

**DPPL-003**

## **DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

**Kelasin Aja**

untuk:

Pelajar untuk Pengajar


Dipersiapkan oleh:

Azhar Baihaqi Nugraha (1301194353)  
Bijak Algifan Putra (1301190461)  
Fendi Irfan Amorokhman (1301191447)  
Zendy Bramantia Alfareza (1301194145)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	<b>Prodi S1- Informatika Universitas Telkom</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<b><i>DPPL-003</i></b>		39
		<b>Revisi</b>	0	<i>Tgl: 5 Juni 2022</i>

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

# Daftar Isi

<b>1. Pendahuluan</b>	<b>5</b>
Tujuan Penulisan Dokumen	6
Lingkup Masalah	6
Definisi dan Istilah	6
Referensi	6
Sistematika Pembahasan	6
<b>Deskripsi Perancangan Global</b>	<b>6</b>
Rancangan Lingkungan Implementasi	7
Deskripsi Arsitektural	7
Deskripsi Komponen	7
<b>Perancangan Rinci</b>	<b>8</b>
Realisasi Use Case	8
Use Case <nama use case 1>	8
Identifikasi Kelas	8
Sequence Diagram	8
Diagram Kelas	8
Perancangan Detil Kelas	8
Kelas <nama kelas>	8
Kelas <nama kelas>	9
Diagram Kelas Keseluruhan	9
Algoritma/Query	9
Diagram Statechart	9
Perancangan Antarmuka	9
Perancangan Representasi Persistensi Kelas	10
<b>Matriks Kerunutan</b>	<b>10</b>
Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar	

# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Deskripsi Perancangan Perangkat Lunak (DPPL) atau disebut juga *Software Requirement Specification* (SRS) untuk KelasinAja. Tujuan penulisan dokumen ini adalah untuk memberikan penjelasan mengenai rancangan perangkat lunak yang berfokus pada pembuatan sistem manajemen pembelajaran *online*. Penjelasan yang akan dimasukkan adalah penjelasan mengenai penjelasan umum terkait spesifikasi dan sebagainya

Dokumen ini digunakan sebagai acuan teknis pengembangan perangkat lunak KelasinAja sekaligus sebagai bahan evaluasi baik proses maupun akhir pengembangannya. Diharapkan dengan adanya dokumen ini, proses pengembangan akan lebih terarah sehingga mendapatkan hasil yang memuaskan dan terhindar dari kekurangan yang lainnya.

## 1.2 Lingkup Masalah

Perangkat lunak yang akan dibangun adalah perangkat lunak berupa sistem manajemen pembelajaran online, perangkat lunak ini adalah perangkat lunak yang digunakan oleh pengajar dan juga pelajar yang nantinya di dalam website ini pengajar dapat mengatur, membuat dan melaksanakan pembelajaran sedangkan pelajar dapat mengikuti atau melaksanakan pembelajaran yang dibuat oleh pengajar mereka masing masing. Dalam website KelasinAja dapat melakukan hal-hal berikut :

1. User Pengajar dapat membuat kelas
2. User Pengajar dapat membuat form exam
3. User Pengajar dapat memberi sebuah materi berupa file maupun tulisan
4. User Pengajar dapat melaksanakan penyampaian materi secara langsung bertatap muka secara online
5. User Pelajar dapat mengikuti kelas
6. User Pelajar dapat mengupload tugas
7. User Pelajar dapat mengunduh materi pembelajaran
8. User Pelajar dapat mengikuti sesi tatap muka online dengan pembelajarannya
9. Sistem Menyediakan fitur notifikasi

## 1.3 Definisi dan Istilah

SKPL	Spesifikasi Kebutuhan Perangkat Lunak dan merupakan spesifikasi dari perangkat lunak yang akan dikembangkan
DPPL	Dokumen Deskripsi Perancangan Perangkat Lunak(DPPL) merupakan dokumen deskripsi dari perancangan perangkat lunak yang akan dikembangkan dan bertujuan untuk memberikan landasan yang diperlukan dalam proses pengkodean aplikasi.
RDMS	adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna.
HTTPS	Singkatan dari Hypertext Transfer Protocol Secure, sebuah ekstensi dari Hypertext Transfer Protocol (HTTP). HTTPS Ini digunakan untuk komunikasi aman melalui jaringan komputer, dan banyak digunakan di Internet.
Laravel	Laravel adalah kerangka kerja aplikasi web berbasis PHP yang sumber terbuka, menggunakan konsep Model-View-Controller. Laravel berada dibawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode.
HTML	Hypertext Markup Language adalah bahasa markah standar untuk dokumen yang dirancang untuk ditampilkan di peramban internet.
PHP	PHP: Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis.
Tailwind	Tailwind CSS merupakan framework yang bersifat utility-first untuk membangun desain antarmuka khusus dengan cepat.
Video Conference	Konferensi video adalah seperangkat teknologi telekomunikasi interaktif yang memungkinkan dua pihak atau lebih di lokasi berbeda dapat berinteraksi melalui pengiriman dua arah audio dan video secara bersamaan.

#### 1.4 Referensi

Dokumen acuan yang dipergunakan dalam penulisan dokumen ini adalah:

- Template dokumen Deskripsi Perancangan Perangkat Lunak ( DPPL )
- Spesifikasi Kebutuhan Perangkat Lunak ( SKPL )

#### 1.5 Sistematika Pembahasan

Dokumen DPPL untuk Perangkat Lunak Pembelajaran Online ini berisi deskripsi kebutuhan pengembangan perangkat lunak secara rinci. Pengorganisasian dokumen dikelompokkan dalam tiga bagian utama, yaitu :

1. Bab 1 Pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah yang ditangani pada perangkat lunak yang akan dibangun, dan deskripsi umum dokumen.

2. Bab 2 Deskripsi Global Perangkat Lunak, berisi penjelasan perangkat lunak yang akan diimplementasikan di lingkungan pengguna secara global. Bagian ini terdiri dari perspektif produk, fungsi produk, karakteristik pengguna, batasan – batasan dan asumsi dan kebergantungan

## 2 Deskripsi Perancangan Global

### 2.1 Rancangan Lingkungan Implementasi

Untuk meningkatkan kinerja sistem ini dibutuhkan jaringan internet yang cepat. Sistem ini akan menggunakan firebase sehingga diharapkan dapat membutuhkan waktu yang sedikit dalam menyelesaikan setiap tahapan proses pada sistem. Meningkatnya kecepatan ini diharapkan dapat terjadi di setiap tahapan proses pada sistem. Sistem ini pula akan dijalankan dengan minimum spesifikasi *device* sebagai berikut:

*Software* pada sisi server yang dibutuhkan oleh KelasInAja adalah:

- Sistem Operasi : *Linux, Windows dan Mac*
- *Scripting Language* : PHP
- RDBMS: MySQL

*Hardware* yang dibutuhkan oleh KelasInAja adalah :

- Diatas Windows , Linux, dan Mac.
- An Intel Pentium 4 processor atau yang lebih baru yang mana mendukung SSE3.

### 2.2 Deskripsi Arsitektural

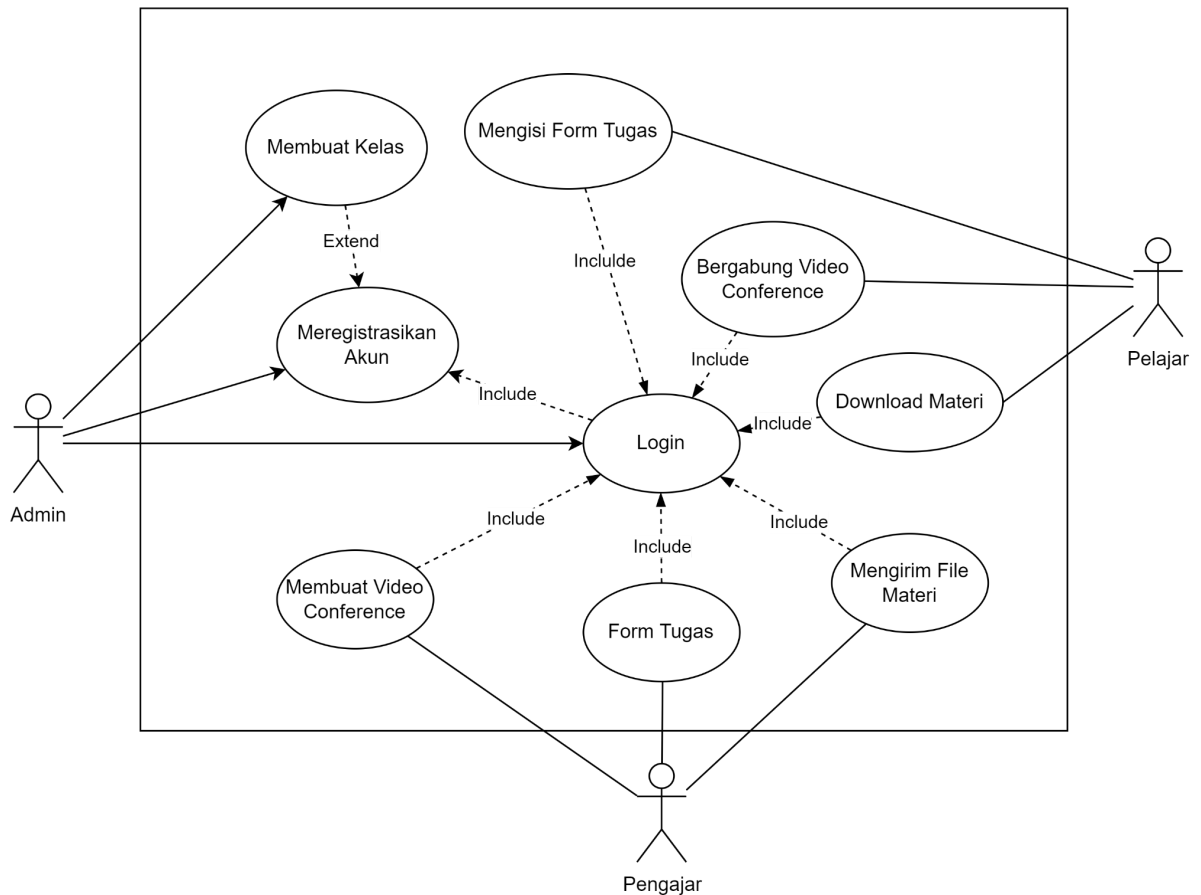
Berikan penjelasan singkat tentang arsitektur /L yang akan dibangun. Gambarkan dalam bentuk diagram komponen.

### 2.3 Deskripsi Komponen

No	Nama Komponen	Keterangan
1	Pengajar ( <i>Teacher</i> )	Menyampaikan Materi secara langsung dan tidak langsung dan memberikan tugas atau ujian
2	Pelajar ( <i>Student</i> )	Mengikuti pembelajaran yang disediakan oleh pengajar, mengerjakan tugas atau ujian dan mengikuti pertemuan tatap muka

### 3 Perancangan Rinci

#### 3.1 Realisasi Use Case



##### 3.1.1 Use Case Admin Meregistrasikan Akun

###### Use Case Scenario :

Precondition: Admin belum meregistrasikan akun.

