

# SISTEM INFORMASI KULIAH ONLINE RUANGGURU

## SOFTWARE REQUIREMENT SPECIFICATION

Version 1.1

KELOMPOK B05

Project Manager : Saqib

Anggota :

Mira Pravitasari

Vasko Dagama

Wahyu Retnowati

## Revision History

Date	Version	Description	Author
06/03/2015	1.0		Mira Pravitasari
			Saqib
			Vasko Dagama
			Wahyu Retnowati
20/03/2015	1.1		Mira Pravitasari
		Merapikan format penulisan	Saqib
		Merevisi Class Diagram	Vasko Dagama
		Membuat Use case table dan merevisi use case diagram Merapikan format penulisan dan pengejaan kata	Wahyu Retnowati

## Daftar isi

1. Introduction .....	3
2. Requirement Statement .....	3
3. Analisis Kebutuhan Fungsional.....	5
4. Analisis Kebutuhan Non-Fungsional.....	8
5. Analisis Data.....	9
6. Annex .....	11

# 1. Introduction

Dokumen *Software Requirements Specification* (SRS) merupakan dokumen yang berisi spesifikasi mengenai *requirement* yang akan dikembangkan dalam Sistem Informasi Kuliah Online Ruangguru. Dokumen SRS ini juga berisi tentang informasi mengenai kebutuhan fungsional maupun kebutuhan non-fungsional, *Entity Relationship Diagram* (ERD), *Class Diagram*, *Use Case Diagram* (UCD), asumsi-asumsi, serta *dependencies* dalam pengembangan sistem. Setelah dokumen SRS ini terbentuk, tim pengembang diharapkan terbantu dalam mendapatkan gambaran sistem yang akan dikembangkan dan membantu *project owner* serta *project sponsor* dalam memahami kebutuhan apa saja yang diperlukan dalam pengembangan sistem informasi.

## 1.1. Purpose

Dokumen SRS ini memiliki tujuan untuk dapat memberikan gambaran secara spesifik mengenai *external behavior* dari Sistem Informasi Kuliah Online Ruangguru. Dokumen ini juga akan menjelaskan kebutuhan fungsional dan non-fungsional, *Entity Relationship Diagram* (ERD), *Use Case Diagram* (UCD), asumsi-asumsi dan *dependencies* dalam pengembangan Sistem Informasi Kuliah Online Ruangguru.

## 1.2. Scope

Dokumen ini berisi tentang :

- Dokumen *Software Requirements Specification* berdasarkan spesifikasi *user*.
- *Use Case Diagram* berdasarkan kebutuhan fungsional
- *Entity Relationship Diagram* (ERD)
- *Class Diagram*
- Analisa asumsi dan *dependencies* system

# 2. Requirement Statement

Ruangguru adalah sebuah lembaga les privat yang berskala nasional. Saat ini ruangguru memiliki lebih dari 7000 jumlah pengajar yang tersebar di seluruh Indonesia. Siswa yang mau belajar di Ruangguru dapat mencari guru privat berdasarkan daerah tempat tinggalnya. Sampai saat ini, pembelajaran yang dilaksanakan Ruangguru hanya bersifat *offline*. Guru dan murid melakukan tatap muka secara langsung dalam proses belajar mengajar seperti halnya lembaga les privat lainnya. Hanya dengan menyediakan pembelajaran secara *offline*, masalah umum yang sering terjadi adalah sering tidak adanya kecocokan waktu antara guru dan murid untuk bertemu. Hal ini membuat Ruangguru tidak dapat mencapai tujuannya secara maksimal. Sementara, *website* Ruangguru hanya digunakan untuk pendaftaran dan pembayaran saja. Hal ini membuat Ruangguru merasa perlu untuk memanfaatkan Teknologi Informasi untuk mendukung pelaksanaan pembelajaran yang diberikan Ruangguru dan mengoptimalkan fungsi dari *website* Ruangguru.

