Requirements Specification

for

LEARNING MANAGEMENT SYSTEM

**March 2019**

**Version 0.1**

**Prepared By**

**J2Lite Software**

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Updated By** |
| 12 Mar 2019 | 0.1 | Initial Draft | J2Lite Software |

**Document Approvals**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Signature** |
|  |  |  |

**Table of Contents**

[1 Introduction 6](#_Toc3620689)

[1.1 Purpose 6](#_Toc3620690)

[1.2 Document Conventions 6](#_Toc3620691)

[1.3 Project Scope 6](#_Toc3620692)

[1.4 References 6](#_Toc3620693)

[2 System Description 7](#_Toc3620694)

[3 Functional Requirements 7](#_Toc3620695)

[3.1 System Features 7](#_Toc3620696)

[3.1.1 System Feature 1 7](#_Toc3620697)

[3.1.2 System Feature 2 7](#_Toc3620698)

[3.1.3 System Feature 3 8](#_Toc3620699)

[3.1.4 System Feature 4 8](#_Toc3620700)

[3.1.5 System Feature 5 8](#_Toc3620701)

[3.1.6 System Feature 6 8](#_Toc3620702)

[3.1.7 System Feature 7 8](#_Toc3620703)

[3.2 Use Cases 8](#_Toc3620704)

[3.2.1 Use Case Diagrams 8](#_Toc3620705)

[3.2.2 Use Case 1 9](#_Toc3620706)

[3.2.3 Use Case 2 9](#_Toc3620707)

[3.2.4 Use Case 3 9](#_Toc3620708)

[3.2.5 Use Case 4 10](#_Toc3620709)

[3.2.6 Use Case 5 10](#_Toc3620710)

[3.2.7 Use Case 6 11](#_Toc3620711)

[3.2.8 Use Case 7 11](#_Toc3620712)

[3.2.9 Use Case 8 12](#_Toc3620713)

[3.2.10 Use Case 9 12](#_Toc3620714)

[3.2.11 Use Case 10 12](#_Toc3620715)

[3.2.12 Use Case 11 13](#_Toc3620716)

[3.2.13 Use Case 12 13](#_Toc3620717)

[3.3 Entity Relationship Diagrams 14](#_Toc3620718)

[3.4 Data Dictionary 15](#_Toc3620719)

[3.4.1 Entity 1 15](#_Toc3620720)

[3.4.2 Entity 2 15](#_Toc3620721)

[3.4.3 Entity 3 16](#_Toc3620722)

[3.4.4 Entity 4 16](#_Toc3620723)

[3.4.5 Entity 5 16](#_Toc3620724)

[4 External Interface Requirements 16](#_Toc3620725)

[4.1 External Interface Requirements Output 16](#_Toc3620726)

[4.2 External Interface Requirements Input 16](#_Toc3620727)

[5 Technical Requirements (Non functional) 16](#_Toc3620728)

[5.1 Performance 17](#_Toc3620729)

[5.2 Scalability 17](#_Toc3620730)

[5.3 Security 17](#_Toc3620731)

[5.4 Maintainability 17](#_Toc3620732)

[5.5 Usability 17](#_Toc3620733)

[5.6 Multi lingual Support 18](#_Toc3620734)

[5.7 Auditing and Logging 18](#_Toc3620735)

[5.7.1 Auditing 18](#_Toc3620736)

[5.7.2 Logging 18](#_Toc3620737)

[5.8 Availability 18](#_Toc3620738)

[6 Open Issues 18](#_Toc3620739)

# Introduction

## Purpose

*<Specify the purpose of the document including its target audience.>*

Untuk membantu pelajar dan pengajar memberikan materi ataupun pembelajaran secara sistem daring. Dalam program aplikasi ini pengajar dan pelajar bisa lebih interaktif dalam pemberian pelajaran dan pengumpulan tugas. Pengajar bisa juga menjadi murid dan murid bisa juga menjadi pengajar saat membuat kelas dalam aplikasi ini.