Postcondition : Admin pengguna berhasil membuat akun pelajar/pengajar.

Primary Flow:

1. Pengguna masuk ke halaman website dalam keadaan belum login dan berada di halaman login.
2. Pengguna mengklik tombol registrasi.
3. Sistem akan memberikan form untuk memasukan nama, memasukan email, dan memilih kelas.
4. Pengguna memasukan nama, email dan pilih kelas.
5. Pengguna mengklik submit.
6. Sistem akan menyimpan seluruh atribut yang dimasukan oleh user kedalam database
7. Pengguna tinggal menunggu konfirmasi dari admin
8. Bila sudah dikonfirmasi maka akun sudah dibuat

Alternate Flow:

1. Pada Primary Flow langkah ke 3, ketika pengguna gagal/tidak memasukan nama, email dan kelas maka akan diminta kembali untuk melakukan langkah ke 3.
2. Pada Primary Flow langkah ke 6, sistem gagal menyimpan data ke database maka pengguna akan diarahkan ke login page dan sistem akan memberikan pemberitahuan kepada pengguna

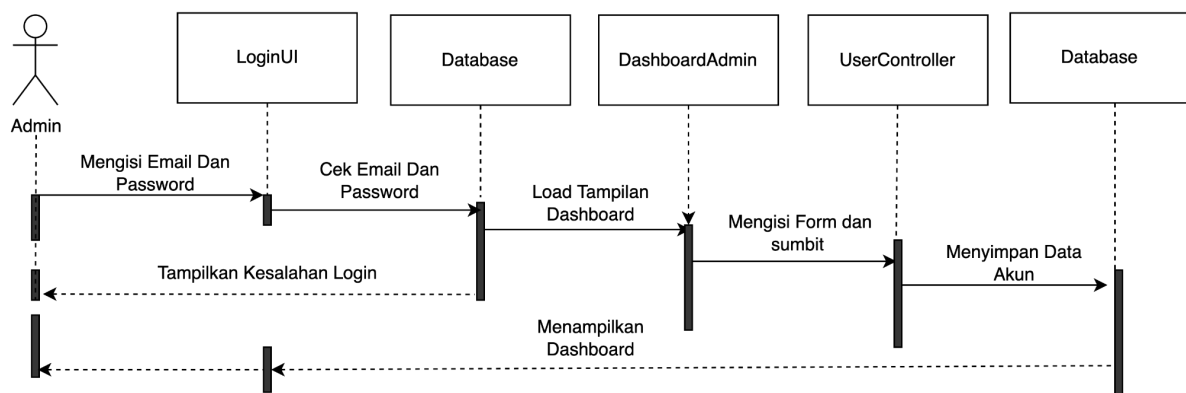


bahwa proses gagal.

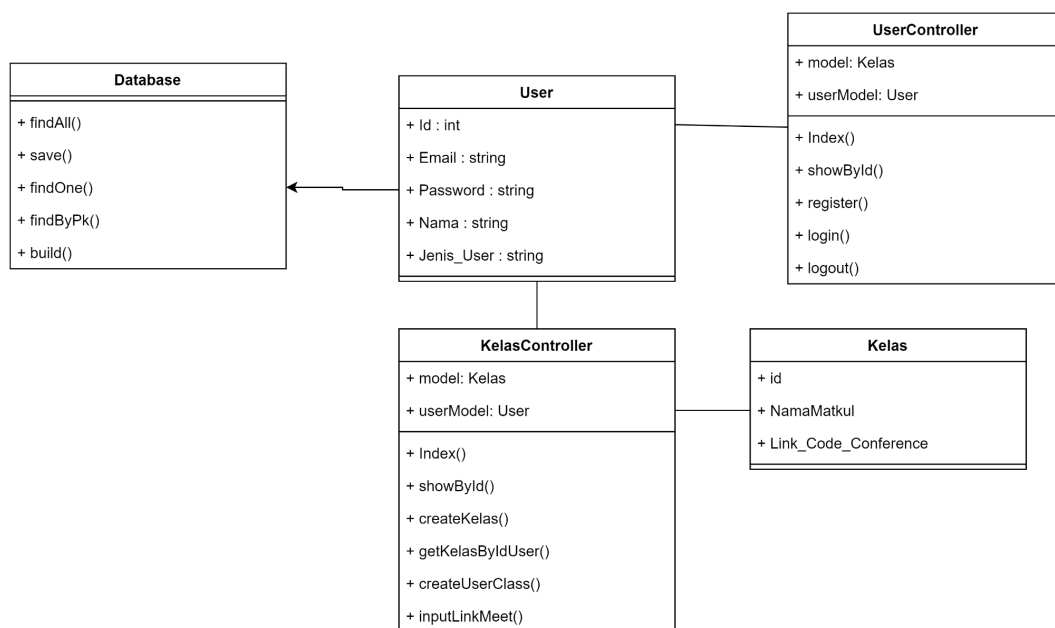
### 3.1.1.1 Identifikasi Kelas

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	Kelas	Entity
3	UserController	Controller
4	KelasController	Controller

### 3.1.1.2 Sequence Diagram




### 3.1.1.3 Diagram Kelas



### 3.1.1.4 Rancangan Antar Muka

**Kelasin**

 Fendi Irfan

Pages

Pendaftaran Akun

Pendaftaran Kelas

Keluar

### Mendaftarkan Akun

Nama

Password

Email

Pilih Kelas

+ Tambah Pilihan Kelas

☒ Pengajar

☐ Pelajar

SUBMIT

[Privacy Policy](#) [Terms of Use](#)

© 2022 Kelasin, Made with ♥

<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>pendaftaran akun</i>	<i>Button</i>	<i>Pendaftaran Akun</i>	<i>Button yang jika ditekan akan mengarahkan user ke halaman pendaftaran Akun</i>
<i>pendaftaran kelas</i>	<i>Button</i>	<i>Pendaftaran Kelas</i>	<i>Button yang jika ditekan akan mengarahkan user ke halaman pendaftaran Kelas</i>
<i>keluar</i>	<i>Button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>nama</i>	<i>Text Field</i>	<i>Nama</i>	<i>Text field yang digunakan untuk memasukkan nama pengguna.</i>
<i>password</i>	<i>Text Field</i>	<i>Password</i>	<i>Text field yang digunakan untuk memasukkan password pengguna.</i>
<i>email</i>	<i>Text Field</i>	<i>Email</i>	<i>Text field yang digunakan untuk memasukkan email pengguna.</i>
<i>kelas</i>	<i>Text Field</i>	<i>Kelas</i>	<i>Text field yang digunakan untuk memasukkan kelas pengguna.</i>
<i>jenis_user</i>	<i>Radio Button</i>	<i>Radio button jenis pengguna</i>	<i>Radio Button yang digunakan untuk memilih apakah pengguna yang akan didaftarkan adalah pelajar atau pengajar</i>
<i>submit</i>	<i>Button</i>	<i>Submit</i>	<i>Button yang akan mendaftarkan akun apabila berhasil melewati proses validasi</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.1.2 Use Case Actor melakukan login

#### Use Case Scenario :

Precondition: mengakses platform atau website

Postcondition : Actor telah masuk kedalam akun mereka

Primary Flow:

1. Actor masuk ke dalam website dan dibawa ke halaman login page
2. Actor menekan tombol login, dan sistem melanjutkan ke halaman yang berisi form untuk memasukkan email dan password
3. Actor memasukkan email dan password dan menekan tombol
4. Sistem akan melakukan pengecekan email dan password ke database sistem
5. Actor akan dibawa ke halaman dashboard sesuai dengan rolenya

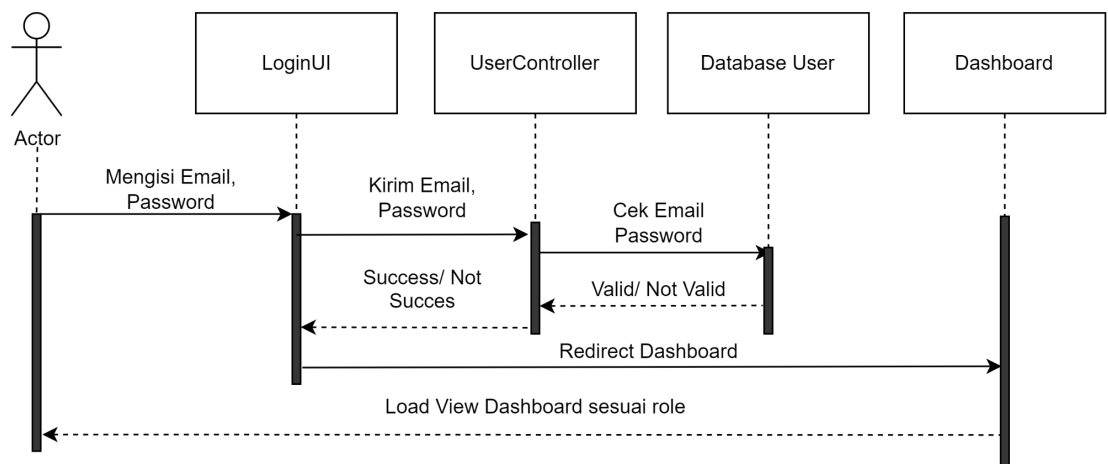
Alternate Flow:

1. Pada langkah 4, apabila disaat pengecekan ternyata tidak ditemukan. maka sistem akan memberikan informasi bila email dan password salah. dan akan diminta kembali pada langkah 3

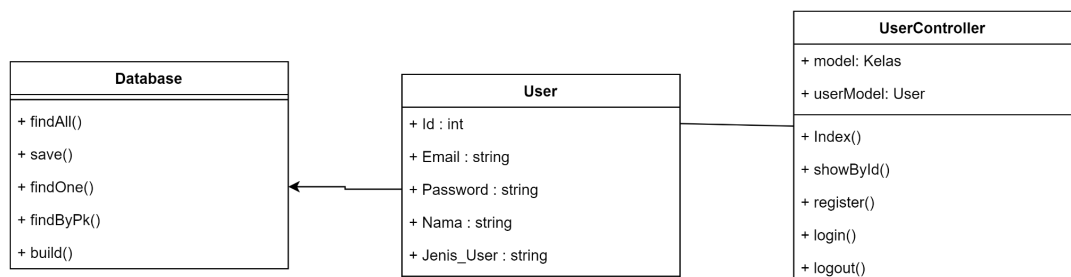
### 3.1.2.1 Identifikasi Kelas

No	Nama Kelas Perancangan	Tipe Kelas
1	User	Entity
2	UserController	Controller

### 3.1.2.2 Sequence Diagram



### 3.1.2.3 Diagram Kelas



### 3.1.2.4 Rancangan Antar Muka

KelasinAja

Login

## Mulai Pembelajaranmu

Email

Password

MASUK

<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>email</i>	<i>TextField</i>	<i>Email</i>	<i>Text field yang digunakan untuk memasukkan email pengguna.</i>
<i>password</i>	<i>TextField</i>	<i>Password</i>	<i>Text field yang digunakan untuk memasukkan password pengguna.</i>
<i>submit</i>	<i>button</i>	<i>MASUK</i>	<i>Button yang jika ditekan akan mengarahkan user masuk ke dalam aplikasi jika email dan password valid.</i>

### 3.1.3 Use Case Mendaftarkan Kelas

#### Use Case Scenario :

Precondition : Pengguna telah berada di halaman membuat kelas dan telah login.

