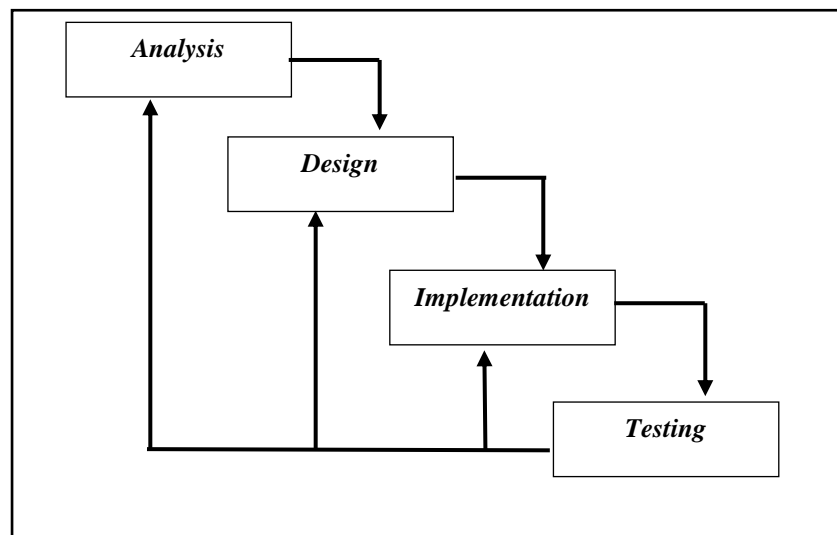


## BAB III METODE PENELITIAN

### 3.1 Metode Pengembangan Sistem

Proses pengembangan sistem pada penelitian ini menggunakan *model SDLC (Software Development Life Cycle)*. *Model SDLC* yang digunakan dalam penelitian ini adalah *model Waterfall*. Dalam pembuatan sistem ini penulis menggunakan metode *waterfall* menurut referensi Bassil (2011). Akan tetapi, tahapan *model Waterfall* ini hanya sampai pada tahap *testing*, dikarenakan pada penelitian ini tidak sampai pada tahap *maintenance*. Dan *testing* yang dilakukan hanyalah *testing* untuk fitur admin, hal ini dikarenakan fitur lain sudah diuji oleh anggota tim yang bertanggungjawab atas fitur tersebut. Untuk lebih jelasnya dapat dilihat pada **Gambar 3.1**.



**Gambar 3.1** Metode *Waterfall* Pada Penelitian

#### 3.1.1 *Analysis*

Sebelum melakukan analisis, penulis terlebih dahulu melakukan observasi terhadap web admin yang bertanggung jawab pada beberapa SMA di Sumatera dan Jawa. Data yang dikumpulkan adalah tugas dan proses kerja seorang web admin di sekolah, dan bagaimana sistem penyimpanan data yang dilakukan di sekolah.

Analisis difokuskan pada kebutuhan fungsi *software* untuk memenuhi kendala yang dialami oleh *user*.

Salah satu *requirement* dari sistem adalah satu sistem digunakan oleh satu sekolah. Hal ini dikarenakan data siswa dan guru pada setiap sekolah berbeda dan bersifat rahasia, sehingga sekolah – sekolah mempunyai *server* sendiri yang digunakan untuk penyimpanan data.

### **3.1.2 Design**

Setelah peneliti mendapatkan dokumen hasil dari analisis, maka peneliti mengubah kebutuhan-kebutuhan fungsi *software* tersebut menjadi sebuah bentuk desain sistem dan aliran proses dari sistem yang akan dirancang. Rancangan sistem dibuat menggunakan diagram-diagram UML. Diagram UML yang digunakan diantaranya: Diagram *Use Case*, Diagram *Activity*, ERD, Diagram *Class*, dan Diagram *Database*.

### **3.1.3 Implementation**

Bahasa pemrograman yang digunakan untuk mengimplementasikan desain yang telah dibuat menjadi sebuah sistem adalah bahasa pemrograman *C#*. Pada tahap ini, peneliti membangun sebuah aplikasi berdasarkan desain sistem yang telah dibuat. Pengembangan aplikasi ini dilakukan dari awal hingga aplikasi siap dijalankan. Penelitian ini merupakan *project* yang dilaksanakan oleh tim dengan masing-masing anggota mempunyai tanggung jawab yang berbeda. Agar data dapat diakses oleh semua anggota, maka data disimpan pada Azure.

### **3.1.4 Testing**

*Testing* yang dilakukan adalah *Unit Testing*. Pengujian ini dilakukan dengan melakukan pengujian terhadap *method – method* yang terdapat dalam sistem. Pengujian dilakukan dengan menggunakan metode *black box testing*.

## **3.2 Metode Pengumpulan Data**

### **3.2.1 Observasi**

Observasi yang dilakukan dalam penelitian bertujuan untuk mengetahui tugas dan proses kerja dan tugas seorang web admin di sekolah. Observasi yang digunakan adalah jenis observasi nonpartisipan. Dalam proses observasi ini,

peneliti hanya mengamati tanpa terlibat langsung dalam proses yang sedang diamati. Pada penelitian ini peneliti mendapatkan data dengan mengamati objek yang akan diteliti di beberapa tempat penelitian yaitu SMAN 1 Batang dan SMAN 1 Lubuk Alung Sumatera Barat. Pengumpulan data meliputi proses kerja yang dilakukan oleh admin dan tugas yang diberikan kepada seorang admin sekolah.