Sistem Informasi yang akan dikembangkan guna mengoptimalkan kinerja Ruangguru adalah Sistem Informasi Kelas Online atau disingkat dengan Sistem Informasi Kuliah Online Ruangguru. Sistem Informasi Kuliah Online Ruangguru adalah suatu sistem yang digunakan untuk melakukan kegiatan belajar mengajar antara guru dan murid secara *online*, dimana guru membuat kelas tertentu dan murid dapat mengakses materi di dalam kelas tersebut. Guru-guru yang terdaftar di Ruangguru dapat membuat kelas *online* yang telah disetujui oleh *admin* dari Ruangguru. Siswa dapat belajar dengan mengakses materi-

materi sesuai kelas yang diambil. Sistem ini akan memudahkan siswa Ruangguru untuk belajar kapanpun dan dimanapun asalkan terkoneksi dengan internet.

Sistem ini diharapkan bisa membantu area fungsi bisnis pemasaran dan penjualan (*Marketing and Sales*). Proses bisnis yang dibantu adalah proses pemasaran, penyediaan produk, dan penjualan. Pengguna sistem ini adalah *staff* IT Ruangguru sebagai “*admin*”, para guru yang memiliki akun di ruangguru.com yang juga sudah diverifikasi sebagai “*guru*”, dan para murid yang memiliki akun di ruangguru.com sebagai “*murid*”.

Semua *user* yang memiliki akun ruangguru.com dan telah melakukan *login* ke dalam *website* ruangguru.com dapat mengakses sistem kelas *online*. Karena sistem ini merupakan sebuah subsistem dari ruangguru.com, *username* dan *password* yang digunakan terintegrasi dengan *database* ruangguru.com. Pengguna sistem kelas *online* terbagi ke dalam beberapa *role*, yaitu *Admin*, *Guru*, dan *Murid*. *Admin* memiliki peranan untuk menyetujui pembuatan kelas, mem-*publish* kelas, dan mengelola partisipan kelas. *Guru* memiliki peranan antara lain membuat kelas dan mengunggah materi ke dalam kelas yang dibuatnya. Selain itu, ada pula *role* *Murid* yang memiliki peranan mendaftar kelas serta mengikuti aktivitas kelas.

Informasi *user* yang harus disimpan adalah *username*, *password*, *email*, dan nama. Jika *user* tersebut memiliki *role* *guru*, *user* tersebut dapat membuat kelas. Untuk membuat kelas, *guru* harus mengakses fitur pembuatan kelas dan setelahnya memasukkan informasi terkait kelas. Informasi kelas yang harus disimpan adalah nama kelas, deskripsi kelas, kelas *id*, dan subjek dari kelas tersebut. Setelah kelas dibuat, *guru* harus memasukkan rancangan pembelajaran berupa topik-topik yang akan diajarkan di dalam kelas. Setiap topik nantinya akan memiliki beberapa materi yang berfungsi untuk memberikan pemahaman terhadap topik tersebut. *Guru* harus mengunggah beberapa materi ke dalam kelas tersebut. Terdapat ketentuan minimal terhadap jumlah materi yang harus diunggah ke dalam kelas yaitu sebanyak lima materi sebelum kelas dapat disetujui oleh *admin*. Informasi terkait materi yang harus disimpan adalah judul materi, dan deskripsi atau catatan terkait materi.

Setelah batas *minimum* pengunggahan materi tercapai, *guru* menunggu kelasnya disetujui oleh *admin*. *Admin* akan mengecek kelas-kelas yang telah dibuat oleh *guru* untuk memastikan kelayakan dari materi-materi yang ada di dalam kelas-kelas tersebut. Jika *admin* menganggap bahwa materi-materi yang ada di dalam sebuah kelas layak, *admin* lalu menyetujui pembuatan kelas tersebut. Jika tidak, *admin* memberikan *feedback* ke materi yang ada dan kelas tersebut tidak disetujui. *Guru* yang kelasnya disetujui oleh *admin* kemudian harus melengkapi materi-materi untuk tiap-tiap topik di kelas yang ia buat. Setelah *guru* melengkapi materi sesuai pembelajaran yang telah dibuat di awal, maka *guru* akan melakukan *publish* kelas. Namun kelas tersebut belum resmi di *publish* dan menunggu *approval* dari *admin* untuk melakukan *publish* kelas tersebut pada galeri kelas. Sejak kelas sudah berhasil dibuat, *admin* dapat memberi *feedback* dan *guru* pun dapat memberi tanggapan pada setiap *feedback*.

