

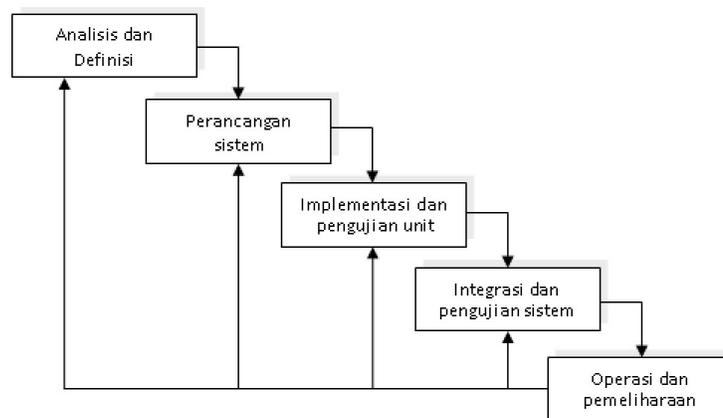
BAB III

METODE PENELITIAN

3.1 Metode Pengembangan Sistem

Pengembangan sistem dalam penelitian ini menggunakan pendekatan SDLC (*Software Development Life Cycle*) yang merupakan siklus pengembangan perangkat lunak. Di dalam SDLC, terdapat beberapa tahapan pengembangan di antaranya: perencanaan (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), pengujian (*testing*) dan pemeliharaan (*maintenance*).

Pada penelitian ini, peneliti menggunakan model SDLC *waterfall*. Model yang pertama kali diperkenalkan oleh Winston Royce pada tahun 1970 ini merupakan sebuah model klasik di mana siklus pengembangan perangkat lunaknya dilakukan dengan aliran sistem yang linear sehingga sebuah tahapan harus selesai terlebih dahulu sebelum berpindah ke tahapan selanjutnya. Selain itu, dalam model *waterfall* luaran dari tahapan sebelumnya menjadi masukan bagi tahapan berikutnya. Peneliti menggunakan metode *waterfall* menurut referensi Sommerville namun tidak termasuk tahapan *maintenance*. Gambar berikut menunjukkan metode yang dimaksud:



Gambar 3.1 Metode *Waterfall* Menurut Referensi Sommerville

3.1.1 Analisis dan Definisi Kebutuhan

Sebelum menganalisis, peneliti perlu melakukan pengumpulan data terkait permasalahan-permasalahan yang dialami oleh para narasumber dalam melaksanakan ujian dan menyusun strategi pembelajaran. Analisis ini terfokus pada kebutuhan fungsi perangkat lunak yang dapat menangani permasalahan-permasalahan tersebut.

3.1.2 Perancangan Sistem

Pada tahapan ini peneliti menyusun proses, data, aliran proses dan hubungan antar data yang paling optimal untuk menjalankan proses sesuai dengan *Requirement Analysis*. Setelah mendapatkan hasil dari analisis kebutuhan, peneliti mengubah kebutuhan-kebutuhan tersebut menjadi sebuah desain sistem beserta aliran prosesnya.

3.1.3 Implementasi dan Pengujian Unit

Dalam tahapan ini, peneliti mengubah desain yang telah dibuat sebelumnya menjadi sebuah sistem yang dapat berjalan sesuai kebutuhan. Pengembangan sistem ini dilakukan dari awal hingga sistem siap dijalankan. Sistem akan dijalankan pada perangkat komputer sehingga proses pembuatan pun harus menggunakan metode yang dapat dimengerti oleh komputer. Dalam hal ini, metode yang dipakai adalah *coding* menggunakan bahasa pemrograman C#.

3.1.4 Integrasi dan Pengujian Sistem

Pada tahapan ini, seluruh fungsi dari sistem yang dibangun akan diuji untuk memastikan bahwa fungsi-fungsi tersebut bebas dari *error*. Selain itu, proses pengujian ini juga dimaksudkan untuk memastikan bahwa hasil yang didapat benar-benar sesuai dengan kebutuhan yang telah didefinisikan.

3.2 Alat dan Bahan

3.2.1 Alat

A. Perangkat Keras

Spesifikasi laptop yang digunakan peneliti untuk membangun sistem ini dapat dilihat pada tabel berikut:

Tabel 3.2.1.1 Tabel perangkat keras

Perangkat Keras	Tipe
Sistem Operasi	Windows 10 Pro
Prosesor	Intel(R) Core(TM) i7-6700HQ CPU 2.60 GHz
RAM	8 GB
Hard Disk	1 TB

B. Perangkat Lunak

Tabel berikut menunjukkan daftar perangkat lunak yang digunakan peneliti dalam membangun sistem ini:

Tabel 3.2.1.2 Tabel perangkat lunak

No	Perangkat Lunak	Fungsi
1	Visual Studio 2017	Digunakan sebagai IDE (<i>Integrated Development Environment</i>) dalam membangun sistem.
2	SQL Server Express	Digunakan sebagai pengolah sekaligus server basis data.
3	IIS Server	Digunakan sebagai web server lokal agar sistem dapat diakses baik oleh guru maupun siswa.
4	Balsamiq	Digunakan untuk merancang tampilan sistem.
5	Violet UML Editor	Digunakan untuk merancang diagram.