## Document Conventions

*<Indicate any special document conventions used including notation or approaches used for requirements capture. For example, you can specify that Crow's notation is used for ER diagrams.>*

## Project Scope

*<Specify the exact project scope indicating project boundaries. This can also include the purpose of the software project, its benefits and overall goals. In the case of a software product, this should contain product vision and should indicate the exact user base for the product. If you are aware of features that should go into a future version, list them here or add a new section on “Features for Future Releases”.>*

Program aplikasi ini memiliki jangkauan pengguna dalam konteks pendidikan yaitu :

1. Program aplikasi dapat memberikan interaksi pembelajaran yang mudah di gunakan.
2. Program aplikasi ini berbasis website dengan memberikan fitur pembuatan kelas, pemberian tugas, serta pengumpulan tugas yang di lakukan secara *online.*
3. Program aplikasi ini memberikan hak akses terhadap user tertentu yaitu hak akses terhadap pengajar, pemgajar memiliki hak akses penuh terhadap kelas yang di buat meliputi pembuatan dokumen baru atau tugas, penambahan guru baru dalam satu kelas tersebut.
4. Program aplikasi dapat memberikan infomasi yang di butuh oleh pelajar yang telah di berikan oleh pengajar sesuai dengan waktu pelampiran tugas.
5. Program aplikasi memberikan kode keamanan pada saat pembentukan kelas untuk memaksimalkan keamanan data yang telah di lampirkan.

## References

*<This should contain links to all external documents and web resources referred in the requirements. Sometimes prototypes or product vision are specified in a separate document. Link them here. Also ensure that you link to the correct version.>*

1. [*https://www.academia.edu/28993846/Software\_Requirement\_Specifications\_SRS\_untuk\_Aplikasi\_E-Commerce*](https://www.academia.edu/28993846/Software_Requirement_Specifications_SRS_untuk_Aplikasi_E-Commerce)diakses pada *12/03/2019 14.19WIB*
2. [*https://new.edmodo.com/home*](https://new.edmodo.com/home) diakses pada *12/03/2019 14.38 WIB*

# System Description

*<This section can contain a detailed overview of the software system being built. It can contain a subsection detailing the existing system followed by the proposed system details. If relevent, this section can also contain business details such as organizational hierarchy. Basically provide everything that is necessary to understand the Fuctional Requirements section which follows this.>*

Program aplikasi dijalankan oleh user pelajar atau pengajar, dari salah satu tersebut bisa menjadi admin kelas atau pengajar dan bisa mengundang user lain sebagai pelajar ataupun bisa menjadi pengajar, sebagai pengajar bisa memberikan tugas berupa file lalu mengunggah di kelas yang bersangkutan tersebut sehingga dapat di unduh oleh pelajar.

# Functional Requirements

*<This section contains system requirements followed by various requirement models which can be used for detailed design. In addition to the following sections, you can also add process flow diagrams, data flow diagrams, flowcharts and decision tables if required.>*

Kebutuhan sistem ini memerlukan beberapa fungsi yang seperti:

1. Register user
2. Login
3. Membuat kelas dapat sebagai pelajar atau pengajar
4. Mengunggah atau Mengunduh tugas/file
5. Pencarian kelas dengan kode kelas

## System Features

*<This section specifies the high level system features required in the software product. Each requirement given below has a unique tag that can be referred in the traceability matrix.>*

Dalam sistem ini memiliki beberapa fitur yaitu:

1. Register User
2. Log In/Out
3. Tambah kelas
4. Mengundang pelajar
5. Mengunggah file/tugas
6. Mengunduh file/tugas
7. Komentar

### System Feature 1

Register User dengan cara mengisi form yang sudah tersedia. Data yang perlu mereka masukkan adalah Nama, Username, Password, Foto.

### System Feature 2

Log In/Out dengan cara setelah melakukan register lalu user harus mengisi username dan password agar dapat mengakses halaman utama. Lalu kalau mau keluar tinggal klik log out yang sudah tersediah disetiap halaman kecuali form log in dan register.

### System Feature 3

User dapat membuka kelas setelah melakukan sesi log in. menambah kelas ini dapat dilakukan dengan cara memasukan nama kelas yang dibuka dan kode unik kelas sehingga murid dapat mencari kelas dengan mencari kode unik tersebut. Pembuat kelas tersebut akan berperan sebagai guru.

### System Feature 4

User yang membuka kelas di feature tersedia, dapat mengundang orang lain yang menjadi murid kelas tersebut sehingga semua pembaharuan yang dilakukan didalam kelas tersebut, akan ditampilkan di timeline setiap murid.

### System Feature 5

Mengunggah file/tugas dapat dilakukan oleh kedua peran baik sebagai guru atau murid. Guru akan mengunggah file dan membuat tugas sebagai tugas bagi para murid. Murid akan bisa mengumpulkan tugas.