Postcondition : Kelas telah berhasil didaftarkan dan disimpan di database.

Primary Flow :

1. Pengguna mengisi data nama kelas
2. Pengguna mengklik submit.
3. Sistem menyimpan data kelas.

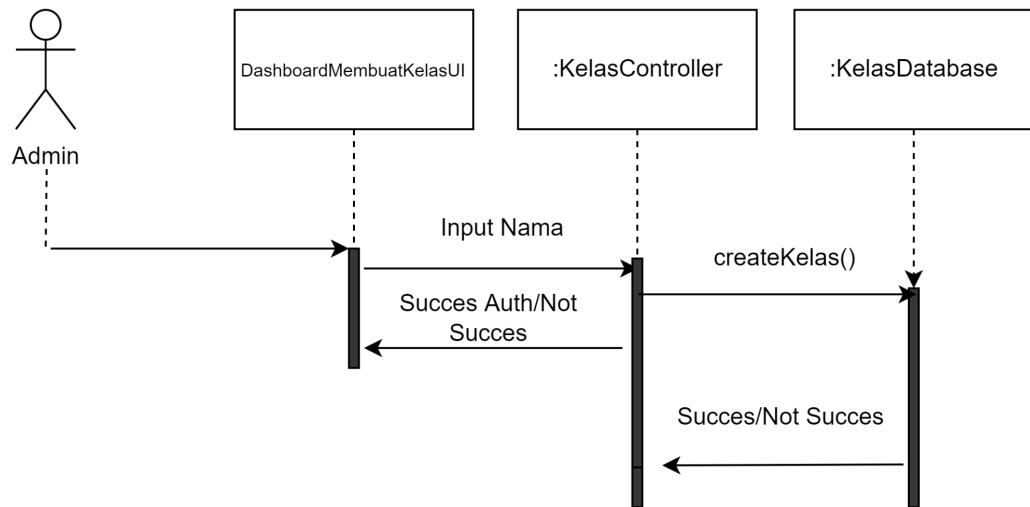
Alternate Flow :

1. Pada tahap 3, sistem gagal menyimpan data, Sistem akan menampilkan notifikasi bahwa pembuatan kelas gagal.

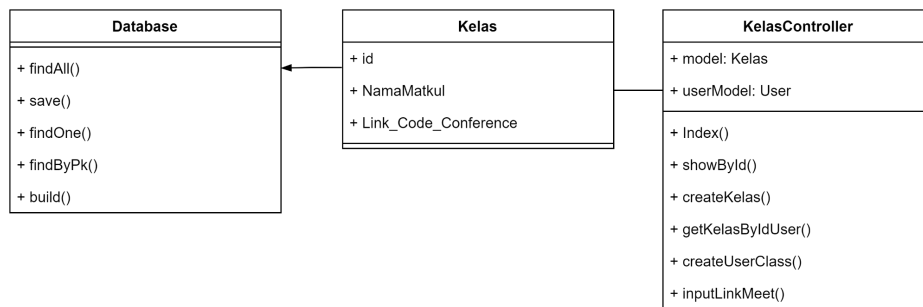
#### 3.1.3.1 Identifikasi Kelas

<i>No</i>	<i>Nama Kelas Perancangan</i>	<i>Tipe Kelas</i>
1	<i>Kelas</i>	<i>Entity</i>

### 3.1.3.2 Sequence Diagram




### 3.1.3.3 Diagram Kelas



### 3.1.3.4 Rancangan Antar Muka

Kelasin

 Fendi Irfan

Pages

Pendaftaran Akun

Pendaftaran Kelas

Keluar

Mendaftarkan Kelas

Nama Kelas

SUBMIT

Privacy Policy

Terms of Use

© 2022 Kelasin, Made with ♥

<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>pendaftaran akun</i>	<i>Button</i>	<i>Pendaftaran Akun</i>	<i>Button yang jika ditekan akan mengarahkan user ke halaman pendaftaran Akun</i>
<i>pendaftaran kelas</i>	<i>Button</i>	<i>Pendaftaran Kelas</i>	<i>Button yang jika ditekan akan mengarahkan user ke halaman pendaftaran Kelas</i>
<i>keluar</i>	<i>Button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>nama_kelas</i>	<i>Text Field</i>	<i>Nama Kelas</i>	<i>Text field yang digunakan untuk memasukkan nama Kelas.</i>
<i>submit</i>	<i>Button</i>	<i>Submit</i>	<i>Button yang akan menambahkan kelas.</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.1.4 Use Case Membuat Video Conference

Precondition : Pengajar sudah login dan belum ada video conference yang sudah dibuat.

Postcondition : Pengajar berhasil membuat video conference

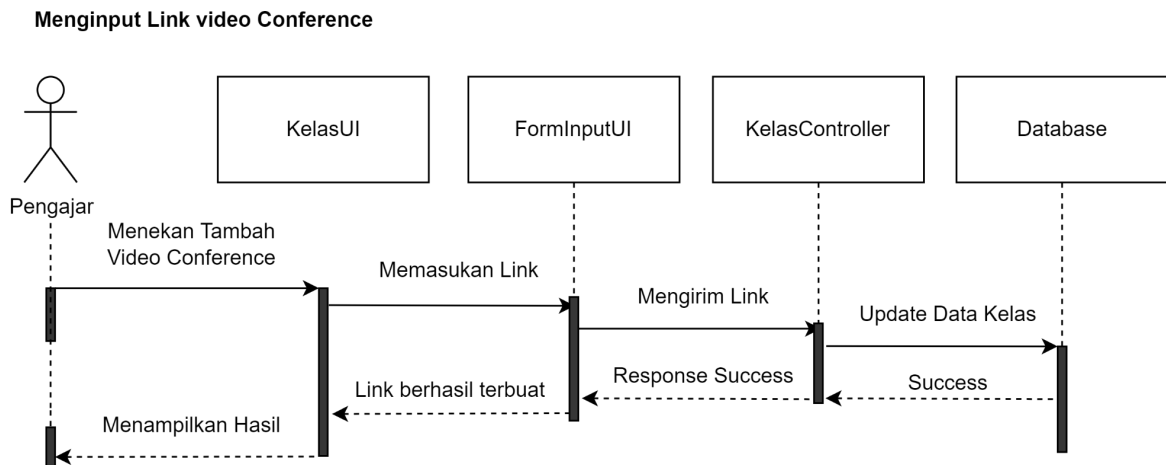
Primary Flow :

1. Pengajar sudah berada di menu kelas yang ingin dibuatkan video conference
2. Pengajar mengklik tombol “Meet”
3. Pengajar memilih opsi “instant meeting” atau “Schedule meeting”
4. Pengajar masuk ke menu setup audio/video device
5. Pengajar berhasil membuat video conference