### **3.2.2 Studi Literatur**

Studi literatur digunakan oleh penulis untuk mendapatkan tambahan informasi tentang tugas admin pada sekolah yang menggunakan *e-learning*. Hal ini disebabkan oleh, data yang diperoleh dari hasil observasi menunjukkan tugas web admin pada sekolah yang tidak menggunakan sistem *e-learning*, sehingga tugas admin pada sekolah yang sudah menerapkan sistem pembelajaran dengan menggunakan *e-learning* belum tergambar dengan jelas.

Studi literatur dilakukan dengan mempelajari jurnal dan makalah mengenai *e-learning*, sistem ujian *online*, dan penelitian serupa yang sudah pernah dilakukan sebelumnya.

## **3.3 Analisis Data**

### **3.3.1 Analisis Pengguna**

Berdasarkan hasil observasi pada web admin di SMAN 1 Batang dan SMAN 1 Lubuk Alung Sumatera Barat, yang ditugaskan sebagai web admin adalah guru yang mengajar bidang studi TIK di sekolah. Tugas seorang web admin sekolah adalah menambah, memperbaharui, dan menghapus informasi – informasi yang terdapat pada *website*. Informasi – informasi yang dimaksud berupa data guru, data alumni, galeri kegiatan dan berita seputar sekolah.

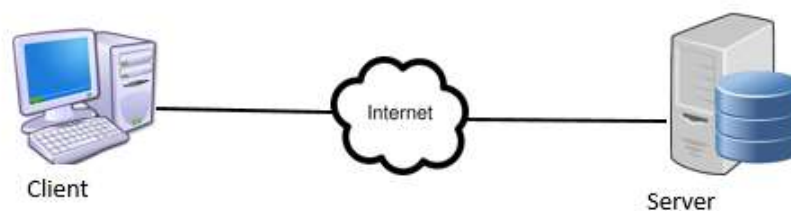
Pada SMAN 1 Lubuk Alung Sumatera Barat, rapor siswa dibagikan melalui aplikasi yang dapat diakses secara *online* oleh siswa dan wali murid. Tugas admin disini adalah mendaftarkan guru wali kelas dan siswa setiap kelas, sedangkan proses pengisian nilai dilakukan secara langsung oleh guru wali kelas. Kedua sekolah tersebut belum menerapkan proses pembelajaran dengan menggunakan *e-learning*, sehingga tugas admin hanya terfokus pada pengelolaan *website* sekolah.

Berdasarkan hasil studi literatur, tugas utama seorang web admin dalam penggunaan *e-learning* dan pengerjaan ujian *online* adalah mendaftarkan *user* yaitu guru dan siswa dan membuat jadwal ujian. Sedangkan pengisian soal dilakukan langsung oleh guru.

Dari tugas – tugas yang diberikan kepada seorang web admin, peneliti dapat menyimpulkan bahwa tugas seorang admin terfokus pada pengelolaan data *user* berupa data guru maupun data siswa, baik pada proses pengelolaan web sekolah, penggunaan *e-learning*, maupun ujian *online*. Sehingga sebuah sistem pembelajaran ataupun ujian yang berbasis *online*, membutuhkan seorang web admin untuk mengelola data siswa maupun guru yang dapat mengakses sistem tersebut.

### 3.3.2 Arsitektur

Aplikasi ini akan digunakan pada setiap sekolah dengan web *server* dan basis data yang berbeda untuk setiap sekolah, sehingga peneliti mengusulkan sebuah analisis rancangan arsitektur web secara umum seperti pada **Gambar 3.2**.



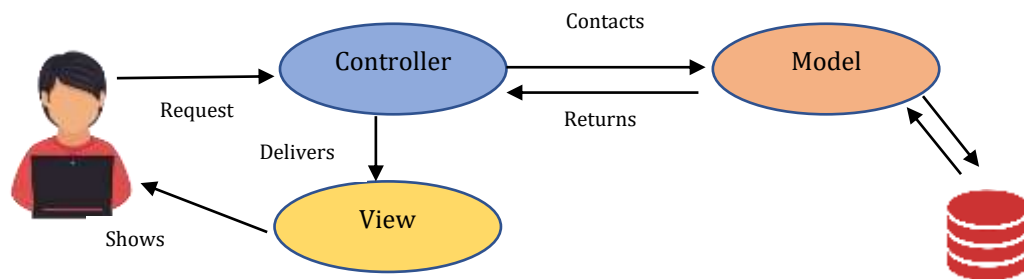
**Gambar 3.2** Arsitektur Aplikasi

*Database* yang digunakan pada *database server* adalah Microsoft SQL *Server* 2014, kemudian akan terhubung melalui web *server*. *User* mengakses web dengan menggunakan komputer client, saat *user* mengakses web, sistem akan memuat antarmuka web dan melakukan pengambilan data dari *database server*. Melalui antarmuka web, *user* dapat menambahkan data dan disimpan kedalam *database*.