Siswa yang ingin mengikuti kelas *online* terlebih dahulu dapat melihat daftar kelas yang ada di *galery* kelas. Di sana siswa dapat melihat detail kelas yang meliputi profil singkat pengajar, deskripsi kelas, kurikulum dan juga *review* kelas. Deskripsi kelas mencakup keterangan singkat kelas, judul kelas, dan harga. Sedangkan profil pengajar yang ditampilkan mencakup nama, jenjang pendidikan dan biografi singkat pengajar.

Siswa yang sudah menentukan pilihan kelasnya bisa langsung mendaftarkan diri menjadi partisipan kelas tersebut jika sudah melakukan *login* terlebih dahulu. Dalam proses pendaftaran pembayaran dilakukan di luar sistem dengan cara transfer ke rekening ruangguru dan mengirim *email* konfirmasi. Setelah *admin* memasukan siswa ke dalam daftar partisipan sebuah kelas, siswa tersebut memiliki akses untuk melihat materi.

Setiap partisipan di dalam sebuah kelas dapat memberi *review* terhadap kelas tersebut dengan cara memberi *rating* dan komentar. Komentar tersebut tidak akan ditampilkan pada publik sebelum mendapat *approval* dari *admin*. Setelah mendapat *approval*, guru dapat memberi tanggapan terhadap setiap komentar yang diberikan.

Komentar dapat diberikan berkali-kali selama masih terdaftar sebagai partisipan pada kelas tersebut. Di samping itu, jika seorang siswa sudah terdaftar sebagai partisipan pada sebuah kelas maka siswa tersebut akan terdaftar selamanya kecuali jika *admin* mengeluarkannya.

### 3. Analisis Kebutuhan Fungsional

Bagian ini terdiri atas daftar aktor yang menggunakan Sistem Informasi Kuliah Online Ruangguru, *Use Case*, dan daftar *use case* yang terdapat dalam sistem informasi ini.

#### 3.1. Use Case Diagram

Gambar *use case diagram* dapat dilihat pada Annex A.

#### 3.2. Action Description

Sistem Informasi Kuliah Online Ruangguru akan digunakan oleh ;

*Tabel 1. Deskripsi Aktor*

No	Nama Aktor	Deskripsi Aktor
1	User	Bertindak sebagai pengguna sistem
2	<i>Admin</i>	<i>Admin</i> adalah IT <i>officer</i> dari ruangguru.com
3	Guru	Semua guru yang telah masuk dalam seleksi ruangguru.com
4	Murid	Semua orang yang telah disetujui oleh <i>Admin</i> untuk mengikuti kelas tertentu
5	<i>AdminDanGuru</i>	Bertindak sebagai pengguna sistem yang bisa saling berkomunikasi

### 3.3. Daftar Use Case

Daftar *use case* yang terdapat dalam Sistem Informasi Kuliah Online Ruangguru antara lain :

Tabel 2. Daftar Use Case

Kode	Nama Use case	Deskripsi	Primary Actor
UCS-01	Melihat Daftar Calon Partisipan	<i>Use case</i> ini bertujuan untuk menjelaskan skenario ketika <i>Admin</i> meminta sistem untuk menampilkan daftar semua partisipan yang telah mendaftar kelas namun belum disetujui oleh <i>Admin</i> . <i>Admin</i> mengelola partisipan dari <i>dashboard Admin</i> .	<i>Admin</i>
UCS-02	Melihat Daftar Partisipan	<i>Use case</i> ini bertujuan untuk menjelaskan <i>scenario</i> ketika <i>User</i> meminta sistem untuk menampilkan daftar semua partisipan yang terdapat pada suatu kelas yang telah dibuat oleh Guru dan telah <i>dipublish</i> oleh <i>Admin</i> .	<i>User</i>
UCS-03	Mengaktifkan Partisipan	<i>Use case</i> Mengaktifkan Partisipan ini menjelaskan skenario ketika murid telah mendaftar kelas tertentu akan tetapi statusnya masih belum aktif sebagai partisipan kelas sehingga membutuhkan <i>Admin</i> untuk mengaktifkan status murid tersebut agar dapat mengikuti kelas yang telah didaftarnya.	<i>Admin</i>
UCS-04	Menonaktifkan Partisipan	<i>Use case</i> Menonaktifkan Partisipan ini menjelaskan skenario ketika murid telah mendaftar kelas tertentu akan tetapi statusnya masih belum aktif sebagai partisipan kelas sehingga membutuhkan <i>Admin</i> untuk mengaktifkan status murid tersebut agar dapat mengikuti kelas yang telah didaftarnya.	<i>Admin</i>
UCS-05	Mendaftar Kelas	<i>Use case</i> Mendaftar kelas menjelaskan mengenai skenario proses murid dalam mendaftar kelas yang telah dibuat Guru dan telah <i>dipublish Admin</i> .	Murid
UCS-06	<i>Retrieve</i> Kelas	<i>Use case</i> ini menerangkan skenario ketika <i>user</i> ingin melihat detail sebuah kelas	<i>User</i>
UCS-07	<i>Create</i> Kelas	<i>Use case</i> ini menerangkan skenario ketika guru membuat sebuah kelas baru pada sistem. Guru mengisi isian mengenai informasi kelas dan sistem akan melakukan validasi <i>output</i> . Jika data yang diisi <i>valid</i> , sistem akan menyimpan sebuah kelas baru.	Guru
UCS-08	<i>Update</i> Kelas	<i>Use case</i> ini menerangkan <i>scenario</i> ketika guru mengubah detail kelas yang sebelumnya telah ditentukan oleh guru tersebut. Mencakup nama, deskripsi, harga, <i>tag</i> dan topik-topik pada kelas tersebut.	Guru