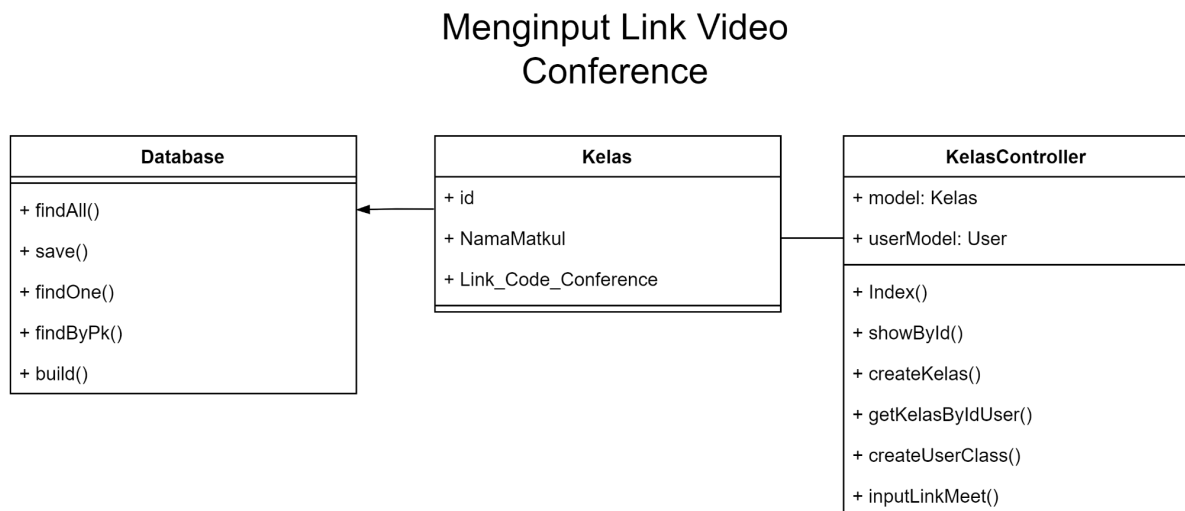
### 3.1.4.1 Identifikasi Kelas

No	Nama Kelas Perancangan	Tipe Kelas
1	Kelas	Entity
2	KelasController	Controller

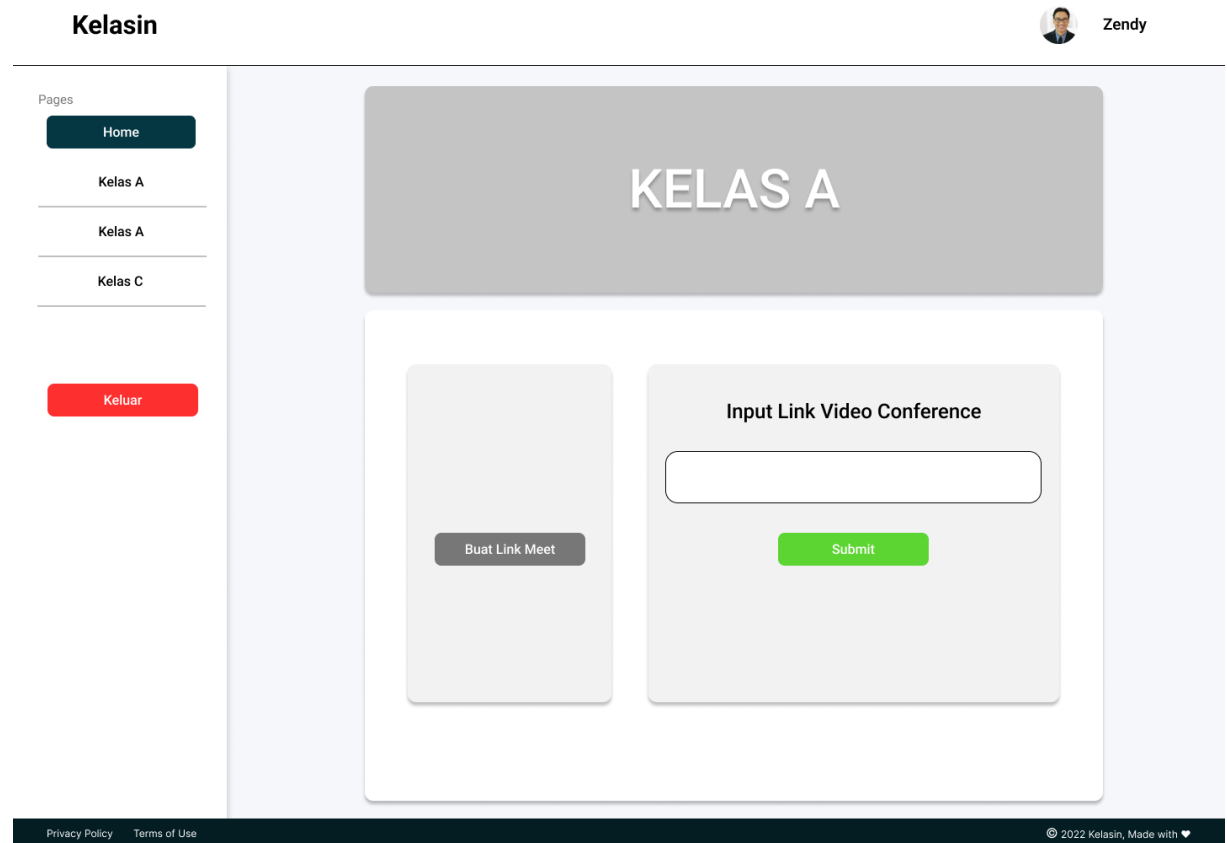
### 3.1.4.2 Sequence Diagram



### 3.1.4.3 Diagram Kelas



### 3.1.4.4 Rancangan Antar Muka



<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>button_kelas</i>	<i>button</i>	<i>kelas A</i>	<i>Button yang jika ditekan akan mengarahkan pengguna ke kelas A</i>
<i>link</i>	<i>TextField</i>	<i>Password</i>	<i>Text field yang digunakan untuk memasukkan password pengguna.</i>
<i>keluar</i>	<i>button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>submit</i>	<i>button</i>	<i>Submit</i>	<i>Button yang akan menambahkan link video conference pada kelas.</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.1.5 Use Case Mengakses Video Conference

#### Use Case Scenario :

Precondition: Pengguna berada di halaman kelas

Postcondition : Pengguna sudah masuk ke video conference

Primary Flow :

1. Pengguna berada di halaman kelas.
2. Pengguna membuka menekan tombol untuk masuk ke video conference.
3. Pengguna akan diarahkan ke halaman video conference

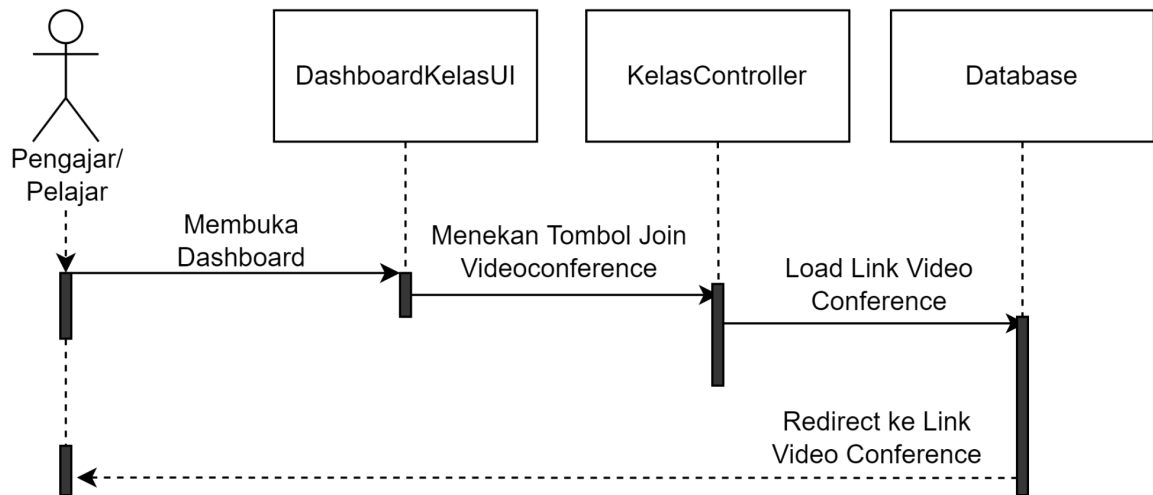
#### 3.1.5.1 Identifikasi Kelas



<i>No</i>	<i>Nama Kelas Perancangan</i>	<i>Tipe Kelas</i>
1	<i>Kelas</i>	<i>Entity</i>
2	<i>KelasController</i>	<i>Controller</i>

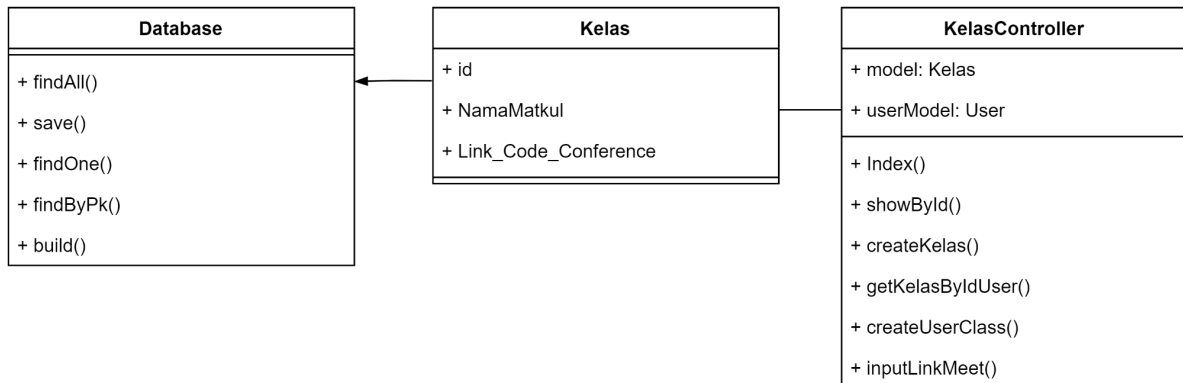
### 3.1.5.2 Sequence Diagram

#### Akses Video Conference

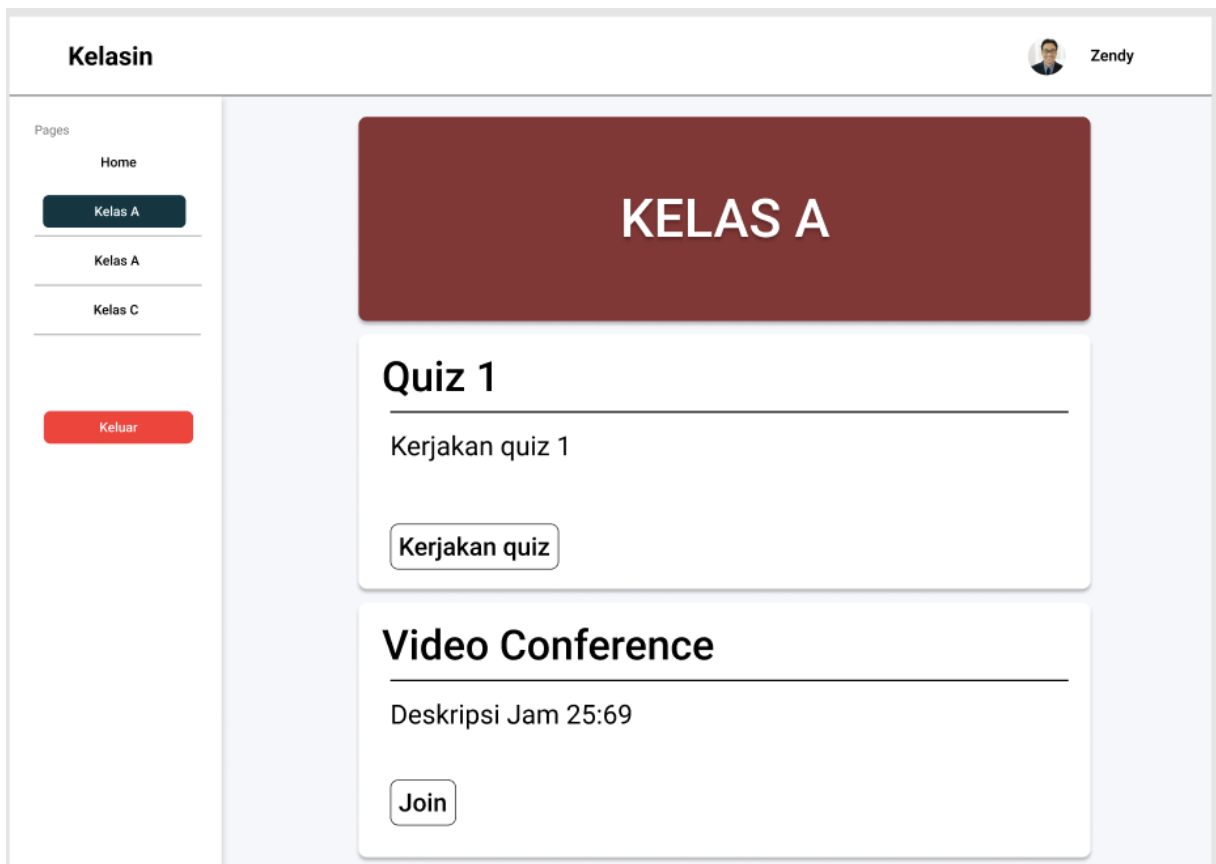


### 3.1.5.3 Diagram Kelas

#### Mengakses Link Video Conference



### 3.1.5.4 Rancangan Antar Muka



Id_Objek	Jenis	Nama	Keterangan
<i>button_kelas</i>	<i>button</i>	<i>kelas A</i>	<i>Button yang jika ditekan akan mengarahkan pengguna ke kelas A</i>
<i>kerjakan_kuis</i>	<i>button</i>	<i>Kerjakan Kuis</i>	<i>Button yang akan mengarahkan pengguna ke link form Exam</i>

<i>join</i>	<i>button</i>	<i>Join</i>	<i>Button yang akan mengarahkan pengguna ke link video conference</i>
<i>keluar</i>	<i>button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.1.6 Use Case Menginput Link Form Exam

#### Use Case Scenario :

Precondition : Pengajar berada di halaman kelas

Postcondition : Tugas sudah terbuat

Primary Flow :

1. Pengajar masuk ke halaman “Buat Form Tugas”.
2. Pengajar input judul form , deskripsi, dan link embed dari google form.
3. Pengajar mengklik submit.
4. Tugas sudah terbuat

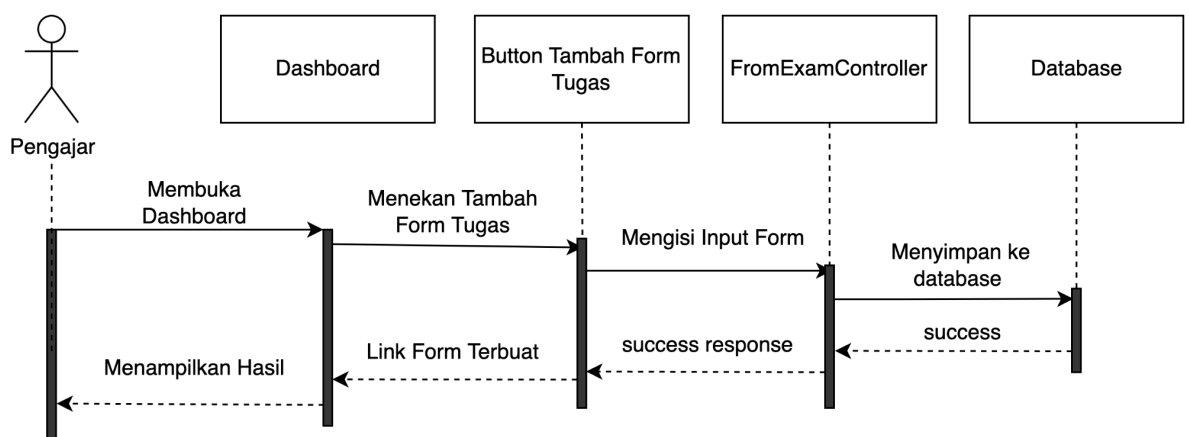
Alternate Flow :

1. Pengajar belum mengisi deskripsi dan embed dari google form saat mengklik submit, sistem akan menampilkan peringatan gagal.