Dalam pembuatan aplikasi ini, pola yang akan digunakan adalah MVC. Pola MVC memisahkan sebuah aplikasi, khususnya aplikasi *website* menjadi tiga bagian, yaitu:

1. *Model* : *Model* berperan untuk mengelola data, menyimpan dan mengambil *Entity* yang digunakan aplikasi dari sebuah *database*.
2. *View* : *View* memiliki fungsi untuk menampilkan data yang sebelumnya diolah oleh model. Berhubungan kepada *interface* akhir aplikasi yang akan dilihat oleh *user*.
3. *Controller*: *Controller* berguna sebagai penghubung antara *model* dan *view*. *Controller* menerima permintaan dari *client*, lalu memanggil *model* untuk melakukan operasi, kemudian mengirimkan data pada *view*. Akhirnya *view* akan menampilkan data tersebut dalam bentuk sebuah halaman html.

Arsitektur proses kontrol dapat dilihat pada **Gambar 3.3**.



**Gambar 3.3** Arsitektur proses kontrol MVC

### 3.3.3 Analisis Kebutuhan Sistem

Berikut ini merupakan analisis kebutuhan sistem berdasarkan analisis yang telah dilakukan:

1. Sistem harus mempunyai fitur untuk mendaftarkan guru dan siswa. Hal ini diperlukan karena tugas utama seorang admin adalah mengelola data *user* yang dalam penelitian ini adalah guru dan siswa.
2. Sistem harus terjaga keamanannya. Sistem mempunyai fitur *login* menggunakan *email* dan *password*. Hal ini diperlukan agar sistem hanya dapat digunakan oleh *user* yang telah didaftarkan sebelumnya.
3. Sistem harus dapat membatasi otorisasi *user* sesuai dengan perannya. Hal ini diperlukan karena fitur yang terdapat pada sistem berbeda untuk masing-masing *user*.

4. Sistem harus dapat memperbaharui data *user*. Hal ini dikarenakan perubahan pada data siswa kelas XII terjadi setiap tahun, sehingga data yang terdapat pada sistem haruslah data yang terbaru.

### 3.4 Alat dan Bahan

#### 3.4.1 Alat

Alat yang digunakan dalam penelitian ini mencakup perangkat keras dan perangkat lunak.

##### 1. Perangkat lunak untuk pengembang

Daftar perangkat lunak yang digunakan pengembang dapat dilihat pada **Tabel 3.1**.

**Tabel 3.1** Tabel perangkat lunak pengembang

No	Perangkat Lunak	Fungsi
1.	StarUML	Digunakan untuk pembuatan perancangan sistem.
2.	Microsoft Visual Studio 2015	Digunakan untuk pembuatan aplikasi.
3.	SQL Server Management Studio 2014	Digunakan untuk pengolahan <i>database</i> yang akan digunakan dalam aplikasi presensi.
4.	Azure	<i>Platform Cloud Computing</i> yang digunakan untuk menyimpan <i>database</i> agar dapat digunakan oleh semua anggota tim.

##### 2. Perangkat keras untuk pengembang

Spesifikasi laptop yang digunakan untuk pengembang dapat dilihat pada **Tabel 3.2**.

**Tabel 3.2** Tabel perangkat keras pengembang

<b>Perangkat Keras</b>	<b>Tipe</b>
<i>Sistem Operasi Processor</i>	Windows 10 Pro
<i>Processor</i>	Intel® Core™ i5-2450M Processor 2.50 GHz
<b>RAM</b>	4 GB
<i>Hard Disk</i>	500 GB

### 3.2.2 Bahan

Bahan yang digunakan adalah hasil analisis terhadap observasi yang telah dilakukan dan data yang diperoleh melalui studi literatur berdasarkan penelitian sebelumnya yang masih memiliki keterkaitan dengan aplikasi yang dikembangkan.

## 3.5 Rancangan Sistem

### 3.5.1 Use Case Diagram

Aplikasi ini melibatkan 3 aktor yaitu admin, guru, dan siswa. Setiap aktor dapat menggunakan fitur yang berbeda – beda pada system yaitu:

#### 1. Siswa

Dalam sistem ini hal yang dapat dilakukan oleh siswa adalah sebagai berikut:

- a. Mengerjakan ujian: Pada sistem ini siswa dapat mengerjakan ujian yang sudah disediakan di dalam sistem saat pertama kali siswa *login* ke halaman utama sistem.
- b. Rekap ujian : Siswa dapat langsung melihat *review* soal yang sudah dikerjakan dan dapat melihat grafik nilai dari ujian yang dikerjakan.pada grafik ini siswa dapat melihat hasil keseluruhan maupun hasil per *tag* atau setiap bab dalam mata pelajaran tersebut.
- c. Mengelola profil : Siswa dapat mengelola profil dengan melihat profil yang berisi data diri dan mengedit data dengan sendiri.