3.2.2 Bahan

Bahan yang digunakan dalam penelitian ini adalah data yang didapat dari hasil observasi terhadap sistem-sistem serupa yang sudah ada dan kuesioner yang disebar secara *online* oleh peneliti pada bulan Mei sampai Juli 2017. Selain itu peneliti juga melakukan observasi terhadap web yang sudah menerapkan sistem ujian berbasis komputer.

3.3 Metode Pengumpulan Data

3.3.1 Observasi

Kegiatan ini bertujuan untuk mengetahui bagaimana proses kerja dari sebuah web yang sudah menerapkan sistem ujian berbasis komputer. Observasi sejatinya memiliki 2 (dua) tipe di antaranya adalah *participant observation* (observasi berperan serta) dan *non-participant observation* (observasi tanpa berperan serta).

Observasi yang dilakukan peneliti adalah jenis observasi *non-participant*. Dalam proses ini, peneliti mengamati tanpa terlibat langsung ke dalam proses tersebut. Adapun objek yang diteliti adalah web *e-learning* Universitas Muhammadiyah Yogyakarta. Pengumpulan data meliputi proses kerja yang dilakukan penyelenggara dan peserta ujian dalam pelaksanaan ujian.

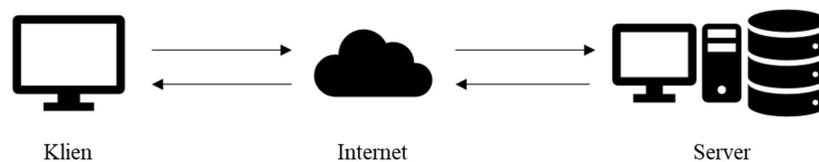
3.3.2 Kuesioner

Peneliti menyebarkan kuesioner secara *online* yang diperuntukkan bagi siswa SMA dan mahasiswa baru serta melakukan wawancara kepada guru SMA. Kegiatan ini adalah bentuk lain dari proses pengumpulan data.

3.4 Analisis Data

3.4.1 Arsitektur

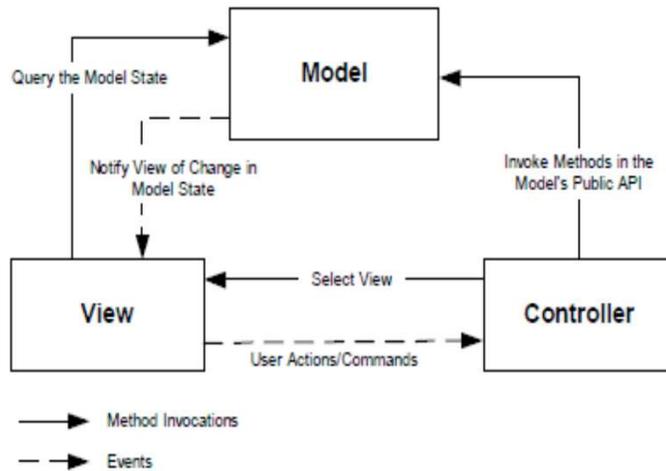
Sistem ini dijalankan pada sebuah server web lokal dalam suatu instansi. Begitu pula dengan basis data yang digunakan. Hal ini membuat masing-masing instansi memiliki server dan basis data yang hanya bisa diakses dari dalam instansi tersebut. Maka dari itu, peneliti menyimpulkan bahwa rancangan arsitektur dari sistem ini adalah sebagai berikut:



Gambar 3.4.1.1 Arsitektur Sistem

Server web yang digunakan adalah IIS Server karena sudah menjadi sepaket dengan sistem operasi Windows. Sedangkan server basis data menggunakan SQL Server 2017 Express. Siswa dan guru mengakses sistem melalui PC yang terhubung ke jaringan lokal (internet) lalu sistem memberikan *feedback* berupa antarmuka yang dapat digunakan untuk melakukan komunikasi dengan basis data seperti membuat ujian, menambah soal dan menambah peserta bagi guru serta mengerjakan ujian bagi siswa.