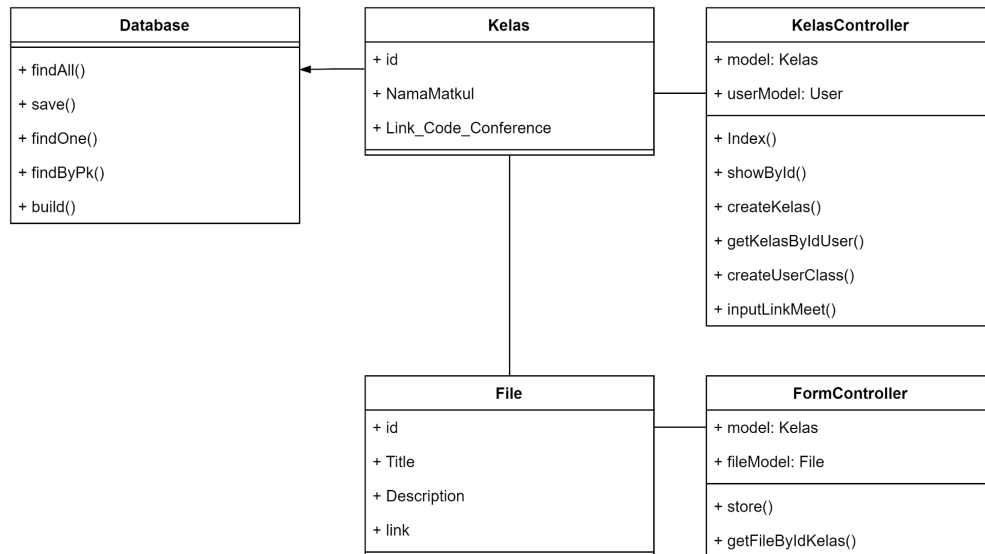
#### 3.1.6.1 Identifikasi Kelas

<i>No</i>	<i>Nama Kelas Perancangan</i>	<i>Tipe Kelas</i>
1	<i>Kelas</i>	<i>Entity</i>
2	<i>File</i>	<i>Entity</i>
3	<i>KelasController</i>	<i>Controller</i>
4	<i>FileController</i>	<i>Controller</i>

#### 3.1.6.2 Sequence Diagram



### 3.1.6.3 Diagram Kelas



### 3.1.6.4 Rancangan Antar Muka

The screenshot shows the 'Kelasin' web application. On the left is a sidebar with a 'Pages' section containing 'Home', 'Kelas A', 'Kelas A', 'Kelas C', and a 'Keluar' button. The main content area has a large grey header with 'KELAS A'. Below it is a 'Buat Form Tugas' section with three input fields: 'Judul Tugas', 'Link Embed Form', and 'Deskripsi Tugas', followed by a 'Submit' button. A 'Buat Link Embed' button is also visible on the left side of the main content area. The footer contains 'Privacy Policy', 'Terms of Use', and '© 2022 Kelasin, Made with ♥'.

Id Objek	Jenis	Nama	Keterangan
<i>button_kelas</i>	<i>button</i>	<i>kelas A</i>	<i>Button yang jika ditekan akan mengarahkan pengguna ke kelas A</i>
<i>judul</i>	<i>TextField</i>	<i>Judul Tugas</i>	<i>Text field yang digunakan untuk memasukkan judul tugas</i>
<i>link</i>	<i>TextField</i>	<i>Link Embed Form</i>	<i>Text Field yang digunakan untuk memasukkan link form</i>
<i>deskripsi</i>	<i>TextField</i>	<i>Deskripsi Tugas</i>	<i>Text Field yang digunakan untuk memasukkan deskripsi tugas</i>
<i>keluar</i>	<i>button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>

<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>
<i>submit</i>	<i>button</i>	<i>Submit</i>	<i>Button yang akan menambahkan form tugas dan mengarahkan pengguna ke halaman home apabila telah berhasil melalui validasi.</i>

### 3.1.7 Use Case Mengakses Link Form Exam

#### Use Case Scenario :

Precondition : Pengguna berada di halaman tugas dan telah login.

Postcondition : Pengguna berhasil mengisi tugas

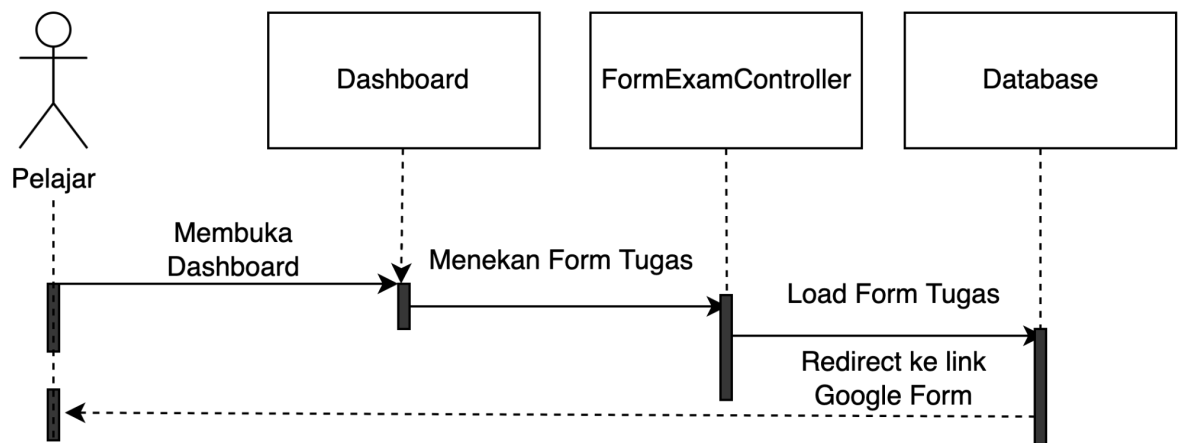
Primary Flow :

1. Pelajar telah berada pada halaman kelas
2. Pelajar menekan tombol pada section tugas.
3. sistem memanggil permintaan pelajar, dan memanggil google form
4. Pelajar dapat mengisi google form

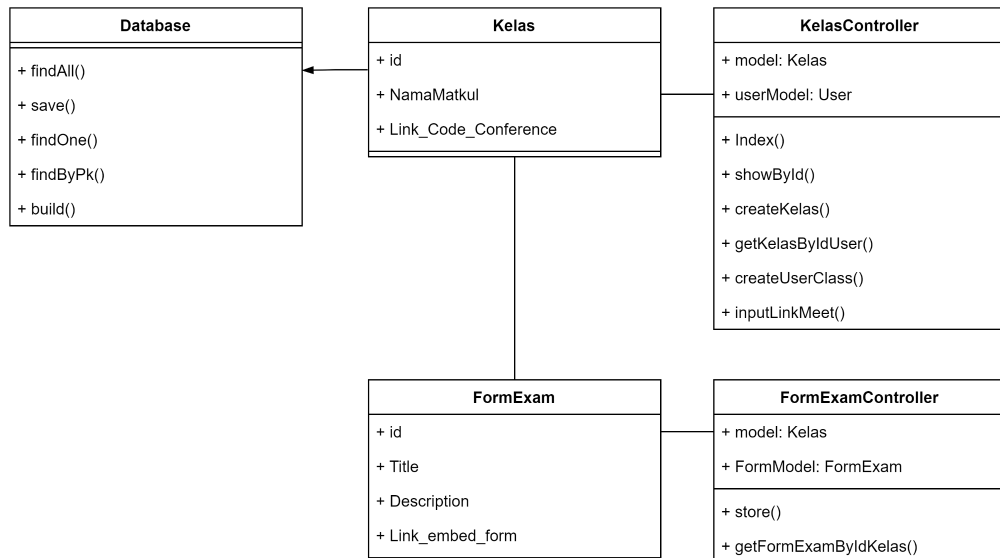
#### 3.1.7.1 Identifikasi Kelas

<i>No</i>	<i>Nama Kelas Perancangan</i>	<i>Tipe Kelas</i>
1	<i>Kelas</i>	<i>Entity</i>
2	<i>Form</i>	<i>Entity</i>
3	<i>KelasController</i>	<i>Controller</i>
4	<i>FormExamController</i>	<i>Controller</i>