## 2. Admin

Dalam sistem ini hal yang dapat dilakukan oleh admin adalah sebagai berikut:

- a. Mengelola data siswa: Pada sistem ini admin dapat mengelola data siswa berupa melihat, menambah, mengedit, dan menghapus data siswa. Selain itu admin juga dapat mereset *password* siswa dengan mengirim *link* pada *email* siswa tersebut.
- b. Mengelola data guru: Pada sistem ini admin dapat mengelola data siswa berupa melihat, menambah, mengedit, dan menghapus data guru. Selain itu admin juga dapat mereset *password* siswa dengan mengirim *link* pada *email* guru tersebut.
- c. Mengelola *tag* soal: Pada sistem ini admin dapat mengelola *tag* soal berupa melihat, menambah, mengedit, dan menghapus *tag*.

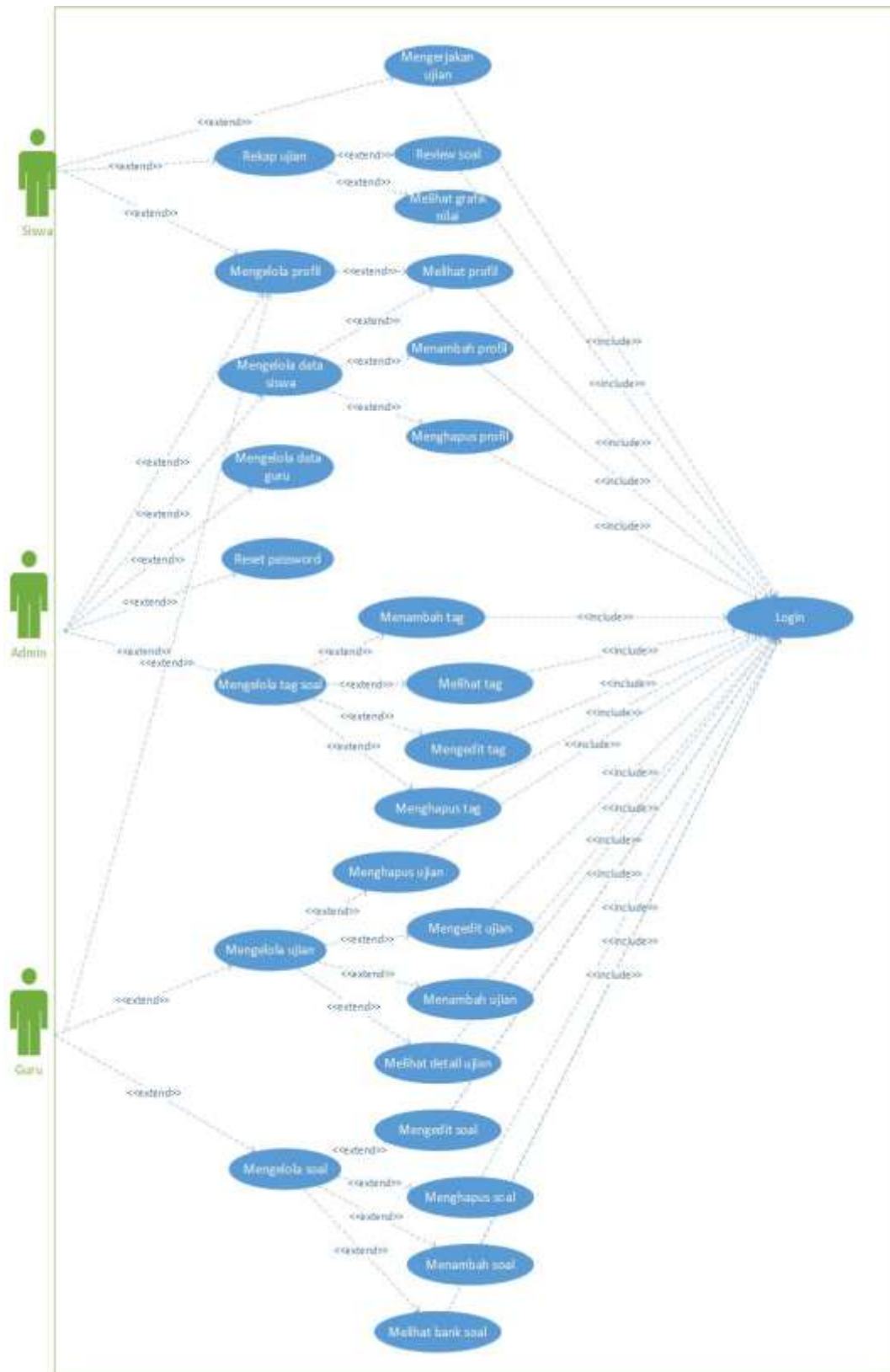
## 3. Guru

Dalam sistem ini hal yang dapat dilakukan oleh guru adalah sebagai berikut:

- a. Mengelola profil : Guru dapat mengelola profil dengan melihat dan mengedit profil yang berisi data diri.
- b. Mengelola soal : Guru dapat memasukkan, mengedit, dan menghapus soal sesuai dengan kebutuhan ujian yang akan dilaksanakan
- c. Mengelola ujian: Guru dapat melihat detail ujian, menambah dan mengedit ujian yang akan dilaksanakan.
- d. Rekap ujian: Guru dapat melihat langsung hasil ujian siswanya.