UCS-09	<i>Delete Kelas</i>	<i>Use case</i> ini menerangkan skenario ketika guru ingin menghapus sebuah kelas yang pernah dibuatnya.	Guru
UCS-10	Melihat Daftar Kelas	<i>Use case</i> ini menerangkan skenario ketika <i>user</i> ingin melihat daftar kelas yang sesuai dengan hak akses <i>user</i> tersebut.	<i>User</i>
UCS-11	Mengubah Status Kelas	<i>Use-case</i> ini menerangkan tentang skenario ketika <i>admin</i> mengubah status kelas. Dianggap <i>admin</i> telah membuka sebuah kelas dan mengakses materi pada kelas tersebut. Jika <i>admin</i> menganggap materi di kelas tersebut layak, <i>admin</i> akan mengubah status kelas dan sistem akan menyimpan perubahan status tersebut.	<i>Admin</i>
UCS-12	Melihat <i>List Feedback</i>	<i>Use case</i> ini menerangkan skenario ketika <i>user</i> ingin melihat daftar <i>feedback</i> yang diberikan kepada sebuah kelas.	<i>AdminDanGuru</i>
UCS-13	Memberi <i>Feedback</i>	<i>Use-case</i> ini menerangkan tentang skenario ketika <i>admin</i> memberikan <i>feedback</i> terhadap kelas. <i>Admin</i> mengisi sebuah isian yang memuat pendapat dan keterangan terkait kelas serta materi kelas dan sistem akan melakukan validasi input dan jika <i>input valid</i> secara otomatis menyimpan data.	<i>Admin</i>
UCS-14	Menanggapi <i>Feedback</i>	<i>Use-case</i> ini menerangkan tentang skenario ketika guru memberikan tanggapan terhadap <i>feedback</i> yang didapat terhadap kelas yang ia buat. Guru mengakses halaman kelas kemudian melihat <i>feedback</i> dan memberikan tanggapan. Sistem akan melakukan validasi <i>input</i> dan jika <i>input valid</i> secara otomatis menyimpan data.	Guru
UCS-15	Melihat Daftar Topik Kelas	<i>Use Case</i> ini menjelaskan tentang bagaimana aktor dapat melihat daftar topik yang ada pada suatu kelas. Aktor memilih salah satu kelas yang ingin dilihat topiknya, kemudian sistem akan menampilkan semua daftar topik yang ada pada kelas tersebut.	<i>User</i>
UCS-16	Melihat Daftar Materi	<i>Use Case</i> ini menjelaskan bagaimana aktor dapat melihat daftar materi pada suatu topik kelas.	<i>User</i>
UCS-17	<i>Retrieve Materi</i>	<i>Use Case</i> ini menjelaskan tentang bagaimana aktor dapat <i>meretrieve</i> /mengakses materi pembelajaran yang akan ditampilkan dalam bentuk video atau pdf.	<i>User</i>
UCS-18	<i>Create Materi</i>	<i>Use Case</i> ini akan menjelaskan bagaimana proses guru menambah atau <i>upload</i> materi pada suatu kelas. Guru dapat <i>upload</i> materi berupa video atau file sesuai dengan format yang ditentukan. Setelah guru <i>upload</i> materi,	Guru



		sistem akan menyimpan materi tersebut.	
UCS-19	<i>Delete Materi</i>	<i>Use Case</i> ini menjelaskan tentang bagaimana guru menghapus materi kelas.	Guru

*Use Case Specification* berada pada dokumen terpisah.