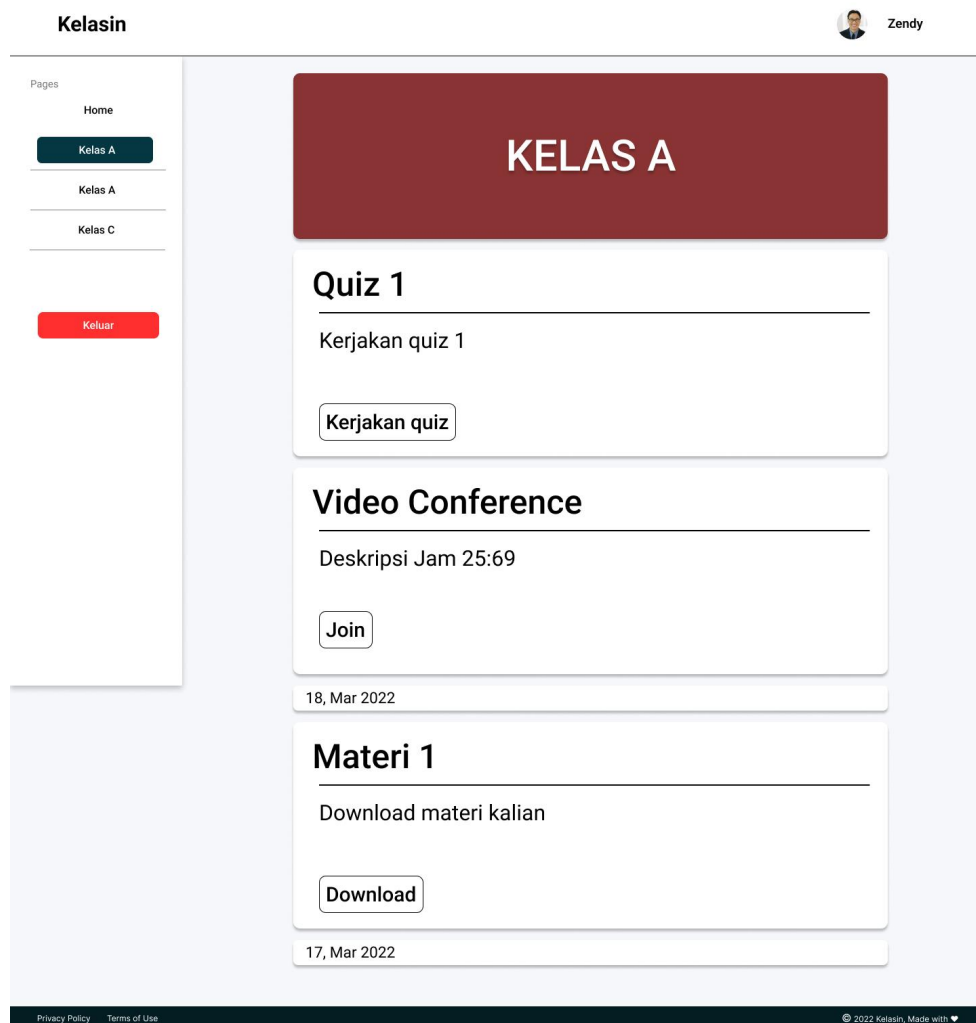
#### 3.1.7.2 Sequence Diagram



### 3.1.7.3 Diagram Kelas



### 3.1.7.4 Rancangan Antar Muka



<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>button_kelas</i>	<i>button</i>	<i>kelas A</i>	<i>Button yang jika ditekan akan mengarahkan pengguna ke kelas A</i>
<i>kerjakan_kuis</i>	<i>button</i>	<i>Kerjakan Kuis</i>	<i>Button yang akan mengarahkan pengguna ke link form Exam</i>
<i>join</i>	<i>button</i>	<i>Join</i>	<i>Button yang akan mengarahkan pengguna ke link video conference</i>
<i>download</i>	<i>button</i>	<i>Download</i>	<i>Button yang akan mengarahkan pengguna ke link file tugas</i>
<i>keluar</i>	<i>button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.1.8 Use Case Membuat Link Materi

#### Use Case Scenario :

Precondition : Pengguna berada di halaman Kelas, pengguna telah menyiapkan file yang akan ditambahkan.

Postcondition : File materi telah diunggah dan bisa diakses oleh seluruh anggota kelas.

Primary Flow :

1. Pengguna menekan tombol “Upload Materi”.
2. Pengguna memasukkan judul link, deskripsi, dan link file.
3. Pengguna menekan tombol submit
4. Sistem menyimpan data materi yang telah diunggah ke database.

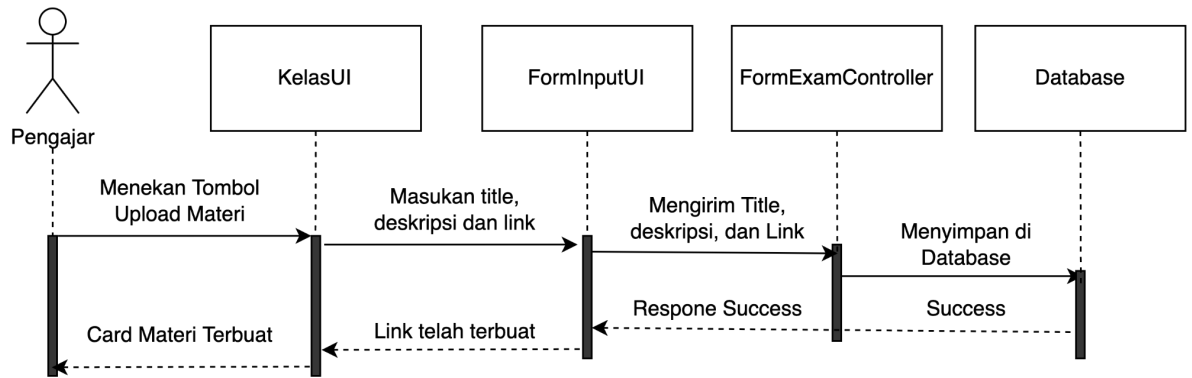
Alternate Flow :

1. pada primary flow langkah ke 3, sistem akan memberikan peringatan “harap masukkan data materi” apabila pengguna belum input data yang diperlukan.

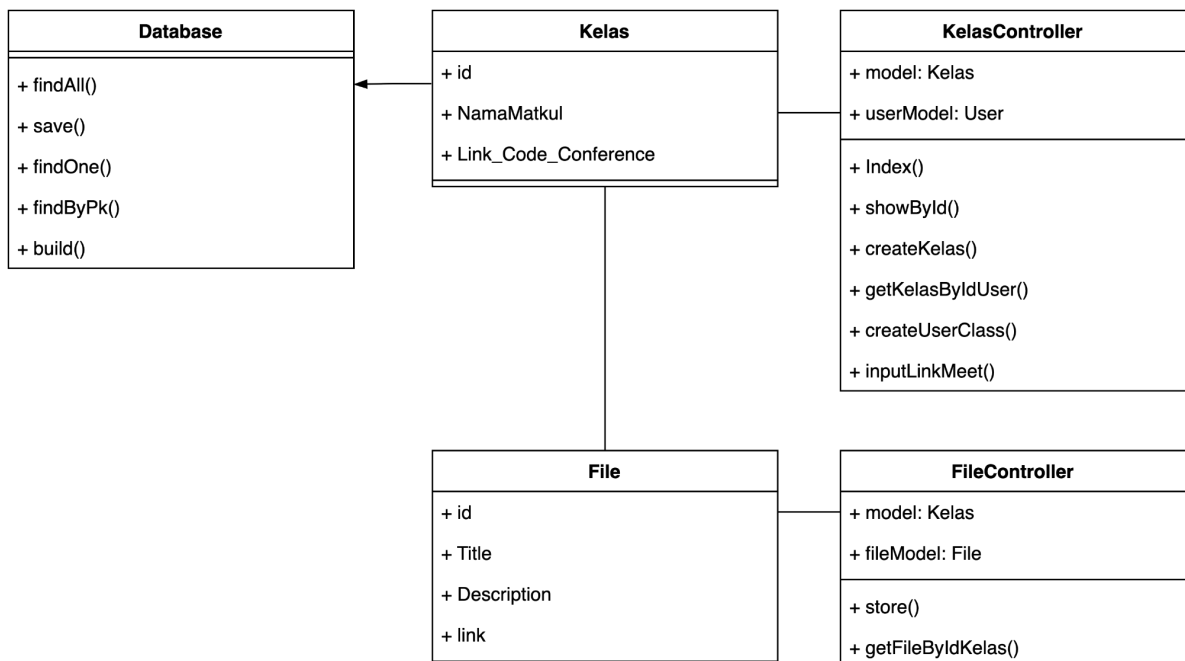
#### 3.1.8.1 Identifikasi Kelas

<i>No</i>	<i>Nama Kelas Perancangan</i>	<i>Tipe Kelas</i>
1	<i>Kelas</i>	<i>Entity</i>
2	<i>Database</i>	<i>Database</i>
3	<i>FormExamController</i>	<i>Controller</i>

### 3.1.8.2 Sequence Diagram



### 3.1.8.3 Diagram Kelas





### 3.1.8.4 Rancangan Antar Muka

<b>Id Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
<i>button_kelas</i>	<i>button</i>	<i>kelas A</i>	<i>Button yang jika ditekan akan mengarahkan pengguna ke kelas A</i>
<i>judul</i>	<i>TextField</i>	<i>Judul Link</i>	<i>Text field yang digunakan untuk memasukkan judul file materi</i>
<i>link</i>	<i>TextField</i>	<i>Upload link materi</i>	<i>Text Field yang digunakan untuk memasukkan link file</i>
<i>deskripsi</i>	<i>TextField</i>	<i>Deskripsi Link</i>	<i>Text Field yang digunakan untuk memasukkan deskripsi file</i>
<i>keluar</i>	<i>button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>submit</i>	<i>button</i>	<i>Submit</i>	<i>Button yang akan menambahkan form tugas dan mengarahkan pengguna ke halaman home apabila telah berhasil melalui validasi.</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.1.9 Use Case Mengakses Link Materi

#### Use Case Scenario :

Precondition : Pengguna telah berada di halaman kelas, Materi telah dikirim oleh Pengajar.

Postcondition : Pengguna diarahkan ke link materi.

Primary Flow :

1. Pengguna berada di halaman kelas
2. Pengguna menekan materi yang dibagikan oleh pengajar.
3. Pengguna menekan tombol download materi.

4. Pengguna diarahkan ke link materi.

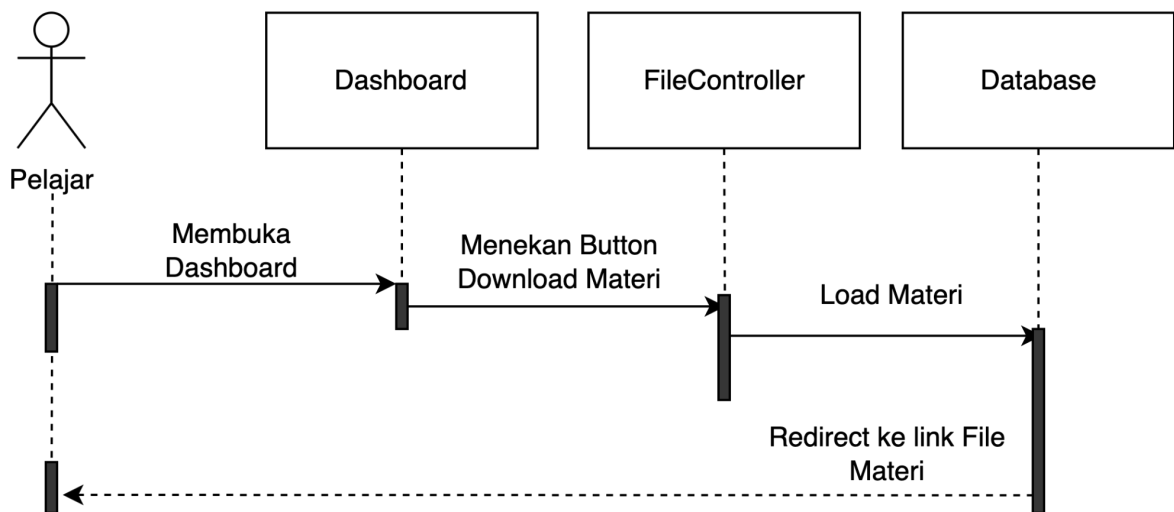
Alternate Flow :