*Use case* diagram admin yang digunakan dalam sistem dapat dilihat pada **Gambar 3.4.**





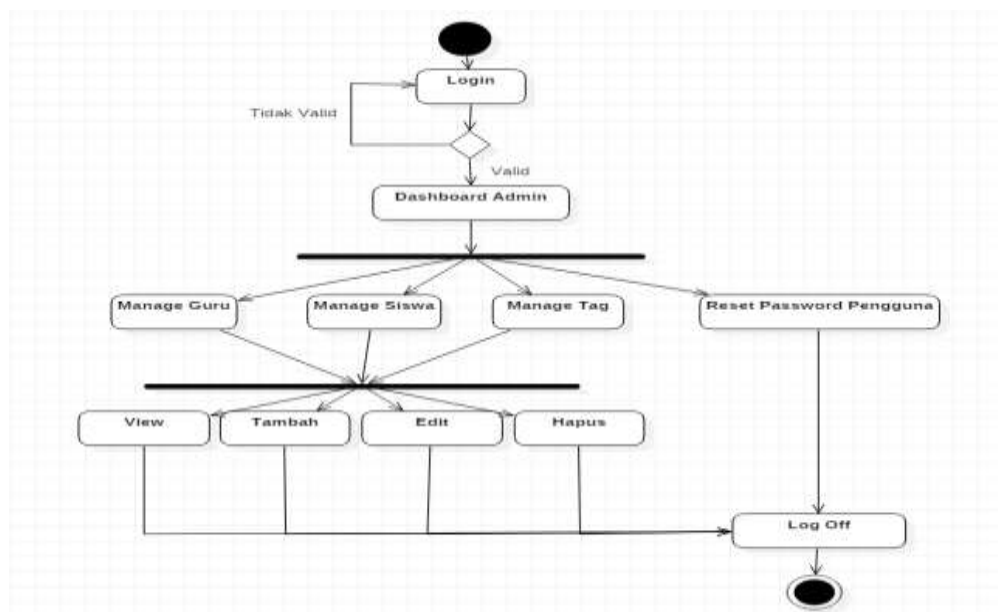
Gambar 3.4 Use Case Diagram Sistem

### 3.5.2 Activity Diagram

Berdasarkan *use case* yang telah dibuat, maka dapat diperoleh *Activity diagram* berdasarkan aktor yang terlibat pada *use case diagram*. *Activity diagram* untuk admin, menggambarkan beberapa proses yang harus dilewati ketika admin menggunakan aplikasi.

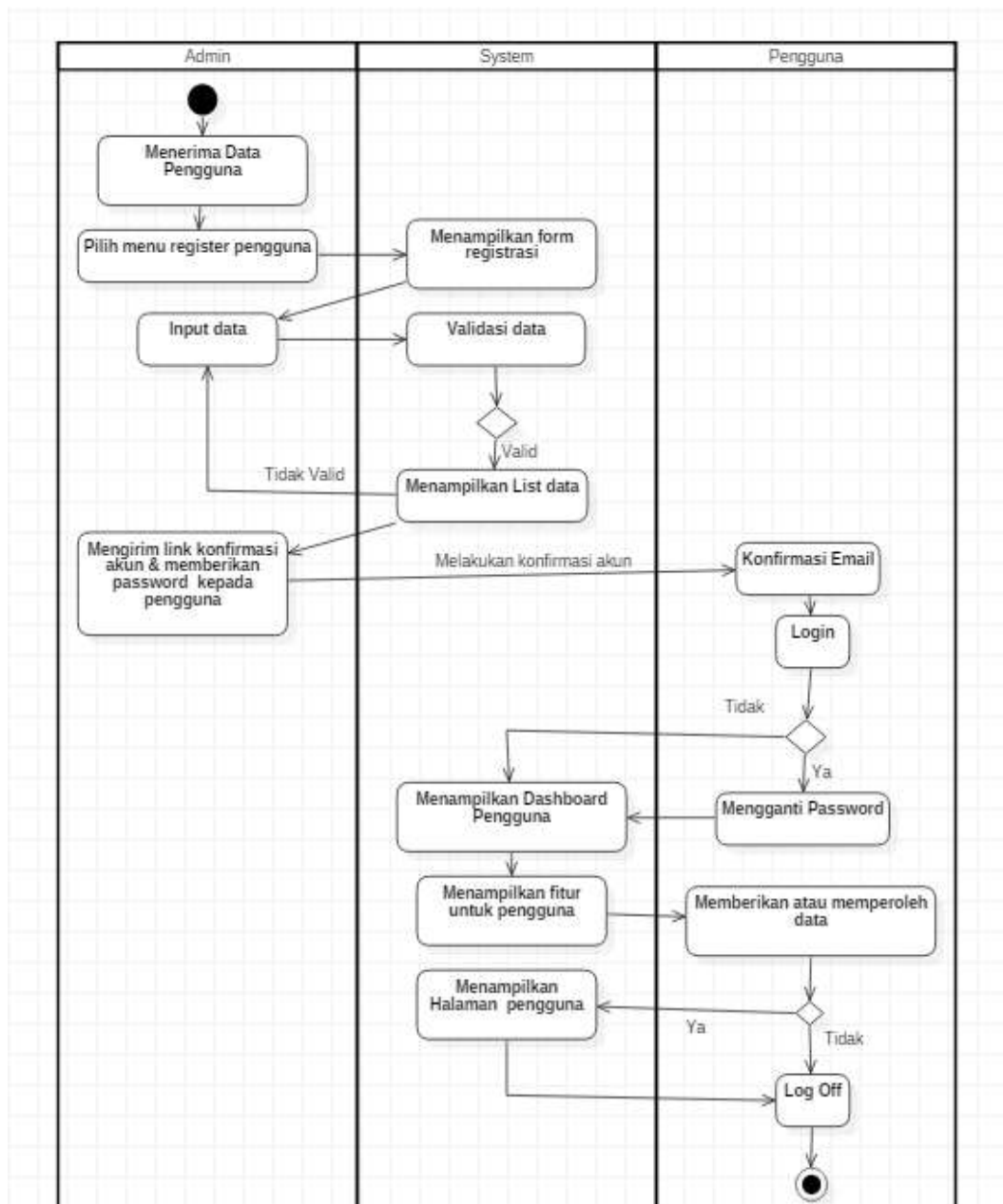
1. Sebelum dapat menggunakan aplikasi, admin diharuskan untuk melakukan *login* terlebih dahulu.
2. Apabila *login* berhasil, maka akan ditampilkan halaman *dashboard* admin, akan tetapi, apabila *login* gagal, admin akan diminta untuk memasukkan ulang *email* dan *password*.
3. Pada halaman *dashboard*, terdapat beberapa pilihan untuk pengolahan data yang dapat dilakukan admin, admin dapat memilih salah satu pilihan, dan melakukan pengolahan data seperti : lihat data, tambah data, edit data, dan hapus data.
4. Pada halaman *reset password*, admin akan diminta untuk memasukkan *email user* yang ingin melakukan *reset password*. Admin akan mengirimkan *link* untuk *reset password* ke alamat *email user*.