Adapun pola yang digunakan dalam membangun sistem ini adalah MVC (Model-View-Controller). MVC adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (*Trygve Reenskaug*) untuk meng-enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface* (Deacon, 2009). Hubungan dasar dari *model*, *view* dan *controller* dalam pola MVC dapat dilihat pada gambar berikut:



Gambar 3.4.1.2 Hubungan antara *model*, *view* dan *controller* (Gulzar, 2002)

3.4.2 Analisis Pengguna

Berdasarkan hasil observasi dan penyebaran kuesioner pada bulan Mei sampai Juli 2017 secara *online*, dapat disimpulkan bahwa dalam pelaksanaan ujian terdapat 2 (dua) jenis pengguna yaitu guru dan siswa. Berikut ini adalah tugas dari pengguna jenis guru:

- Membuat, memperbaharui dan menghapus informasi ujian. Informasi-informasi ini nantinya diakses oleh siswa dalam pelaksanaan ujian.
- Membuat, memperbaharui dan menambah data ke dalam bank soal. Soal-soal yang berada dalam bank soal nantinya dapat dijadikan soal ujian.
- Menambah dan menghapus data peserta dalam sebuah ujian. Peserta-peserta ini diisi dengan data siswa. Setelah terdaftar menjadi peserta barulah siswa dapat melaksanakan ujian.

Sedangkan tugas dari siswa adalah sebagai berikut:

- Berpartisipasi ke dalam ujian. Siswa dapat bergabung ke dalam suatu ujian yang telah dibuat oleh guru.
- Mengerjakan soal ujian.

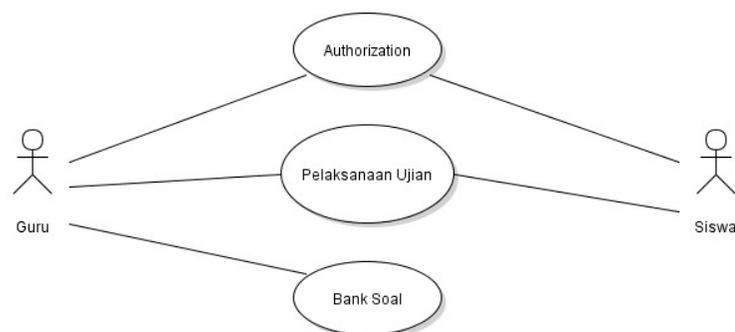
3.4.3 Analisis Kebutuhan Sistem

Berikut ini daftar kebutuhan-kebutuhan sistem berdasarkan analisis yang telah dilakukan:

1. Sistem harus memiliki fitur untuk membuat ujian karena untuk mendapatkan rekap terkait hasil ujian, perlu diadakan ujian terlebih dahulu.
2. Sistem harus memiliki pengguna dengan peran yang berbeda agar dapat menangani fitur di mana hanya pengguna dengan peran tertentu yang dapat mengaksesnya.
3. Sistem harus memiliki fitur yang berfungsi untuk membuat soal agar dapat dimasukkan ke dalam sebuah ujian. Hal ini diperlukan agar siswa dapat mengerjakan soal-soal yang diuji.
4. Sistem harus memiliki fitur yang berfungsi untuk menambahkan beberapa siswa ke dalam daftar peserta ujian. Siswa-siswa yang dapat mengerjakan suatu ujian adalah yang sudah terdaftar menjadi peserta ujian bagi ujian tersebut.

3.5 Rancangan Sistem

3.5.1 Diagram *Use Case*



Gambar 3.5.1.1 Diagram *Use Case* Sistem

Pada gambar diatas, terdapat 2 (dua) pengguna yang dapat mengoperasikan sistem ini di antaranya guru dan siswa. Baik guru maupun siswa memiliki ID yang digunakan sebagai pembeda antara pengguna satu dan pengguna lainnya. Setelah

melakukan proses otorisasi, barulah pengguna dapat menjalankan sistem sesuai dengan *role* (peran) masing-masing.

Dalam pelaksanaan ujian, guru dapat melakukan hal-hal berikut:

