya informasi penting saja yang ditampilkan di layar	Informasi sudah cukup jelas ditampilkan	Informasi yang ditampilkan sudah cukup informatif
Membantu pengguna untuk mengenal, mendiagnosa dan menghilangkan kesalahan (<i>help users recognize, diagnose, and recover from error</i>).	Pesan kesalahan dinyatakan jelas akibat dari kesalahan pengguna ? Sehingga pengguna mengerti kesalahan apa yang telah diperbuat ?	Pesan kesalahan tepat, sudah sesuai dengan kesalahan pengguna	Pesan kesalahan sudah tepat, isi pesan kesalahan sesuai dengan kesalahan penggunaan

Dari hasil pengujian penguji pertama didapatkan 12 aspek sudah bagus dan baik dan 2 aspek tidak bagus dan tidak membantu. Selanjutnya dari penguji kedua didapatkan hasil 11 aspek sudah bagus dan baik, 1 aspek biasa saja, dan 2 aspek tidak bagus dan tidak membantu. Kemudian dari hasil pengujian aspek tampilan dari dua orang penguji dapat disimpulkan bahwa sistem manajemen sudah memenuhi aspek tampilan pengguna walaupun ada beberapa aspek yang belum diterapkan.

Hasil dari pengujian menunjukkan bahwa sistem manajemen kuis sudah memenuhi aspek-aspek dalam pengembangan tampilan pengguna. Tetapi masih ada catatan negatif yang diberikan oleh penguji yaitu dari aspek Visibilitas status sistem (*visibility of system status*), terdapat beberapa menu yang belum memiliki feedback ketika melakukan sebuah aksi, hal ini menjadi perhatian karena akan mengakibatkan berkurangnya rasa interaksi antara pengguna dengan sistem. Selanjutnya dari aspek mudah dikenal dibandingkan mengingat (*recognition rather than recall*), Terdapat menu tidak aktif yang tidak berwarna abu-abu atau dihilangkan, Hal tersebut dapat membingungkan pengguna ketika berinteraksi dengan sistem, karena pengguna dapat bingung dan tidak mengetahui berada pada menu aktif yang mana. Kemudian sebagai rencana tindak lanjut, semua catatan negatif dari penguji akan menjadi bahan perbaikan dan evaluasi dari sistem manajemen kuis dan tentunya juga akan dilakukan penambahan menu yang belum tersedia.

Wawancara yang dilakukan dengan dua orang *front-end developer* tersebut, secara umum 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.

4.2.3 Pengujian Kelayakan Pengembangan Lanjutan (*Back-End*)

Kemudian juga telah dilakukan wawancara dengan *back-End developer* untuk menguji apakah *front-end* dapat dikembangkan lebih lanjut pada sisi *back end*. Pengujian dilakukan dengan melakukan wawancara dan penunjukan 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 dapat dikembangkan lebih lanjut pada sisi *back end*.

4.2.4 Pengujian Prinsip SOA

Dalam pengujian selanjutnya dilakukan untuk mencari tahu prinsip dan sifat SOA yang telah terpenuhi oleh sistem. Pengujian sifat SOA dapat dilihat pada Tabel 4.6.