1. Sistem akan menampilkan peringatan “file gagal diunduh” apabila unduh file tidak berhasil.

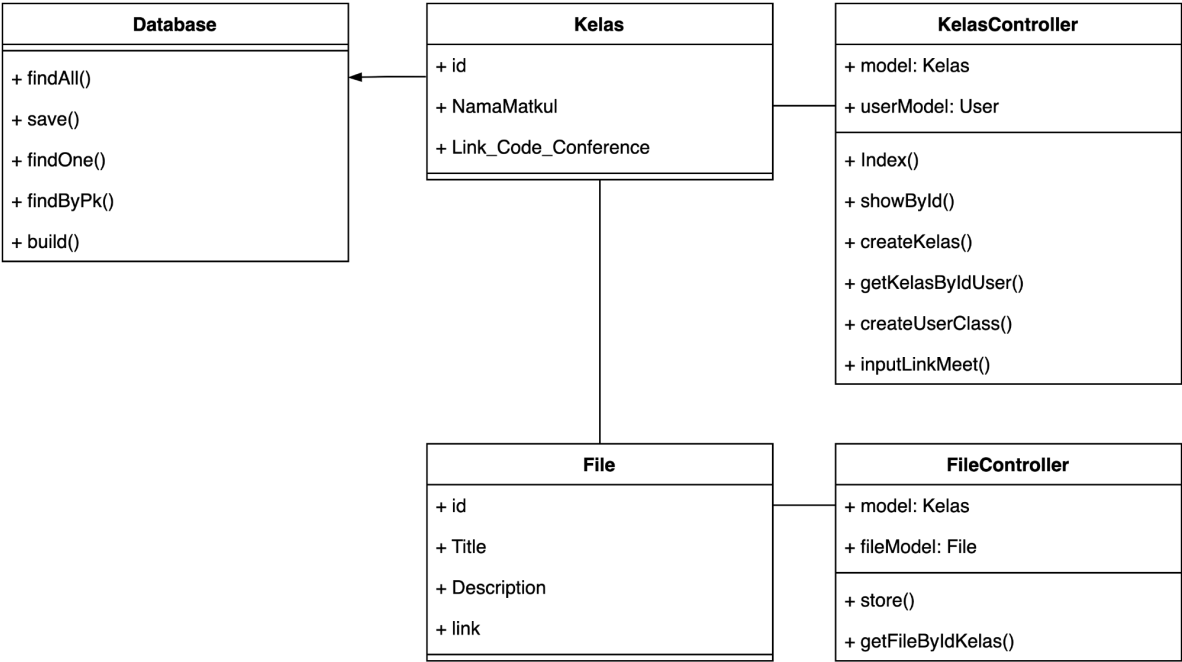
### 3.1.9.1 Identifikasi Kelas

No	Nama Kelas Perancangan	Tipe Kelas
1	Kelas	Entity
2	Database	Database
3	FileController	Controller

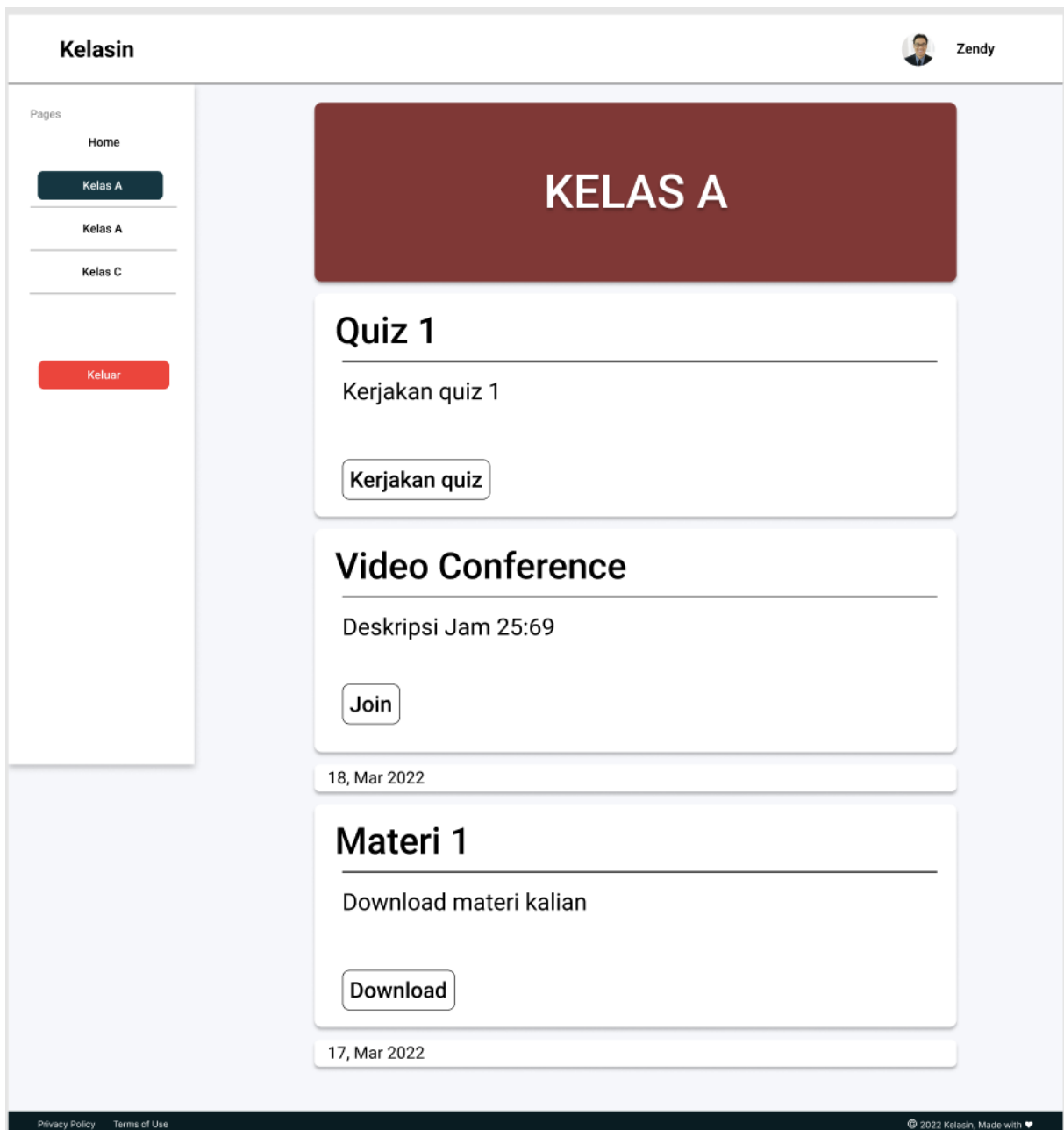
### 3.1.9.2 Sequence Diagram



3.1.9.3 Diagram Kelas



### 3.1.9.4 Rancangan Antar Muka



Id Objek	Jenis	Nama	Keterangan
<i>button_kelas</i>	<i>button</i>	<i>kelas A</i>	<i>Button yang jika ditekan akan mengarahkan pengguna ke kelas A</i>
<i>kerjakan_kuis</i>	<i>button</i>	<i>Kerjakan Kuis</i>	<i>Button yang akan mengarahkan pengguna ke link form Exam</i>
<i>join</i>	<i>button</i>	<i>Join</i>	<i>Button yang akan mengarahkan pengguna ke link video conference</i>
<i>download</i>	<i>button</i>	<i>Download</i>	<i>Button yang akan mengarahkan pengguna ke link file tugas</i>
<i>keluar</i>	<i>button</i>	<i>Keluar</i>	<i>Button yang jika ditekan akan mengeluarkan pengguna dan menuju halaman login</i>
<i>home</i>	<i>Button</i>	<i>KelasinAja</i>	<i>Button yang akan mengarahkan pengguna ke halaman home</i>

### 3.2 Perancangan Detail Kelas

Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait	status
1	User	User	Done
2	KelasController	KelasController	Done
3	Database	Database	Done
4	Kelas	Kelas	Done
5	UserKontroller	UserController	Done
6	UserKelas	UserKelas	Done
7	File	File	Done
8	FileKontroller	FileController	Done
9	FormExam	FormExam	
10	FormExamController	FormExamController	

Untuk setiap kelas:

- identifikasi operasi (mengacu pada tanggung-jawab kelas), termasuk visibility-nya
- identifikasi atribut, termasuk visibility-nya

#### 3.2.1 Kelas User

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : Pelajar

Nama Operasi	Visibility (private, public)	Keterangan
<b>Pelajar</b>		
Mengakses Conference	public	Method untuk join meeting
Mengisi_Form_Tugas	public	Method untuk mengisi form tugas
<b>Pengajar</b>		
Membuat_Video_Conference	public	Method untuk membuat meet conference
Membuat Form Tugas	public	Method untuk membuat form tugas
Upload Materi	public	Method untuk upload file materi
<b>Admin</b>		
Meregistrasi_Akun()	public	Method untuk meregistrasikan akun pengajar dan pelajar
Membuat_Kelas()	public	Method untuk membuat kelas baru ke database
Nama Atribut	Visibility (private, public)	Tipe
Email	private	string
Password	private	string
Nama	private	string
Jenis_user	private	string

#### 3.2.2 Kelas UserController

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : UserController

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
showById()	public	Method untuk mendapatkan user berdasarkan id yang dicari
register()	public	Method untuk mendaftarkan user beserta pemilihan jenis user
login()	public	Method untuk melakukan Authentication dan juga mendeteksi jenis_user dari akun tersebut
logout()	public	Method untuk mengeluarkan akun tersebut dari aplikasi dan juga untuk menghapus token yang sudah di daftarkan
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Model	private	Object

### 3.2.3 Kelas Database

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : Database

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
save()	public	Menyimpan Entitas
findall()	public	Method untuk mencari semua entitas yang ada di database
findOne()	public	Method untuk mencari satu entitas
findWhere()	public	Method untuk mencari entitas yang memenuhi kriteria
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
entitas	private	Object

### 3.2.4 Kelas Kelas

Nama Kelas : Kelas

<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
id	private	int
NamaMatkul	private	string
Link Code Conference	private	string

### 3.2.5 Kelas UserKelas

Nama Kelas : UserKelas

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
user_id	private	int

class id	private	int
----------	---------	-----

### 3.2.6 Kelas KelasController

Nama Kelas : KelasController

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
showbyid()	public	Method untuk mengembalikan kelas berdasarkan id kelas
createKelas()	public	Method untuk mendaftarkan kelas baru
getKelasByIdUser()	public	Method untuk mendaftarkan kelas berdasarkan user yang sudah memasuki kelas tersebut
createUserKelas()	public	Method untuk mendaftarkan siswa masuk kedalam kelas
inputLinkMeet	public	Method untuk mendaftarkan link meet ke kelas
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
id	private	int
User()	private	Object
namaKelas	private	string
Link_code_conference	public	string

### 3.2.7 Kelas File

Nama Kelas : File

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
Download_File()	public	Method untuk download file dari database
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
class id	private	int
File Name	private	string
Deskripsi Materi	private	string
link	private	string