### 3.4. Assumptions

Semua *user* yang memakai sistem ini sudah memahami teknik dasar penggunaan teknologi *website*. Mengenai pengelolaan sistem, Sistem Informasi Kuliah Online Ruangguru memerlukan *admin* yang sudah memahami teknik-teknik CRUD pada *database*.

### 3.5. Dependencies

Sistem Informasi Kuliah Online Ruangguru merupakan subsistem dari ruangguru.com sehingga semua hal harus mengacu dari sistem ruangguru.com. Sistem ini bergantung pada *server* di mana sistem disimpan, yaitu sebuah *dedicated server* yang dikelola oleh pihak ruangguru.com. Sedangkan untuk mengakses *server* tersebut dibutuhkan koneksi internet.

## 4. Analisis Kebutuhan Non-Fungsional

Analisa kebutuhan non-fungsional dapat dilihat pada dokumen terlampir yang terpisah dengan dokumen ini yaitu di *Supplementary Specification*.

## 5. Analisis Data

### 5.1. Entity Relationship Diagram

Rancangan konseptual *database* akan delapan entitas yang terdiri dari empat *strong entity* dan empat *weak entity*. Keempat *strong entity* tersebut terdiri dari Entitas Guru, *Admin*, Murid dan *Tag*. Sedangkan keempat lainnya yaitu *weak entity* yang terdiri dari Kelas, Topik, Materi dan *Feedback*. Berdasarkan *strong entity* tersebut terdapat *entity* yang berfungsi sebagai *master data* yaitu Guru, *Admin*, dan Murid.

- **Strong Entity Guru**

Entitas Guru dikategorikan sebagai *strong entity* karena entitas tersebut dapat berdiri sendiri dan tidak bergantung dengan adanya entitas lainnya. Ada atau tidaknya entitas lain, Guru tetap akan ada. Entitas Guru mempunyai atribut *id* sebagai *primary key* dan atribut lainnya yaitu nama, *email*, *password*, bio, pendidikan. Setiap Guru dapat membuat beberapa Kelas.

- **Weak entity Kelas**

Entitas Kelas dikategorikan sebagai *weak entity* karena keberadaan entitas Kelas tidak dapat berdiri sendiri, artinya keberadaan entitas Kelas bergantung keberadaan entitas lain dalam hal ini yaitu entitas Guru. Kelas hanya akan ada apabila Guru membuat Kelas. Entitas Kelas mempunyai atribut *id\_kelas* sebagai *primary key* dan atribut lainnya yaitu nama, deskripsi, *tgl\_publish*, *approval* dan harga. Setiap Kelas mempunyai satu Guru.

- **Strong entity Admin**

Entitas *Admin* dikategorikan sebagai *strong entity* karena entitas tersebut dapat berdiri sendiri dan tidak bergantung dengan adanya entitas lainnya. Ada atau tidaknya entitas lain, *Admin* tetap akan ada. Entitas *Admin* bertugas mengelola Kelas, baik itu melakukan *approval* kelas maupun publish Kelas. *Approval* akan dilakukan jika Kelas yang diajukan oleh Guru berisi minimal lima materi yang sesuai dengan standar ruangguru.com. Setelah *Admin* melakukan *approval*, Guru dapat melengkapi konten materi secara penuh hingga akhir topik. Setelah itu, *Admin* akan mengecek kembali apakah Kelas tersebut benar-benar layak untuk dilakukan *publish* ataukah tidak, jika layak maka akan dipublish dan jika tidak maka akan ditolak dengan disertai *feedback* mengenai konten yang diunggah. Entitas *Admin* mempunyai atribut *email* sebagai *primary key* dan atribut lainnya yaitu nama dan *password*. Setiap *Admin* dapat mengelola beberapa Kelas. Sedangkan satu Kelas hanya akan dikelola satu *Admin*.