Tabel 4.6 Pengujian prinsip SOA

No	Prinsip SOA	Pengertian	Penguji 1
			Diterapkan (√) atau Belum diterapkan(X)
1	<i>Loosely coupled</i>	<i>Service</i> tersebut mempertahankan sebuah hubungan yang meminimalisir ketergantungan dan mereka hanya perlu menjaga kesadaran antar satu sama lain.	√
2	<i>Service contract</i>	<i>Service</i> melekat dan taat pada sebuah kesepakatan komunikasi, yang didefinisikan secara kolektif oleh satu atau lebih deskripsi <i>service</i> dan dokumen yang berhubungan.	√
3	<i>Autonomy</i>	<i>Service</i> mempunyai kendali berdasarkan logika yang dienkapsulasi.	√
4	<i>Abstraction</i>	<i>Services</i> menyembunyikan logika dari dunia luar.	X
5	<i>Reusability</i>	logika terbagi menjadi <i>service</i> dengan tujuan untuk digunakan kembali.	√
6	<i>Composability</i>	kumpulan dari <i>service</i> dapat dikoordinasikan dan dihimpun untuk membentuk <i>services</i> yang berbeda.	X
7	<i>Statelessness</i>	<i>service</i> meminimalisasi sifat berpegang teguh pada informasi tertentu untuk sebuah aktivitas.	X
8	<i>Discoverability</i>	<i>services</i> dirancang dengan sifat yang deskriptif sehingga mereka dapat ditemukan dan ditentukan dengan menggunakan mekanisme-mekanisme penemuan yang ada.	√

Hasil dari pengujian oleh penguji didapatkan bahwa sistem manajemen kuis telah memenuhi 5 dari 8 prinsip SOA atau sebesar 62.5%. Kemudian dilakukan pengujian SOA kembali oleh peneliti dengan hasil sebagai berikut:

- a. *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. Dapat dilihat pada Gambar 3.4 pada rancangan arsitektur sistem manajemen kuis telah memiliki tiga *service*.
- b. *Service contract*, *services* melekat dan taat pada sebuah kesepakatan komunikasi, yang didefinisikan secara kolektif oleh satu atau lebih deskripsi *service* dan dokumen yang berhubungan. Sebagai kesepakatan komunikasi, sistem manajemen kuis menggunakan *message broker ESB (enterprise service bus)* sesuai dengan rancangan arsitektur pada Gambar 3.6.
- c. *Autonomy*, bahwa *services* mempunyai kendali berdasarkan logika yang dienkapsulasi. Sebagai contoh yang telah diterapkan pada sistem dengan masing-masing *service* telah dirancang untuk memiliki logika tersendiri untuk memproses suatu task. Seperti pada *service* kuis yang mempunyai salah satu fungsionalitas mengerjakan kuis. Fungsionalitas tersebut dapat berjalan sendiri tanpa perlu membutuhkan fungsionalitas lainnya karena telah memiliki *service* tersendiri. Seperti yang ditunjukkan pada Tabel 3.11 Perancangan detail layanan sistem pengelolaan kuis.
- d. *Abstraction*, di luar apa yang dideskripsikan pada *service contract*, *services* menyembunyikan logika dari dunia luar. Aspek ini masih belum diterapkan dalam perancangan sistem manajemen kuis, karena aspek ini tergantung pada logika pada sisi *back-end* yang belum terimplementasi.
- e. *Reusability*, logika terbagi menjadi *services* dengan tujuan untuk digunakan kembali. Sebagai contoh penerapan dari sistem manajemen kuis, prinsip *reusability* dapat digunakan kembali ketika ada pengembangan fitur baru. Karena setiap *service*, data base dan method telah terpisah masing-masing, hal ini tentunya akan memudahkan dalam pengembangan fitur baru tanpa mengganggu *service*, data base dan method yang lainnya. Sesuai dengan perancangan yang dibuat pada Tabel 3.10, Tabel 3.11 dan Tabel 3.12 dan juga perancangan *back-end* pada Tabel 3.29 logika sudah terbagi menjadi masing-masing *service*.
- f. *Composability*, kumpulan dari *services* dapat dikoordinasikan dan dihimpun untuk membentuk *services* yang berbeda. Prinsip *Composability* hampir mirip dengan prinsip *Reusability* dengan memanfaatkan masing-masing *service* untuk membentuk sistem yang

baru. Seperti pembuatan ranking kelas dengan memanfaatkan *service* kelas dan *service* nilai untuk membentuk *service* ranking.