### 3.2.8 Kelas FileController

Nama Kelas : FileController

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
createFile()	public	Method untuk menambahkan file ke kelas
getFileByIdKelas()	public	Method untuk mendapatkan file berdasarkan id kelas
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
class_id	private	int

File Name	private	string
Deskripsi Materi	private	string
link	private	string

### 3.2.9 Kelas FormExam

Nama Kelas : FormExam

<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
class_id	private	int
title	private	string
Deskripsi Materi	private	string
link	private	string

### 3.2.10 Kelas FormExamController

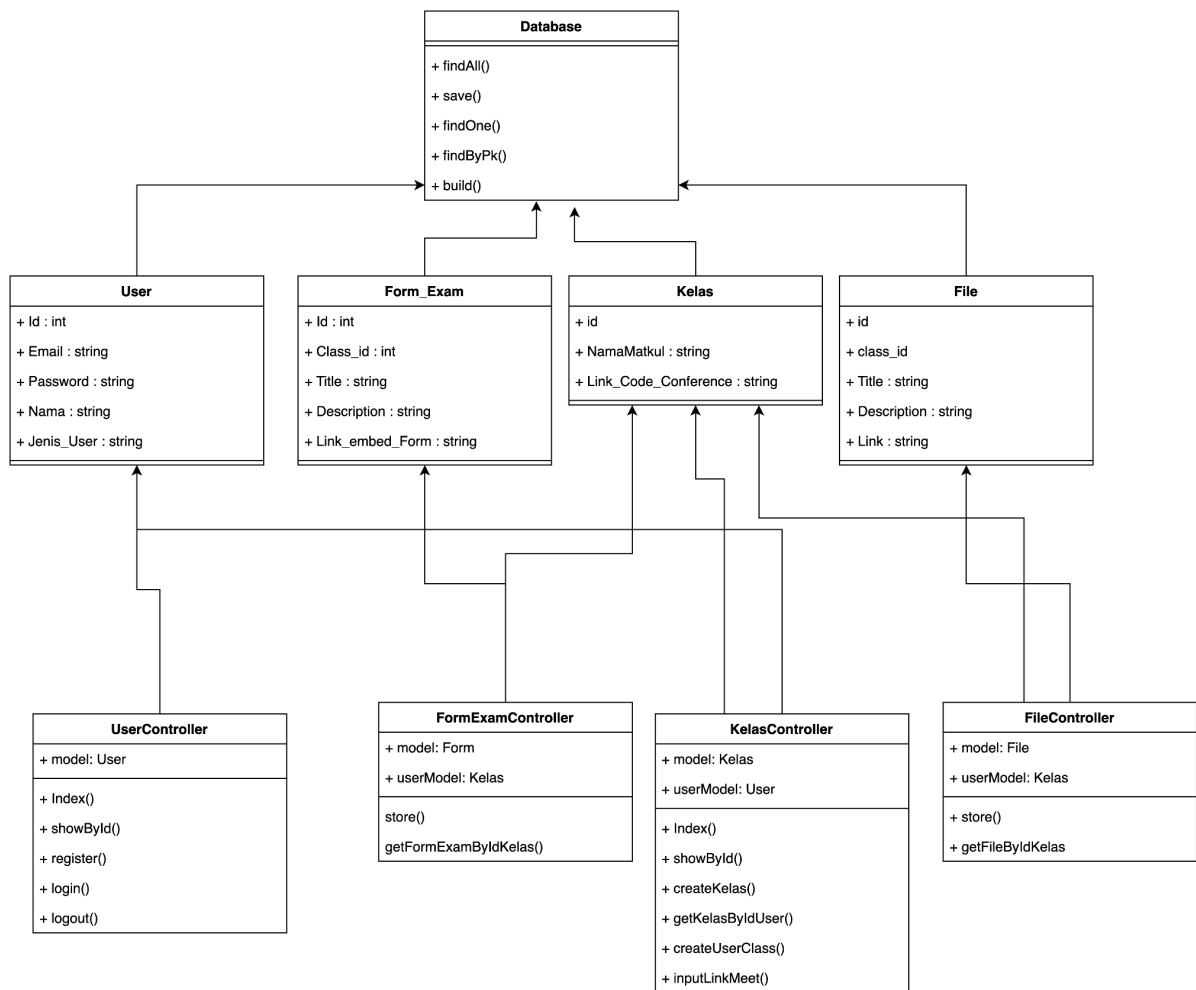
Nama Kelas : FormExamController

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
createFormExam()	public	Method untuk menambahkan FormExam ke kelas
getFormByIdKelas()	public	Method untuk mendapatkan FormExam berdasarkan id kelas
title	private	string
Deskripsi Materi	private	string
link	private	string

## 3.3 Diagram Kelas Keseluruhan

Bagian ini diisi dengan diagram kelas keseluruhan.





### 3.4 Algoritma/Query

#### 3.4.1 Algoritma/Query Login

Nama Kelas : User

Nama Operasi : Login

Algoritma :

```
Input(email: string, password: string)
```

```
X : User
```

```
X = User.findOne(“SELECT * FROM User WHERE ‘email’=email AND  
‘password’=password)
```

```
If X != null {  
    return error;  
}
```

```
return X
```

### 3.4.2 Algoritma Register Akun

Nama Kelas : User, UserKelas

Nama Operasi : CreateUserKelas

Algoritma :

```
Input(email: string, password: string, nama: string, role: string, id_kelas: int)
```

```
X : User
```

```
Y : UserKelas
```

```
X = User.build(“INSERT INTO User VALUES (email, password,nama, role)”).save()
```

```
Y = UserKelas.build(“INSERT INTO Userkelas VALUES (id_kelas, X.id)”).save()
```

```
return Y
```

### 3.4.3 Algoritma Kelas

Nama Kelas : Kelas

Nama Operasi : CreateKelas

Algoritma :

```
Input(nama: string)
```

```
X : Kelas
```

```
X = Kelas.build("INSERT INTO Kelas VALUES (nama)").save()
```

```
return X
```

#### **3.4.4 Algoritma Input Link Conference**

Nama Kelas : Kelas

Nama Operasi : inputLinkConference

Algoritma :

```
Input(link_conference: string, id : int)
```

```
X : Kelas
```

```
X = Kelas.update("UPDATE Kelas SET link_code_conference = link_conference WHERE  
Kelas.id = id")
```

```
return X
```

#### **3.4.5 Algoritma Akses Link Conference**

Nama Kelas : Kelas

Nama Operasi : showById

Algoritma :

Input(id : int)

X : Kelas

X = Kelas.findOne(“SELECT link\_code\_conference FROM Kelas WHERE kelas.id = id)

return X

### 3.4.6 Algoritma Input Form Exam

Nama Kelas : FormExam

Nama Operasi : store

Algoritma :

Input(link\_embed\_form:string, description: string, title : string, id : int)

X : FormExam

X = FormExam.build(“INSERT INTO FormExam VALUES (link\_embed\_form, description, title, id)”).save()

return X

### 3.4.7 Algoritma Akses Form Exam

Nama Kelas : FormExam

Nama Operasi : getFormExamByIdKelas

Algoritma :

```
Input(id : int)
```

```
X : FormExam
```

```
X = FormExam.findOne(“SELECT * FROM FormExam WHERE ‘id’=id”)
```

```
return X
```

### 3.4.8 Algoritma Input Link Materi

Nama Kelas : File

Nama Operasi : store

Algoritma :

```
Input(link:string, description: string, title : string, id : int)
```

```
X : File
```

```
X = File.build(“INSERT INTO File VALUES (link, description, title, id)”).save()
```

```
return X
```

### 3.4.9 Algoritma Akses Link Materi

Nama Kelas : File

Nama Operasi : getFileByIdKelas

Algoritma :

Input(id : int)

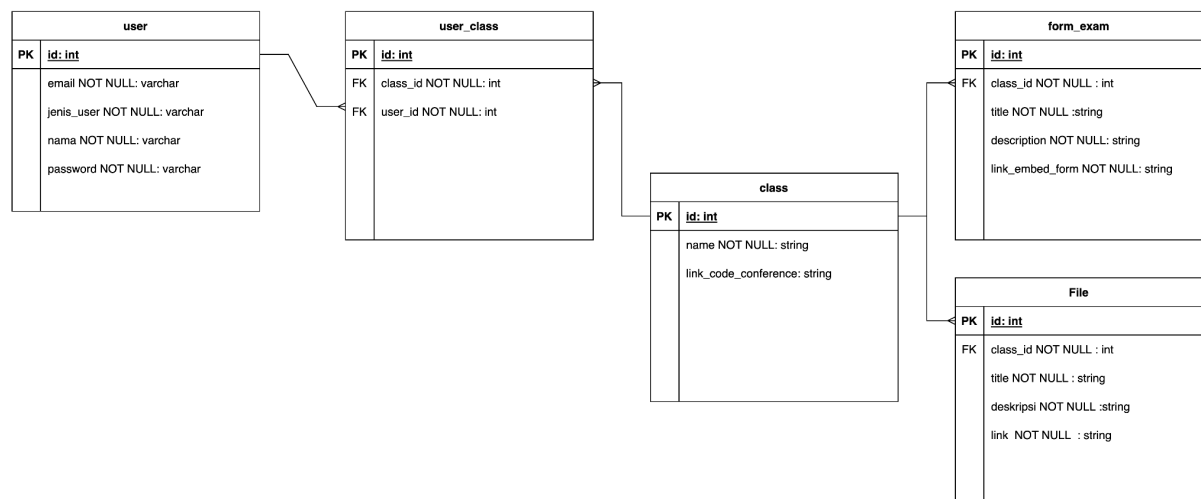
X : File

X = File.findOne(“SELECT \* FROM FormExam WHERE ‘id’=id”)

return X

### 3.5 Perancangan Representasi Persistensi Kelas

Bagian ini diisi dengan rancangan skema basisdata dan traceability-nya terhadap kelas entity. (PEMBUATAN SKEMA RELASI)



## 4 Matriks Kerunutan

Mapping use case dengan kelas-kelas terkait

Requirement	Use Case Terkait
(SKPL-KELASINAJA-001)	1. Membuat Kelas 2. Membuat Link Materi 3. Membuat Link Conference
(SKPL-KELASINAJA-002)	1. Membuat Link Conference
(SKPL-KELASINAJA-003)	1. Membuat Link Exam
(SKPL-KELASINAJA-004)	1. Membuat Link Exam
(SKPL-KELASINAJA-005)	1. Membuat Link Materi
(SKPL-KELASINAJA-006)	1. Membuat Kelas 2. Membuat User

(SKPL-KELASINAJA-007)	1. Mengakses Link Materi
(SKPL-KELASINAJA-008)	1. Mengakses Link Conference
(SKPL-KELASINAJA-009)	1. Mengakses Exam

- *User* Pelajar dapat mengunduh materi pembelajaran
- *User* Pelajar dapat mengikuti sesi tatap muka *online* dengan pembelajarannya (SKPL-KELASINAJA-008)
- Sistem Menyediakan fitur notifikasi (SKPL-KELASINAJA-009)