### System Feature 6

Guru atau murid dapat mengunduh file yang mereka butuhkan. File yang dapat mereka unduh hanya mereka miliki aksesnya.

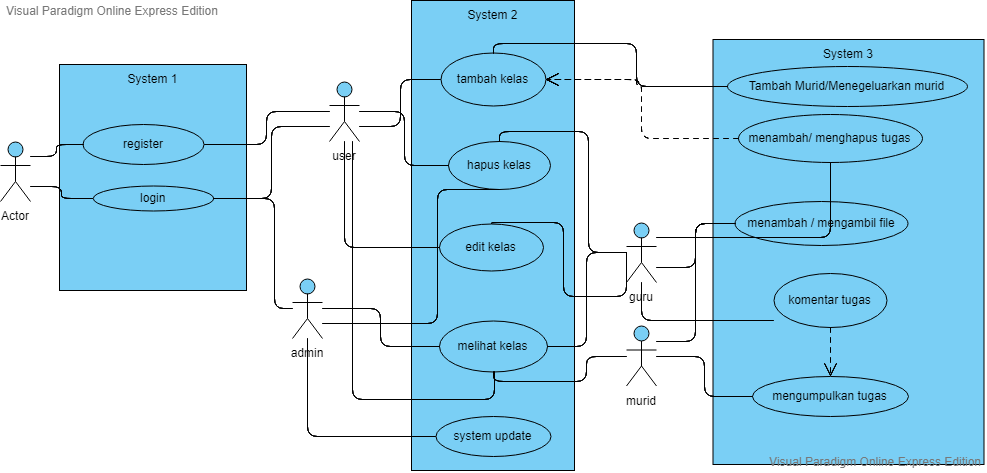
### System Feature 7

Guru dapat memberikan komentar dalam kolom komentar untuk memberikan penjelasan yang tertinggal dalam melampirkan file atau tugas. Murid dapat memberikan komentar terhadap file yang telah di berikan oleh guru serta memberikan tambahan komentar untuk memberikan interaksi terhadap murid dan guru.

## Use Cases

### Use Case Diagrams

*<One or more diagrams depicting how various actors interact with the software system.>*

**

### Use Case 1

<This provides a detailed description of the use case. Usually it is captured in the following table format. Add more rows or removing rows depending on your specific requirement.>

|  |  |
| --- | --- |
| **ID** | Register |
| **Description** | Actor mendaftarkan akun |
| **Actors** | Guru dan murid |
| **Preconditions** | Untuk dapat login kedalam website |
| **Basic Steps** | Memasukan data sesuai dengan form yang disediakan |
| **Alternate Steps** | - |
| **Exceptions** | Jika email sudah terdaftar, akan di hubungkan langsung ke login page. Jika username sudah terpakai, maka akan diminta untuk mencari username lain |
| **Business validations/Rules** | Akan muncul notifikasi bahwa register berhasil, dan akan dihubungkan ke login page sehingga user dapat langsung login |
| **Postconditions** | Dapat melakukan login dengan akun yang sudah dibuat |
|  |  |

### Use Case 2

|  |  |
| --- | --- |
| **ID** | Login |
| **Description** | Actor akan memasukan username dan password di halaman login page |
| **Actors** | Guru, murid dan admin |
| **Preconditions** | User akan bisa menggunakan fitur-fitur yang ada dalam website |
| **Basic Steps** | Memasukan username dan password pada form pada halaman login page |
| **Alternate Steps** | Remember token, sehingga user yang sudah pernah login tidak perlu memasukan username dan password kembali. |
| **Exceptions** | Jika username atau password terdapat kesalahan input maka akan diberi notifikasi pada halaman tersebut sehingga user dapat kembali memasukan username atau password. Namun jika sudah salah beberapa kali, maka akan diberi pilihan ubah password |
| **Business validations/Rules** | User akan langsung masuk ke halaman utama dari website |
| **Postconditions** | Dapat masuk kehalaman utama website dan memakai fitur-fitur |