- **Strong Entity Murid**

Entitas Murid dikategorikan sebagai *strong entity* karena entitas tersebut dapat berdiri sendiri dan tidak bergantung dengan adanya entitas lainnya. Ada atau tidaknya entitas lain, Murid tetap akan ada. Entitas Murid mempunyai atribut *id* sebagai *primary key* dan atribut lainnya yaitu *email*, *password* dan nama. Setiap Murid dapat mendaftar beberapa Kelas yang telah dibuat oleh Guru.

- **Weak entity Topik**

Entitas Topik dikategorikan sebagai *weak entity* karena keberadaan entitas Topik tidak dapat berdiri sendiri, artinya keberadaan entitas Topik bergantung keberadaan entitas lain dalam hal ini yaitu entitas Kelas. Topik hanya akan ada apabila Kelas

dibuat. Setiap Kelas mempunyai beberapa Topik. Atribut entitas Topik terdiri dari *id* sebagai *primary key* dan atribut lainnya yaitu judul.

- **Weak entity Materi**

Entitas Materi dikategorikan sebagai *weak entity* karena keberadaan entitas Materi tidak dapat berdiri sendiri, artinya keberadaan entitas Materi bergantung keberadaan entitas lain dalam hal ini yaitu entitas Topik. Materi hanya akan ada apabila terdapat Topik. Setiap Topik terdiri dari beberapa Materi. Materi-materi yang telah dibuat oleh Guru dapat dibuka oleh Murid. Entitas Materi mempunyai atribut *id* sebagai *primary key* dan atribut lainnya yaitu judul, *notes* dan *urlFile*.

- **Weak Entity Feedback**

Entitas *Feedback* dikategorikan sebagai *weak entity* karena keberadaan entitas *Feedback* tidak dapat berdiri sendiri, artinya keberadaan entitas *Feedback* bergantung keberadaan entitas lain dalam hal ini yaitu entitas Kelas. *Feedback* hanya akan ada apabila Kelas dibuat. Setiap kelas terdiri dari beberapa *Feedback*. Atribut entitas *Feedback* yaitu *id* sebagai *primary key* dan atribut lainnya yaitu waktu\_kirim, pesan, tanggapan dan waktu\_tanggapan.

- **Strong Entity Tag**

Entitas *Tag* dikategorikan sebagai *strong entity* karena entitas tersebut dapat berdiri sendiri dan tidak bergantung dengan adanya entitas lainnya. Ada atau tidaknya entitas lain, *Tag* tetap akan ada. Setiap Kelas mempunyai beberapa *Tag*. Atribut entitas *Tag* terdiri dari *id* sebagai *primary key* dan atribut lainnya yaitu subjek.

*Entity Relationship Diagram* Sistem Informasi Kuliah Online Ruangguru dapat dilihat pada Annex B.

## 5.2. Class Diagram

- **Guru**

Objek ini menyimpan nomor identitas guru, alamat *email* dan *password* sebagai atribut validasi. Selain itu juga disimpan beberapa deskripsi guru yaitu nomor identitas guru, paragraf singkat biografi dan riwayat pendidikan yang dimiliki. Selain itu objek ini juga menyimpan objek Kelas

- **Admin**

Objek ini hanya menyimpan atribut alamat *email*, *password* dan nama untuk data standar. Selain itu objek ini juga menyimpan objek kelas.

- **Murid**

Objek ini hanya menyimpan nomor identitas murid, alamat *email*, *password* dan nama untuk data standar.

- **Kelas**

Objek ini menyimpan nomor identitas kelas, nama, nominal harga, deskripsi singkat mengenai materi dan status kelas. Status kelas dibagi menjadi tiga. Status *not approve* dan *approve* yang ditandai dengan atribut *approval*. Sedangkan status

*published* ditandai jika sudah terisi tanggal pertama publikasi. Selain itu objek ini juga menyimpan objek *Feedback* dan *Topik*

- ***Feedback***

Objek ini menyimpan nomor identitas *feedback*, teks pesan dari *admin* dan teks tanggapan dari guru. Dimana setiap penyimpanan pesan dan tanggapan akan disimpan waktu pembuatannya.