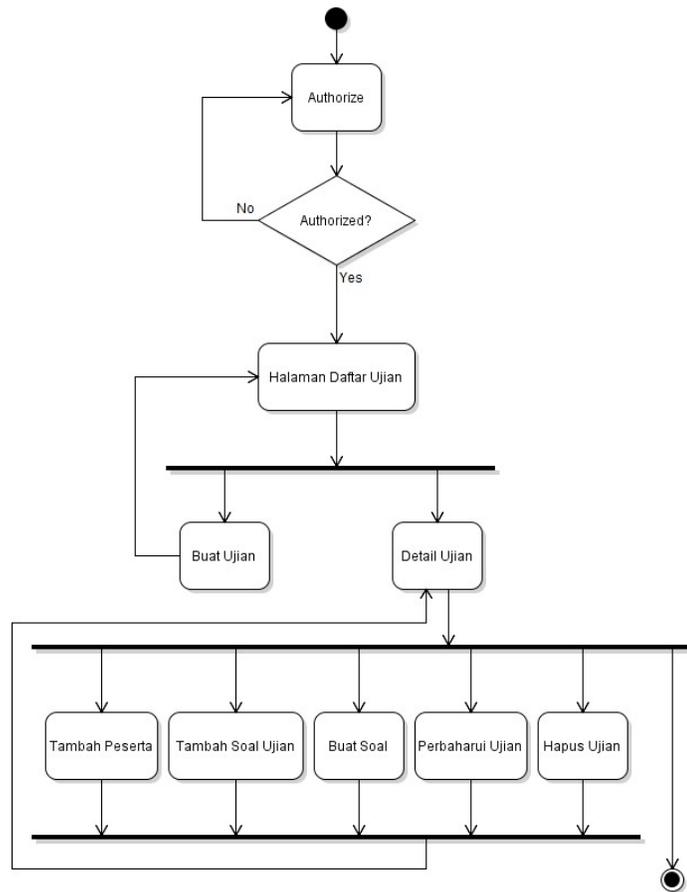
- B. Membuat ujian: Guru dapat membuat ujian sesuai kebutuhan dengan memasukkan informasi mengenai nama, kode masuk (*enrolment key*), tanggal mulai, tanggal batas pengambilan dan durasi pengerjaan ujian.
- C. Memperbaharui ujian: Setelah membuat ujian guru dapat melakukan perubahan terhadap informasi yang telah dimasukkan saat membuat ujian.
- D. Menghapus ujian: Guru dapat menghapus ujian yang telah dibuat baik sebelum maupun sesudah ujian tersebut dilaksanakan.
- E. Menambah soal ujian: Guru dapat menambahkan beberapa soal ke dalam ujian. Soal-soal tersebut didapat dari bank soal.
- F. Menambah peserta: Guru dapat menambahkan siswa ke dalam daftar peserta ujian. Hal ini hanya bisa dilakukan sebelum ujian berlangsung.
- G. Menghapus peserta: Selain dapat menambah peserta ujian, guru juga dapat menghapus peserta yang sudah terdaftar.

Sedangkan hal-hal yang dapat dilakukan siswa dalam pelaksanaan ujian adalah sebagai berikut:

- A. Partisipasi dalam ujian: Apabila guru belum menambahkan siswa ke dalam daftar peserta ujian, siswa dapat berpartisipasi dengan memasukkan informasi kode masuk ujian.
- B. Mengerjakan ujian: Siswa yang terdaftar sebagai peserta dapat mengerjakan soal-soal ujian dengan lama pengerjaan sesuai dengan durasi yang dibuat oleh guru.

3.5.2 Diagram *Activity*

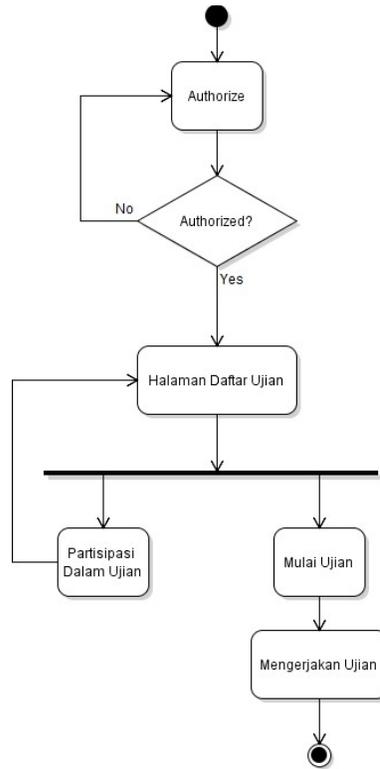
Dari diagram *use case* yang telah dibuat, akan didapat skenario dalam bentuk diagram *activity* bagi aktor-aktor yang berperan dalam lingkup proses pelaksanaan ujian. Gambar berikut menunjukkan diagram *activity* dari aktor guru:



Gambar 3.5.2.1 Diagram Activity Guru

Proses guru memiliki beberapa aktivitas yang dapat dilakukan. Akan tetapi, guru harus melewati tahap *otorisasi*. Setelah tahap tersebut berhasil dilewati, guru akan masuk ke dalam proses di mana guru dapat memilih ingin membuat ujian baru atau melihat detail ujian yang pernah dibuat. Apabila guru memilih untuk membuat ujian baru maka guru akan dimintai data terkait informasi ujian yang ingin digelar. Namun, apabila guru memilih untuk melihat detail ujian maka guru kembali diberi kebebasan untuk memilih apa yang akan dilakukan selanjutnya. Aktivitas-aktivitas yang tersedia dalam detail ujian adalah tambah peserta, tambah soal ujian, buat soal baru, perbaharui informasi ujian dan hapus ujian.