### Use Case 3

|  |  |
| --- | --- |
| **ID** | Tambah kelas |
| **Description** | User dapat menambah kelas |
| **Actors** | User (guru) |
| **Preconditions** | User akan bertindak sebagai guru dalam kelas yang dibuat. |
| **Basic Steps** | User mengisi form dalam website seperti kode unik kelas dan nama kelas |
| **Alternate Steps** | - |
| **Exceptions** | Ketika kode unik yang dimasukan oleh user sudah terpakai, maka user akan diminta untuk mencari kode unik lain |
| **Business validations/Rules** | Ketika kelas sudah dibuat, maka akan muncul halaman website yang menyediakan pilihan menambah tugas/file ataupun menambah murid |
| **Postconditions** | User, yang bertindak sebagai guru menambahkan murid untuk kelas tersebut |

### Use Case 4

|  |  |
| --- | --- |
| **ID** | Menghapus kelas |
| **Description** | Ketika kelas sudah tidak terpakai/ sudah tutup, maka kelas dapat dihapus atau jika admin menemukan bahwa kelas ini tidak pantas, maka akan dihapus oleh admin |
| **Actors** | Guru, admin |
| **Preconditions** | Guru, sebagai user yang berhak menghapus kelas yang dibuat sebelumnya. Semua data yang terlibat dalam kelas tersebut akan terhapus |
| **Basic Steps** | Guru atau admin akan menekan sebuah tombol yang mengarahkan ke proses penghapusan |
| **Alternate Steps** | - |
| **Exceptions** | - |
| **Business validations/Rules** | Kelas yang dihapus tidak akan muncul dihalaman utama baik guru maupun murid |
| **Postconditions** | Jika guru menghapus kelas sendiri, maka kelas tersebut akan otomatis hilang dari daftar kelas user guru tersebut.  Jika dihapus oleh admin maka akan diberi notifikasi sehingga user terkait mengetahui. |

### Use Case 5

|  |  |
| --- | --- |
| **ID** | Edit kelas |
| **Description** | Seperti merubah deskripsi, megubah nama kelas dan lain-lain |
| **Actors** | Guru |
| **Preconditions** | Detail kelas akan berubah sesuai dengan apa yang dimasukan oleh user guru sebagai perubahan |
| **Basic Steps** | User menekan sebuah tombol yang mengarahkan ke form edit kelas dan merubah data sesuai keinginan. Jika sudah selesai, user akan menekan tombol lain yang mengaktifkan proses edit. |
| **Alternate Steps** | - |
| **Exceptions** | - |
| **Business validations/Rules** | Detail kelas akan berubah |
| **Postconditions** | Detail dari kelas yang berubah akan mempengaruhi semua user yang berada dikelas tersebut. Detail kelas yang berubah akan diketahui semua user. |

### Use Case 6

|  |  |
| --- | --- |
| **ID** | Melihat kelas |
| **Description** | Kelas, berisi nama, kode unik dan deskripsi akan ditampilkan |
| **Actors** | User (guru dan murid), dan admin |
| **Preconditions** | - |
| **Basic Steps** | Ketika user membuka halaman tertentu, maka akan ditampilkan semua kelas yang user buat |
| **Alternate Steps** | Beberapa kelas akan ditampilkan dihalaman utama setelah login. |
| **Exceptions** | Kelas tidak bisa ditampilkan ketika kelas dihapus oleh admin, user akan menerima pemberitahuan |
| **Business validations/Rules** | - |
| **Postconditions** | Kelas-kelas yang berada didalam daftar dapat ditekan kemudian akan menyambungkan dengan halaman yang berisi detail kelas tersebut. |

### Use Case 7

|  |  |
| --- | --- |
| **ID** | System update |
| **Description** | Pengembangan website dengan menambah fitur-fitur yang dibutuhkan |
| **Actors** | Admin |
| **Preconditions** | - |
| **Basic Steps** | Admin menambahkan fitur yang dibutuhkan di website |
| **Alternate Steps** | - |
| **Exceptions** | - |
| **Business validations/Rules** | Fitur pada website akan bertambah atau berubah sesuai dengan apa yang dilakukan admin |
| **Postconditions** | - |

### Use Case 8

|  |  |
| --- | --- |
| **ID** | Memasukan / mengeluarkan murid |
| **Description** | Guru akan menekan suatu tombol yang akan mengaktifkan fungsi delete untuk murid tertentu |
| **Actors** | User (guru) |
| **Preconditions** | Pengurangan murid pada kelas terkait |
| **Basic Steps** | User yang berperan sebagai guru menekan tombol yang mengaktifkan fungsi hapus murid yang dimaksud |
| **Alternate Steps** |  |
| **Exceptions** | - |
| **Business validations/Rules** | Murid yang dihapus dari kelas tersebut akan hilang dari daftar murid, dan murid tersebut tidak lagi mendapat akses untuk kelas tersebut |
| **Postconditions** |  |