*Activity diagram* untuk admin dapat dilihat pada **Gambar 3.5**.



**Gambar 3.5** Activity Diagram Admin

Salah satu pengolahan data yang dilakukan admin adalah registrasi *user*. Langkah – langkah dalam registrasi *user* dapat dilihat pada **Gambar 3.6**.



**Gambar 3.6** Activity Diagram Register

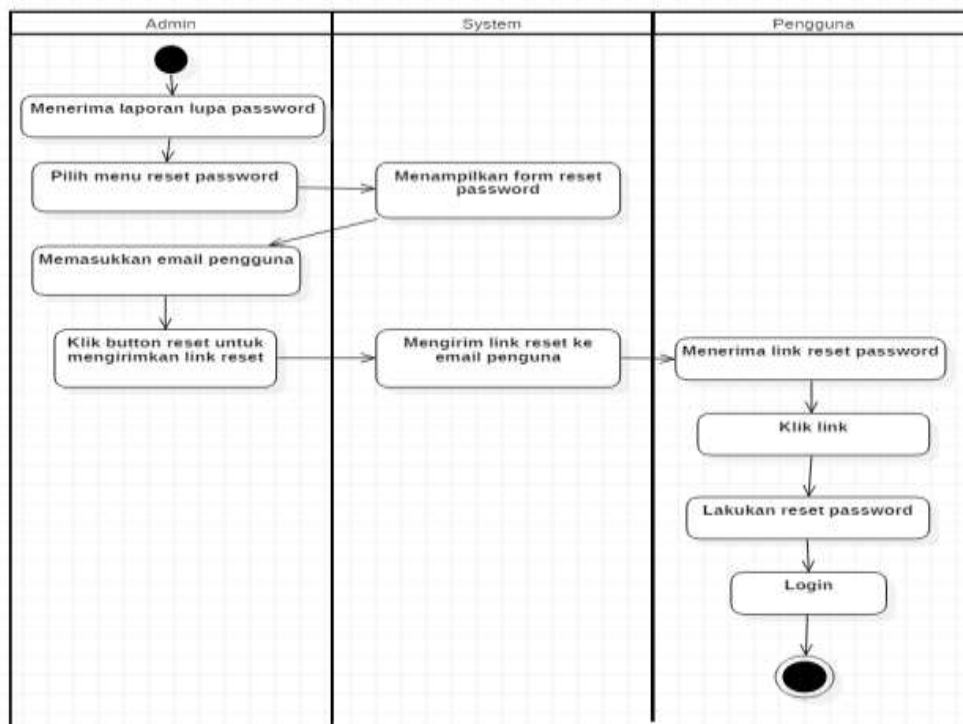
Langkah – langkah yang harus dilakukn admin untuk registrasi adalah:

1. Setelah menerima dan memeriksa data *user* secara lengkap, maka admin akan melakukan registrasi *user*. Admin akan memasukkan *email user* sebagai *username* dan membuatkan *password* untuk *user*.

*Password* yang telah dibuatkan oleh admin, dapat diganti oleh *user* setelah melakukan *login*.