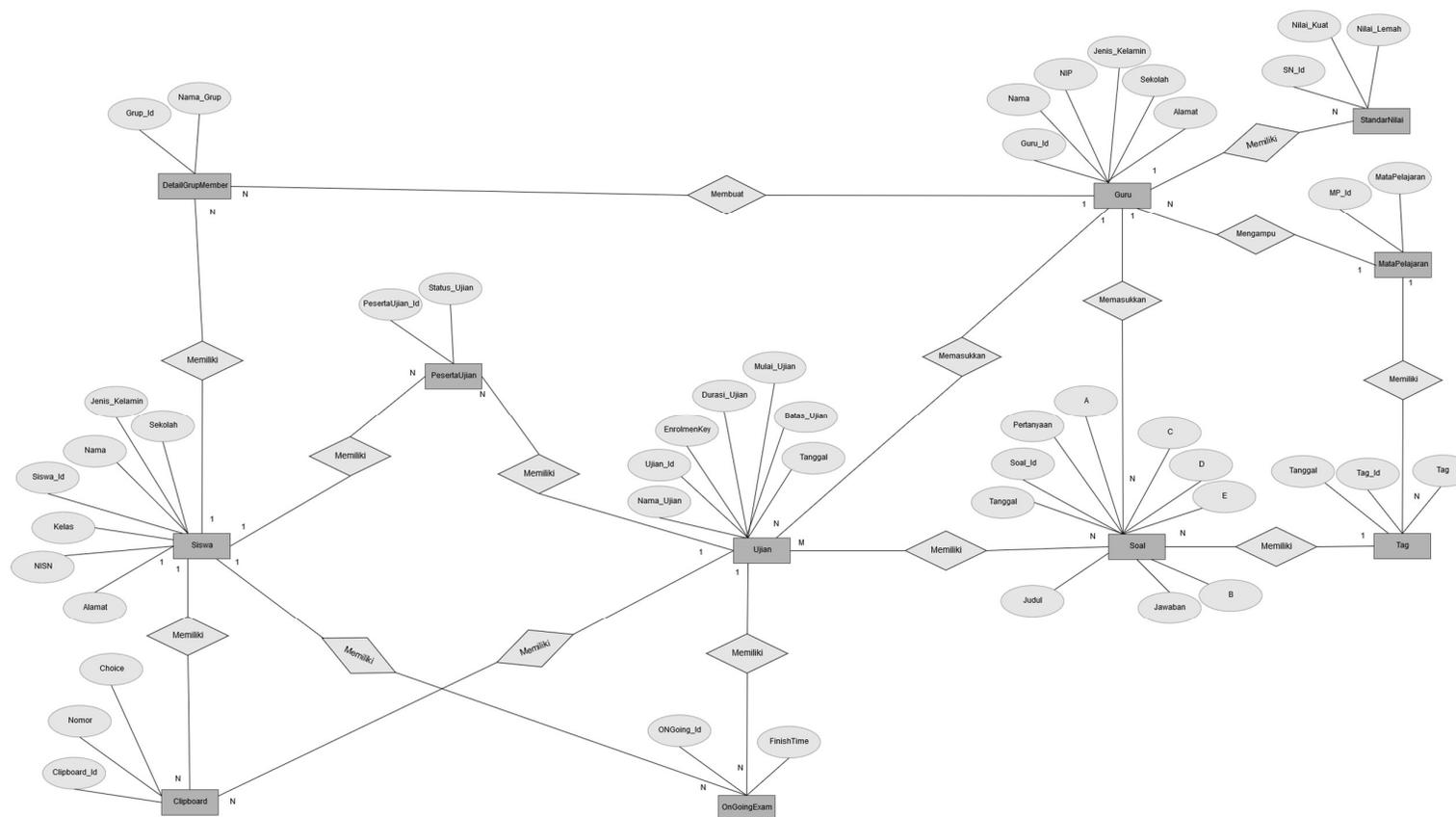
Di sisi lain, aktivitas-aktivitas yang dapat dilakukan siswa dengan sistem lebih sedikit bila dibandingkan dengan guru. Diagram *activity* siswa dapat dilihat pada gambar berikut ini:



Gambar 3.5.2.2 Diagram *Activity* Siswa

Sama seperti guru, siswa juga harus melewati proses *otorisasi* agar dapat melanjutkan ke aktivitas berikutnya. Pada halaman daftar ujian, terdapat informasi mengenai detail ujian yang sedang berlangsung maupun yang akan datang. Dalam proses ini, apabila siswa belum terdaftar menjadi peserta dalam suatu ujian maka siswa tidak diperkenankan untuk berpartisipasi dalam ujian tersebut. Akan tetapi, siswa dapat mendaftarkan diri dengan memasukkan kode masuk. Namun, apabila siswa sudah terdaftar maka siswa tersebut dapat mengikuti ujian pada saat ujian tersebut sedang berlangsung.