### Use Case 9

|  |  |
| --- | --- |
| **ID** | Menambah / menghapus tugas |
| **Description** | User yang berperan sebagai guru menambah atau menghapus tugas |
| **Actors** | User sebagai guru |
| **Preconditions** | Tugas yang ditambahkan akan masuk ke daftar tugas dari kelas tersebut,, sedangkan tugas yang dihapus akan hilang dari daftar tugas |
| **Basic Steps** | User sebagai guru akan mennekan sebuah tombol yang mengaktifkan fungsi tambah atau hapus tugas. Untuk menambah tugas, guru akan mengisi form tambah tugas seperti nama tugas dan tanggal batas pengumpulan |
| **Alternate Steps** |  |
| **Exceptions** | Jika terdapat kolom yang belum diisi, maka tugas tidak ditambahkan, user akan diminta untuk mengisi kolom yang kosong.  Jika tanggal batas pengiriman telah lewat, maka akan kembali untuk mengganti tanggal batas pengumpulan |
| **Business validations/Rules** | Daftar tugas akan bertambah dan murid akan bias mengunggah file yang dibutuhkan sesuai ketentuan tugas |
| **Postconditions** | Ketika tanggal batas pengumpulan telah lewat, maka link upload akan tidak bias digunakan |

### Use Case 10

|  |  |
| --- | --- |
| **ID** | Menambah atau mengambil file |
| **Description** | User dapat mengunggah atau mengunduh file |
| **Actors** | User (guru dan murid) |
| **Preconditions** | - |
| **Basic Steps** | Untuk menambah file, user (guru) akan masuk ke halaman tertentu dan mengunggah file.  Untuk mengambil file, hanya dapat dilakukan guru dan seluruh murid (jika penambah file adalah guru) dan hanya dapat dilakukan oleh guru dan murid yang mengunggah file (jika penambah file adalah murid) |
| **Alternate Steps** | - |
| **Exceptions** | ketika file yang ingin diunggah tidak dapat diunggah (karena ekstensi tidak terdaftar, dll), maka akan diberikan notifikasi bahwa pengunggahan gagal. |
| **Business validations/Rules** | Akan muncul notifikasi apakah proses unggah atau unduh berhasil atau tidak |
| **Postconditions** | File yang di unduh dari website akan masuk kedalam memori penyimpanan device sedangkan file yang di unggah di website akan masuk kedalam daftar file suatu kelas |