- ***ReviewDanStatus***

Objek ini menyimpan nomor identitas murid, *email* murid, dan nilai *rating*. Selain itu juga menyimpan atribut dari objek *Kelas* dan *Murid*. Sedangkan objek yang disimpan adalah objek *Komentar*.

- ***Komentar***

Objek ini menyimpan nomor identitas komentar, teks pesan, dan teks tanggapan.

- ***Tag***

Objek ini menyimpan nomor identitas dan juga hashtag.

- ***Topik***

Objek ini menyimpan nomor identitas, dan judul topik. Selain itu juga menyimpan objek *Materi*.

- ***Materi***

Objek ini menyimpan nomor identitas, judul, *notes* dan alamat file materi.

- ***Akses***

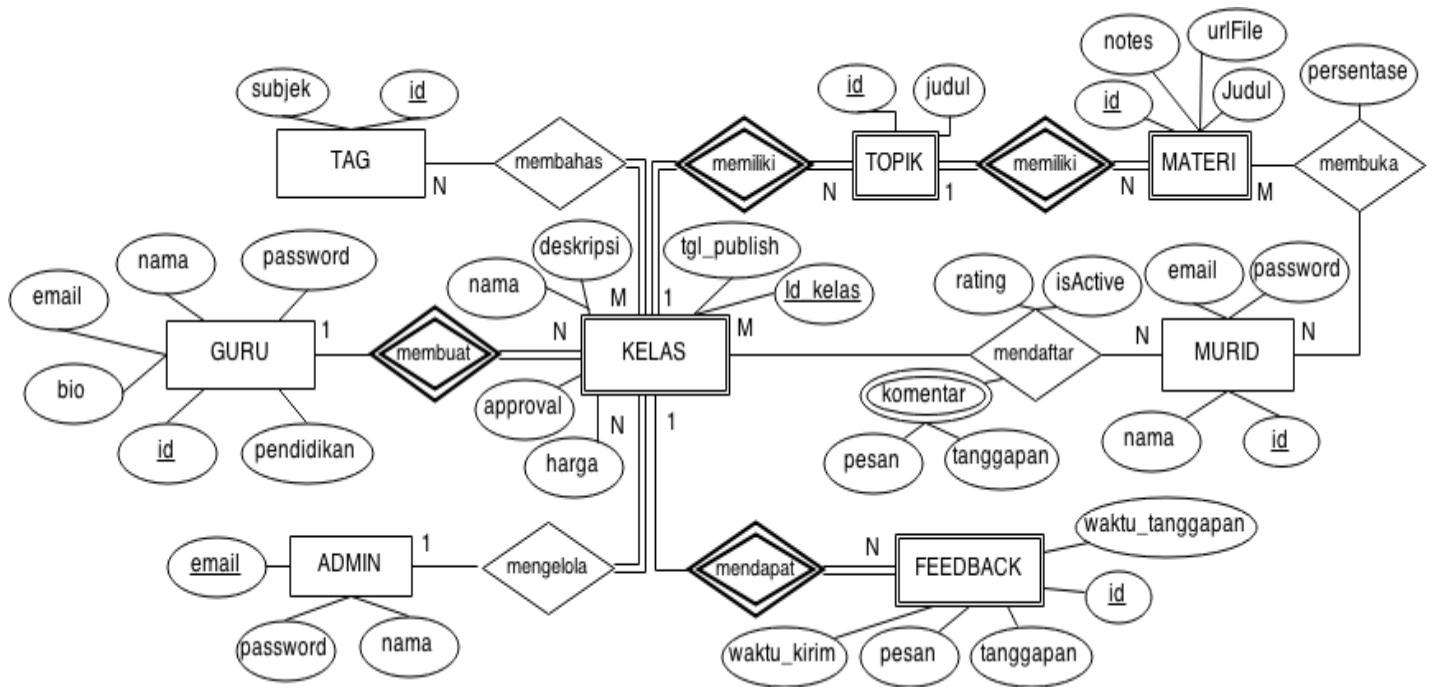
Objek ini menyimpan atribut dari objek *Materi* dan *Murid* serta persentase bagian materi yang sudah dibuka.

Class diagram Sistem Informasi Kuliah Online Ruangguru dapat dilihat pada Annex C.

# ANNEX A



## ANNEX B



## Annex C