3.5.3 Diagram ERD



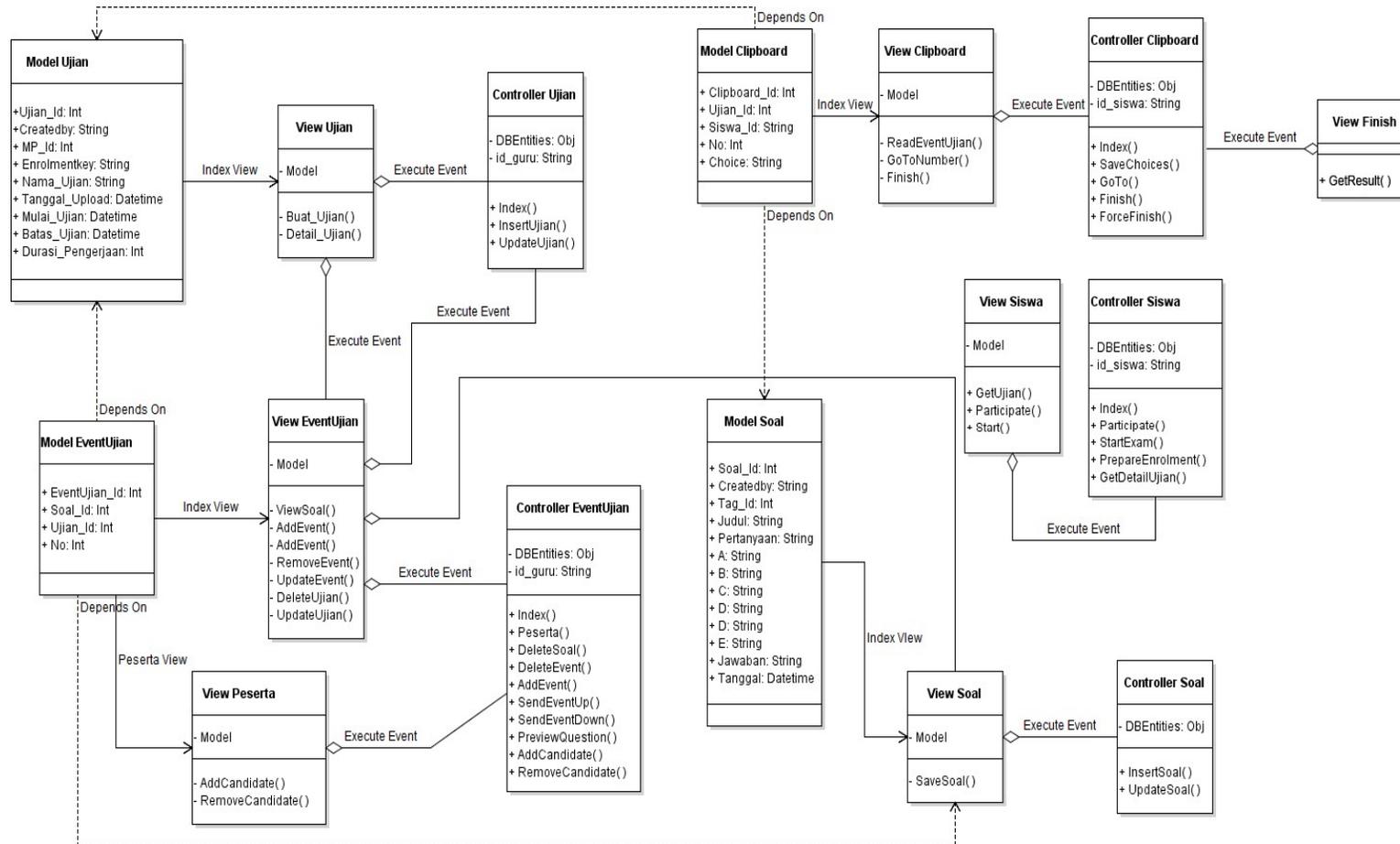
Gambar 3.5.3.1 Diagram ERD Sistem Asesmen dan Pemetaan Hasil Asesmen Berbasis Tag

Penjelasan terkait diagram ERD pada Gambar 3.5.3.1 adalah sebagai berikut:

- a. *Entity* Guru mempunyai relasi *many to 1* terhadap *entity* MataPelajaran.
- b. *Entity* Guru mempunyai relasi *1 to many* terhadap *entity* StandarNilai.
- c. *Entity* Guru mempunyai relasi *1 to many* terhadap *entity* Soal.
- d. *Entity* Ujian mempunyai relasi *many to many* dengan *entity* Soal.
- e. *Entity* Soal mempunyai relasi *many to 1* dengan *entity* Tag.
- f. *Entity* Tag mempunyai relasi *many to 1* terhadap *entity* MataPelajaran.
- g. *Entity* Ujian mempunyai relasi *many to 1* dengan *entity* Guru.
- h. *Entity* DetailGrupMember mempunyai relasi *many to 1* dengan *entity* Guru.
- i. *Entity* Clipboard relasi *many to 1* dengan *entity* Ujian.
- j. *Entity* Clipboard relasi *many to 1* dengan *entity* Siswa.
- k. *Entity* OnGoingExam mempunyai relasi *many to 1* dengan *entity* Siswa
- l. *Entity* OnGoingExam mempunyai relasi *many to 1* dengan *entity* Ujian.
- m. *Entity* PesertaUjian mempunyai relasi *many to 1* dengan *entity* Siswa.
- n. *Entity* PesertaUjian mempunyai relasi *many to 1* dengan *entity* Ujian.
- o. *Entity* Siswa mempunyai relasi *1 to many* dengan *entity* DetailGrupMember.