### Use Case 11

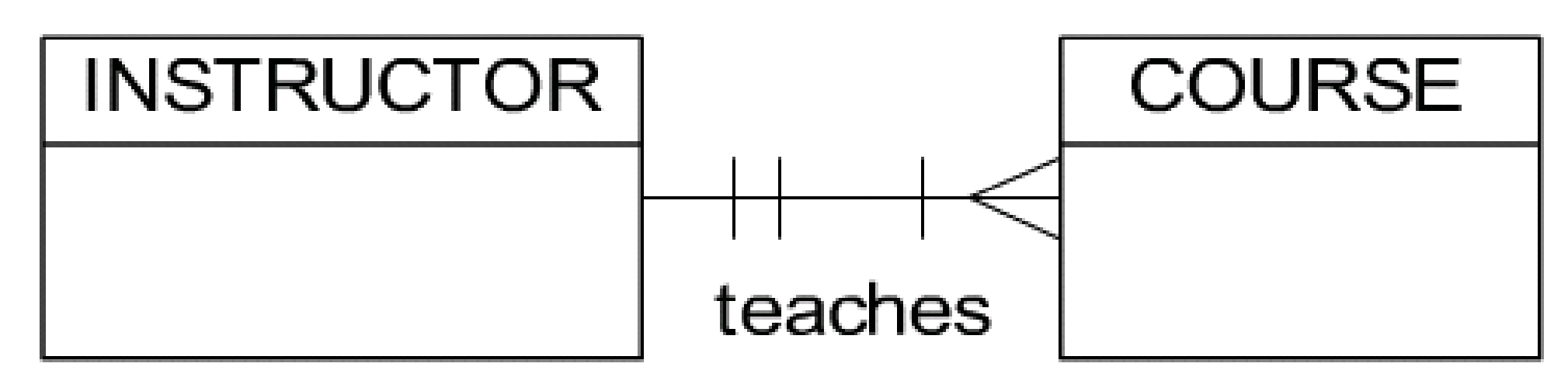
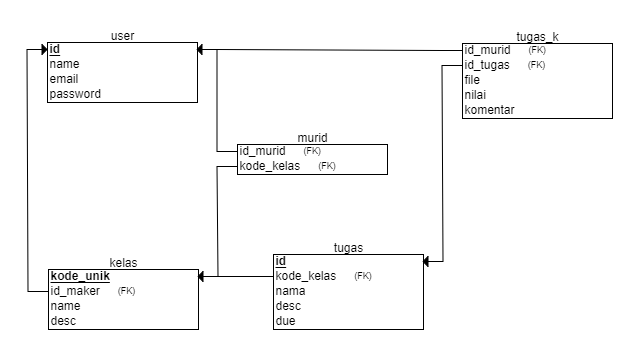
|  |  |
| --- | --- |
| **ID** | Komentar tugas |
| **Description** | User sebagai guru akan memberikan komentar terkait sebuah tugas seperti nilai, permintaan revisi dll. |
| **Actors** | User sebagai guru |
| **Preconditions** | File tugas yang dikumpulkan para murid dibuka dan telah dibaca sehingga dapat dikomentari |
| **Basic Steps** | Mengisi kolom komentar yang tersedia disalah satu halaman website |
| **Alternate Steps** | - |
| **Exceptions** | - |
| **Business validations/Rules** | Jika penambahan komentar berhasil, maka murid ‘pemilik’ tugas tersebut akan dapat melihat komentar yang diberikan. |
| **Postconditions** | - |

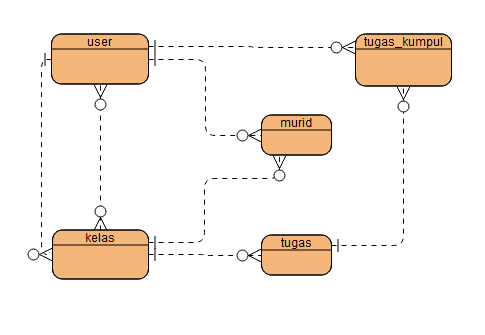
### Use Case 12

|  |  |
| --- | --- |
| **ID** | Mengumpulkan tugas |
| **Description** | User sebagai murid mengunggah file tugas yang dibutuhkan |
| **Actors** | User sebagai murid |
| **Preconditions** | Guru membuat tugas baru dengan detail tertentu, murid akan mengunggah file sebagai expected output |
| **Basic Steps** | Murid akan mengunggah file tugas di salah satu halaman website |
| **Alternate Steps** | - |
| **Exceptions** | Jika file tidak terunggah, maka akan ditampilkan notifikasi gagal |
| **Business validations/Rules** | Jika file terunggah, akan ditampilkan notifikasi yang menandakan unggah berhasil |
| **Postconditions** | - |

## Entity Relationship Diagrams

*<One or diagrams to depict all the entities in the system and their relationships. You can use different notations for ER diagrams. Following is a very simple ER diagram in Crow's notation.>*





## Data Dictionary

### Entity 1

<A brief description of the entity followed by a table containing all its attributes as shown below. >

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Optional?** | **Notes** |
| Id | int | n | auto increment, primary key |
| Name | varchar (100) | n |  |
| Email | varchar(100) | n |  |
| password | varchar(100) | n | password encrypt |

### Entity 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Optional?** | **Notes** |
| kode\_unik | varchar(20) | n | auto increment, primary key |
| id\_maker | int | n | foreign key |
| Nama | varchar(50) | n |  |
| Desc | varchar(200) | y |  |

### Entity 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Optional?** | **Notes** |
| Id | int | n | auto increment, primary key |
| kode\_kelas | varchar(20) | n | foreign key |
| Nama | varchar(100) | n |  |
| Desc | varchar(100) | y |  |
| Due | date | n |  |

### Entity 4

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Optional?** | **Notes** |
| id\_murid | int | n | foreign key |
| kode\_kelas | int | n | foreign key |

### Entity 5

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Optional?** | **Notes** |
| id\_murid | int | n | foreign key |
| id\_tugas | int | n | foreign key |
| File | varchar(225) | n |  |
| Nilai | int | n |  |
| Komentar | varchar(500) | y |  |

# External Interface Requirements

*<Provide sub sections for each external interface. Identify all the input and ouput from the external interfaces.>*

# External Interface Requirements Output

Bagi pengguna/user pertama membuat register yang terdiri dari, Nama, Username, Email, dan Password. Lalu mengisi data log in yaitu masukkan Username/Email dan password lalu akan masuk kehalaman utama pada web ini. User dapat membuat kelas.