- g. *Statelessness*, *services* meminimalisasi sifat berpegang teguh pada informasi tertentu untuk sebuah aktivitas. Data yang ada di satu sistem, harus sama dengan data di sistem yang lain. Belum dapat diuji.
- h. *Discoverability*, *services* dirancang dengan sifat yang deskriptif sehingga mereka dapat ditemukan dan ditentukan dengan menggunakan mekanisme-mekanisme penemuan yang ada. Berdasarkan Tabel 3.29 setiap *service* yang ada telah deskriptifkan secara detail agar setiap *service* dapat dengan mudah ditemukan dan dikembangkan lebih lanjut.



BAB V

KESIMPULAN DAN SARAN

5.1 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*. Pengembangan sistem manajemen kuis dilakukan dengan empat tahap sesuai dengan SOA *life-cycle* yaitu *model*, *assemble*, *deploy*, dan *manage*. Sistem manajemen kuis 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 dikembangkan dalam empat tahap yaitu *model*, *assemble*, *deploy*, dan *manage* yang menghasilkan rancangan *activity diagram*, *db diagram*, desain antarmuka, dan *use case diagram*. Hasil dari pengujian *front-end* sistem dapat disimpulkan bahwa sistem manajemen kuis dapat mudah digunakan dan telah sesuai dengan aspek-aspek tampilan pengembangan perangkat lunak. Kemudian pengujian dari *back-end developer* menyimpulkan bahwa *front-end* sistem manajemen kuis dapat dikembangkan lebih lanjut ke sisi *back-end*.

5.2 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.

DAFTAR PUSTAKA

- 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).
- Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet computing*, 6(2), 86-93.
- Erl, T. (2005). *SOA: Principles of Service Design*.
- 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*.
- Google. (n.d.). *Sign in with google | google developers*. Google. Retrieved December 23, 2021, from <https://developers.google.com/identity/gsi/web>
- Hidayat, A. (2011). *Rancang Bangun Sistem Kuis Online Adaptif*. (2011).
- IBM Cloud Education. (n.d.). *SOA*. IBM. Retrieved December 26, 2021, from <https://www.ibm.com/cloud/learn/soa>
- Kapojos, 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. <https://doi.org/10.35793/jtek.1.1.2012.554>
- Morris, S. (n.d.). (2021). Full Stack, Front End, Back End—What Does It All Mean? <https://skillcrush.com/blog/front-end-back-end-full-stack/>
- McBride, G. (2007). *The Role of {SOA} Quality Management in {SOA} Service Lifecycle Management*. April.
- O'brien, James A. (2006). *Introduction to Information Systems*.
- Pakusadewo, N. A., & Slamet Sudaryanto, N. (2015). IMPLEMENTASI SERVICE ORIENTED ARCHITECTURE PADA SISTEM PENGELOLAAN DATA ASURANSI BPJS KARYAWAN PT. TRANSPORTASI JAKARTA. 2015.
- Rivai, D. A. (2012). Pembuatan Website Profil Sekolah Menengah Kejuruan (SMK) Miftahul Huda Ngadirojo. *IJNS-Indonesian Journal on Networking and Security*, 2(3).
- Sutabri, Tata. (2012). *Analisis Sistem Informasi*. Yogyakarta: ANDI.

- Soedjatmiko, S., & Hartanto, R. (2014). Perancangan Basis Data dan Layanan Akses Berbasis Service Oriented Architecture (SOA) Untuk Dinas Kesehatan Kabupaten Sleman.
- 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.
- Siregar, A. F. (2018). Pengaruh Model Pembelajaran Tipe Quiz Team Terhadap Hasil Belajar Siswa Kelas X Akuntansi SMK YWKA Medan TP 2017/2018 (Doctoral dissertation).
- Tarigan S, I. Y. S., Soedjatmiko, S., & Hartanto, R. (2010). Perancangan Basis Data dan Layanan Akses Berbasis Service Oriented Architecture (SOA) Untuk Dinas Kesehatan Kabupaten Sleman. *Jurnal Buana Informatika*, 1(1), 15–28. <https://doi.org/10.24002/jbi.v1i1.293>
- 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).

LAMPIRAN