3.5.4 Diagram Class

Ilustrasi diagram *class* terkait atribut dan fungsi dari masing-masing *class* dalam sistem dapat dilihat pada gambar berikut:



Gambar 3.5.4.1 Diagram Class Sistem Ujian

Model, view dan *controller* yang memiliki hubungan dengan guru antara lain adalah Ujian, DetailUjian dan Soal. Sedangkan siswa berhubungan dengan *model, view* dan *controller* dari Siswa dan Clipboard. Dalam Ujian guru dapat membuat dan memperbaharui informasi suatu ujian. Kemudian, siswa dapat berpartisipasi dalam ujian tersebut melalui *model, view* dan *controller* Siswa. Pada *class* DetailUjian, guru dapat mengolah soal yang diambil dari bank soal kemudian dimasukkan ke dalam sebuah ujian. Soal-soal tersebut akan dikerjakan siswa melalui Clipboard. Setelah selesai mengerjakan, siswa akan dialihkan menuju *view* Finish untuk melihat hasil dari ujian yang dikerjakan.

3.6 Rancangan Antarmuka

A. Halaman Ujian Guru

The wireframe shows a user interface for a teacher's exam management page. At the top left, there is a header 'GURU (Label)'. On the right side, there are two columns of buttons: 'Rekap (Label)' and 'Ujian (Label)', with sub-buttons 'Rekap Try Out' and 'Rekap Tag' below them. The main area is divided into two sections. The first section, 'Ujian yang Akan Dilaksanakan (Label)', contains a table with columns 'No', 'Nama Ujian', and 'Tanggal Mulai', and a 'Lihat Detail (Button)' for each row. The second section, 'Ujian yang Telah Dilaksanakan (Label)', has an identical table structure. On the left side, there is a 'Foto' placeholder and a list of user profile fields: 'Nama (Label)', 'Kata Nama (Label)', 'ID Guru (Label)', 'Kata ID Guru (Label)', 'Role (Label)', 'Kata Role (Label)', 'Mapel (Label)', and 'Kata Mapel (Label)'. A 'Buat Ujian Baru (Label)' button is located above the first table.

Gambar 3.6.1 Rancangan Antarmuka Halaman Ujian Guru

Halaman ini memiliki beberapa tombol di antaranya adalah Buat Ujian dan Lihat Detail ujian serta daftar ujian baik yang akan dilaksanakan maupun yang sudah.

B. Form Buat Ujian

Gambar 3.6.2 Rancangan Antarmuka *Form* Buat Ujian

Form ini muncul saat guru menekan tombol buat ujian pada halaman Ujian Guru. *Form* Buat Ujian berfungsi untuk menambahkan data ujian baru ke dalam basis data sebelum data tersebut dapat ditampilkan pada halaman Ujian Guru.

C. Halaman Detail Ujian

Gambar 3.6.3 Rancangan Antarmuka Halaman Detail Ujian

Halaman ini muncul ketika guru menekan tombol Lihat Detail pada daftar ujian di halaman Ujian Guru. Terdapat 4 tombol primer pada halaman ini di antaranya adalah tombol Perbarui Ujian, Hapus Ujian, Peserta Ujian dan Tambah Soal.

D. *Form* Perbaharui Ujian

Gambar 3.6.4 Rancangan Antarmuka *Form* Perbaharui Ujian

Form ini dapat diakses setelah guru menekan tombol Perbaharui Ujian pada halaman Detail Ujian. *Form* ini berfungsi untuk mengubah data terkait informasi ujian.

E. *Form* Hapus Ujian

Gambar 3.6.5 Rancangan Antarmuka *Form* Hapus Ujian

Selain *form* Perbaharui Ujian, guru juga dapat menampilkan *form* Hapus Ujian melalui halaman Detail Ujian. *Form* ini dapat diakses setelah guru menekan tombol Hapus Ujian pada halaman tersebut. *Form* ini berfungsi untuk menghapus seluruh data yang berkaitan dengan ujian yang dihapus seperti peserta dan soal-soal ujian.

F. Halaman Tambah Soal

GURU (Label)	Rekap (Label)	Ujian (Label)
Judul Soal (Label)	Rekap Try Out	
	Rekap Tag	
<hr/>		
Pertanyaan		
<hr/>		
Jawaban A		
<hr/>		
Jawaban B		
<hr/>		
Jawaban C		

Gambar 3.6.6 Rancangan Antarmuka Halaman Tambah Soal

Dalam halaman ini guru dapat membuat soal baru yang nantinya dimasukkan ke dalam bank soal untuk mata pelajaran yang diajarkan guru tersebut. Halaman ini muncul setelah tombol Tambah Soal pada halaman Detail Ujian ditekan.