2. Setelah admin mendaftarkan *user*, maka *user* harus melakukan konfirmasi akun menggunakan *link* yang telah dikirim admin ke *email user*.
3. Setelah melakukan konfirmasi akun, *user* dapat *login* dan menggunakan aplikasi.

Permasalahan yang dapat timbul selama penggunaan aplikasi adalah *user* mengalami lupa *password* sehingga tidak bisa *login*, oleh karena itu, admin mempunyai fitur *reset password*, yang langkah – langkahnya dapat dilihat pada



Gambar 3.7.

Berikut penjelasan terhadap langkah-langkah yang terdapat pada **Gambar 3.7**:

1. *User* yang mengalami lupa *password* harus melapor ke admin.
2. Admin akan mengirimkan *link* reset *password* ke *email user* yang telah dikonfirmasi sebelumnya pada saat pendaftaran.
3. *User* melakukan *reset password* dengan memasukkan *password* baru, melalui *link* yang dikirimkan oleh admin.

### 3.5.3 Entity Relationship Diagram (ERD)

Relasi antar *entity* pada aplikasi adalah sebagai berikut:

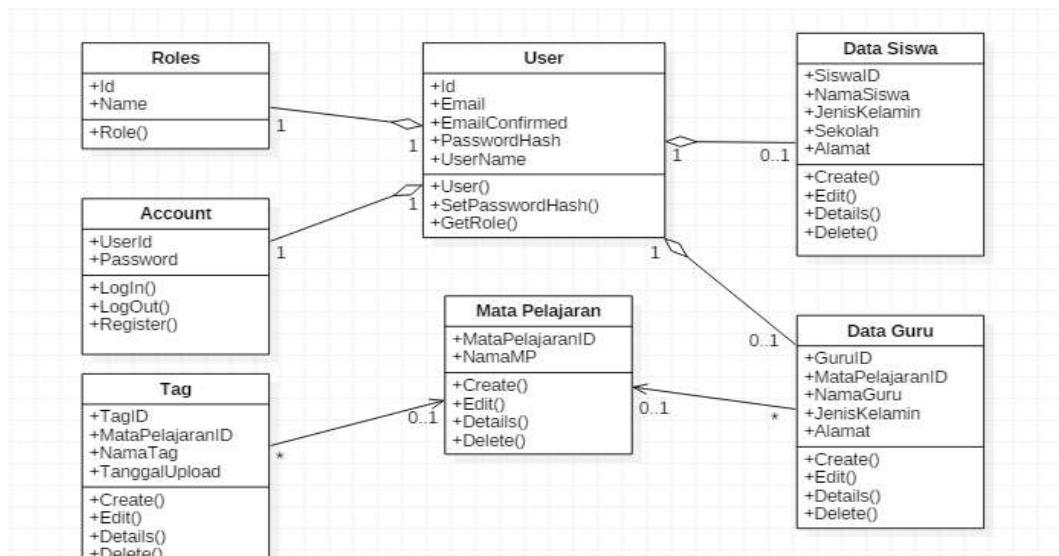
- a. *Entity* Guru mempunyai relasi *many to 1* terhadap *Entity* MataPelajaran.
- b. *Entity* Guru mempunyai relasi *1 to many* terhadap *Entity* StandarNilai.
- c. *Entity* Guru mempunyai relasi *1 to many* terhadap *Entity* Soal.
- d. *Entity* Ujian mempunyai relasi *many to many* dengan *Entity* 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.

Untuk mengetahui hubungan antar obyek (*Entity*) dalam basis data yang digunakan, dapat dilihat dari ERD pada **Gambar 3.8**.



### 3.5.4 Class Diagram

Rancangan *class diagram* yang akan digunakan dalam proses pembuatan aplikasi dapat dilihat pada **Gambar 3.9**.



**Gambar 3.9** Class Diagram untuk Admin

Berikut penjelasan Gambar 3.9:

1. Proses pembuatan aplikasi akan menggunakan *Class Asp.net MVC* untuk pengaturan *user*.
2. *Class Role* berfungsi untuk mengatur *role* apa saja yang dapat digunakan oleh *user* aplikasi. *Role* berperan penting dalam otorisasi *user* dalam penggunaan aplikasi.
3. *Class User* menampung *user* yang sudah didaftarkan, pada *class* ini terdapat method *PasswordHash()* yang bertujuan untuk meningkatkan keamanan akun seorang *user*, dan method *GetRole()* untuk memberikan *role* atau peran kepada seorang *user*.
4. *Class Account* berfungsi dalam pendaftaran *user* dengan method *Register()*, dan pengaturan hak akses *user*, seperti method *Login()* yang berfungsi untuk validasi *login* member.
5. *Class Data Guru* berfungsi untuk pengelolaan data guru, yang mencakup proses lihat data, tambah data, edit data, dan hapus data.

6. *Class Data Siswa* berfungsi untuk pengelolaan data siswa, yang mencakup proses lihat data, tambah data, edit data, dan hapus data.
7. *Class Tag* berfungsi untuk pengelolaan data *tag*, yang mencakup proses lihat data, tambah data, edit data, dan hapus data.
8. *Class Mata Pelajaran* berfungsi untuk menampung daftar mata pelajaran yang digunakan dalam aplikasi.