# External Interface Requirements Input

Monitor, Keyboard, Mouse, PC, Laptop, SmartPhone

# Technical Requirements (Non functional)

*<Please note that all the following subsections may not be applicable for a system. Sometimes you will have to add additional sections (for example, Legal requirements)>*

## Performance

*<For example, What is the response time required?>*

Waktu respon yang diperlukan untuk membuka website ini adalah untuk kecepatan internet 1 Mbps adalah 0.47s untuk mengakses 8.26 Kbytes.

## Scalability

*<For example, how many users the system should support after two years of operation?>*

Target pengguna setelah dua tahun adalah 5000 user.

## Security

*<For example, is data encryption required?>*

Enskripsi data hanya dilakukan di data password. Sedangkan untuk mengamankan file-file yang dari user-user yang tidak berwenang, kami akan menerapkan sistem privilege

## Maintainability

Penambahan fitur-fitur, mencari terjadinya error pada system dan memperbaiki system yang error.

## Usability

* + - 1. Dalam Program ini dapat memudahkan User untuk mengakses tugas yang diberikan guru pada kelas tersebut dan dapat mengirim tugas tersebut kepada guru.
      2. Dalam Program ini dapat memudahkan User karena User tidak perlu memerlukan 2 akun sekaligus hanya untuk menjadi guru dan murid karna pada program ini User hanya membutuhkan 1 user dan bisa menjadi guru ataupun murid.
      3. Analisis berdasarkan 5 komponen usability:
         1. Learnability

Sistem akan kami buat semudah mungkin untuk digunakan, untuk membantu user baru untuk mengerti cara menggunakan produk kami, kami akan menyediakan sebuah video yang diharapkan dapat membantu pengguna baru.

* 1. Efficiency

Website kami hanya menampilkan informasi atau file-file yang berhubungan dengan user tersebut dan akan menyediakan fitur pencarian sehingga akan lebih mempermudah proses pencarian informasi yang dibutuhkan

* 1. Memorability

Pengguna akan mudah mengingat website ini karena semua fitur kami akan sudah aktif sejak pertama kami rilis, perubahan dikemudian hari hanyalah perubahan-perubahan kecil seperti tampilan, use-flow, dan hal-hal kecil lain

* 1. Errors

-

* 1. Satisfaction

Dengan sistem website yang mudah digunakan, menyediakan informasi yang berkaitan dan dibutuhkan serta tidak perlunya pengguna mempelajari kembali website, kami berharap pengguna dapat merasakan secara maksimal manfaat website ini.

## Multi lingual Support

*<What are the languages that software system should support?>*

Menggunakan Bahasa Indonesia dan Bahasa Inggris kecuali terdapat perubahan dikemudian hari.

## Auditing and Logging

## Auditing

Untuk Sistem keamanan dalam program ini semua password sudah di encrypt sehingga tidak mudah untuk orang lain dapat mengetahui password pada user lainnya.

## Logging

Dalam program perlu juga pencatatan jam pada saat meunggah file atau tugas kedalam kelas dan juga perlu pencatatan jam untuk meunggah file kepada guru sehingga guru tau kapan murid kumpul tugasnnya

## Availability

*<For example, Is any kind of downtime acceptable or required?>*

Website aktif 24 jam kecuali ketika ada maintenance / perbaikan sistem yang dilakukan seeminggu sekali dengan waktu 5 jam di hari Sabtu jam 22:00 – 03:00.

# Open Issues

*<There could be open issues even at the end of the requirements elicitation. List of all of them here so it can be tracked and closed later. Some of these issues may later become project risks as well.>*

Dalam Lingkup Masalah Program ini yaitu:

Pembuatan kelas dimana user bisa menjadi Guru atau murid.

Dalam pembuatan kelas terdapat Kode\_Unik sehingga user lain dapat mengakses kelas tersebut.

Melakukan penambahan file atau tugas.

Mengumpulkan tugas.