G. Halaman Peserta Ujian

Guru	Rekap (Label)	Ujian (Label)						
	Rekap Try Out							
	Rekap Tag							
<hr/>								
Kembali ke detail ujian								
Peserta Ujian : Nama Ujian								
<hr/>								
Siswa yang sudah terdaftar menjadi peserta ujian								
<table border="1"> <thead> <tr> <th>No</th> <th>Nama Siswa</th> <th>Hapus</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>			No	Nama Siswa	Hapus			
No	Nama Siswa	Hapus						
Masukkan nama siswa								
<input type="text"/>								
+ Nama siswa								
Nama (Label)	Kolom Nama							
NIP (Label)	Kolom NIP							
Sekolah (Label)	Kolom Sekolah							
Mapel (Label)	Kolom Mapel							
Alamat (Label)	Kolom Alamat							

Gambar 3.6.7 Rancangan Antarmuka Halaman Peserta Ujian

Pada halaman ini guru dapat mendaftarkan siswa menjadi peserta ujian. Terdapat dua jenis pencarian data siswa di antaranya adalah pencarian sesuai dengan nama dan grup (kelas).

H. Halaman Ujian Siswa

The diagram shows a user interface for a student exam page. At the top, there are three labels: 'SISWA (Label)', 'Rekap (Label)', and 'Ujian (Label)'. Below these, there are three buttons: 'Rekap Try Out', 'Rekap Tag', and 'Partisipasi (Button)'. The main content area features a circular placeholder for a 'FOTO' (photo) on the left. To the right of the photo is the text 'Ujian yang Akan Dilaksanakan (Label)'. Below this text is a table with the following headers: 'No', 'Nama Ujian', 'Pelaksana Ujian', and 'Tanggal Mulai'. At the bottom left, there are three sets of labels: 'Nama (Label)' and 'Kolom Nama (Label)', 'ID Siswa (Label)' and 'Kolom ID Siswa (Label)', and 'Role (Label)' and 'Kolom Role (Label)'. A 'Partisipasi (Button)' is located at the bottom right of the main content area.

Gambar 3.6.8 Rancangan Antarmuka Halaman Ujian Siswa

Halaman yang diperuntukkan bagi siswa ini memiliki daftar-daftar ujian yang sedang berlangsung, akan datang dan telah selesai dikerjakan.

I. Form Partisipasi

The diagram shows a participation form. At the top, there are three labels: 'SISWA (Label)', 'Rekap (Label)', and 'Ujian (Label)'. Below these, there are three buttons: 'Rekap Try Out', 'Rekap Tag', and 'Partisipasi (Button)'. The main content area contains a text input field with the label 'Anda akan berpartisipasi dalam ujian : (Label)'. Below this is another text input field with the label 'Masukkan enrolment key untuk ujian ini (Label)'. A 'Partisipasi (Button)' is located at the bottom left of the main content area.

Gambar 3.6.9 Rancangan Antarmuka Form Partisipasi

Apabila siswa belum terdaftar menjadi peserta ujian, siswa dapat mendaftarkan diri dengan mengisi *enrolment key* dalam *form* Partisipasi yang bisa didapat setelah menekan tombol Partisipasi pada halaman Ujian Siswa.

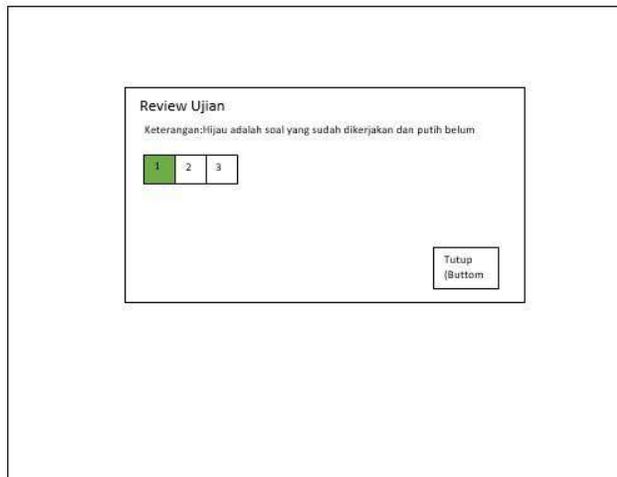
J. Halaman *Clipboard*



Gambar 3.6.10 Rancangan Antarmuka Halaman Clipboard

Pada halaman ini siswa mengerjakan soal-soal ujian yang dimasukkan guru pada halaman Detail Ujian. Terdapat empat tombol primer pada halaman ini yakni tombol Tinjau Pekerjaan, *Back*, *Next*, *Finish*.

K. *Popup* Tinjau Pekerjaan



Gambar 3.6.11 Rancangan Antarmuka Popup Tinjau Pekerjaan

Dalam *popup* ini siswa dapat melihat tombol-tombol berisikan nomor soal dengan warna hijau sebagai penanda bahwa soal tersebut sudah diisi dan warna putih jika belum. Saat menekan tombol-tombol tersebut maka siswa akan berpindah ke soal sesuai dengan nomor yang ditekan.