### 3.5.5 Rancangan Antarmuka

*Interface* atau antarmuka merupakan tampilan dari aplikasi yang menjadi sarana komunikasi antara *user* dengan sistem, serta memberikan kemudahan kepada *user* dalam melakukan aktivitas didalam penggunaan *web*.

#### 1. Rancangan Halaman *Login*

Rancangan antarmuka ini merupakan halaman validasi *email* dan *password* yang berfungsi untuk melakukan *login* kedalam aplikasi. Admin, Guru, dan Siswa melakukan *login* dengan menggunakan halaman yang sama. Namun, setelah berhasil *login* kedalam aplikasi, masing – masing *user* akan diarahkan ke halaman *dashboard* yang sesuai dengan *role* atau perannya. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.10**.

The diagram illustrates the layout of a login page. It features four main components arranged vertically on the left side, each with a corresponding label on the right side:

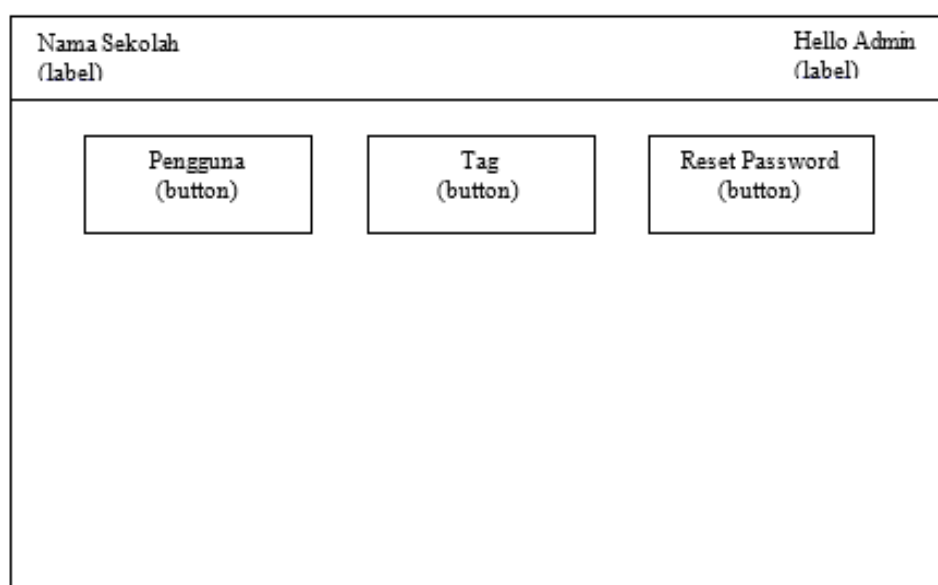
- Email (label)**: A text input field labeled "Kolom email".
- Password (label)**: A text input field labeled "Kolom password".
- Login (Buttom)**: A rectangular button labeled "Login (Buttom)".
- Lupa password? Silahkan hubungi admin (label)**: A text label at the bottom left.

**Gambar 3.10** Rancangan antarmuka halaman *login*



## 2. Rancangan Halaman *Dashboard* Admin

Rancangan antarmuka ini merupakan halaman *dashboard* yang akan tampil apabila *user* yang *login* berperan sebagai admin. Pada halaman ini terdapat fitur – fitur yang dapat digunakan oleh admin dalam melakukan pengelolaan data. Terdapat 3 fitur utama pada halaman ini, yaitu *Pengguna*, *Tag*, dan *Reset Password*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.11**.



**Gambar 3.11** Rancangan antarmuka halaman *Dashboard*

## 3. Rancangan Antarmuka Pengguna

Rancangan antarmuka ini merupakan halaman yang berisi daftar semua *user*. Pada halaman ini terdapat kolom untuk melakukan *search* yang memudahkan admin dalam mencari data *user*. Halaman ini juga dilengkapi dengan fitur *sorting* dan *paging*. Fitur *sorting* dapat digunakan admin untuk mengurutkan data dari yang terkecil sampai terbesar, atau mengurutkan sesuai abjad. Admin juga dapat mengatur jumlah data yang ingin ditampilkan pada setiap halaman. Pada halaman ini admin juga dapat mengetahui user yang sudah melakukan konfirmasi email, dan yang belum melakukan konfirmasi. Admin juga dapat menghapus user yang tidak lagi berhak menggunakan

sistem dengan tombol hapus yang juga terdapat pada halaman ini. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.12**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)	
Daftar Pengguna (Button)	Daftar Profil Guru (Button)	Daftar Profil Siswa (Button)			
Tambah Admin (label, icon)					
Show (Label)	Kolom Angka _____	Entries (Label)	Search (Label)		Kolom Search _____
Nama Pengguna (label)	Email (Label)	Konfirmasi Email? (Label)	Sekolah (Label)	Alamat (Label)	
					Hapus (Label)
				Previous (Label)	Next (Label)

**Gambar 3.12** Rancangan antarmuka halaman pengguna

#### 4. Rancangan Antarmuka *Tag*

Salah satu fitur unggulan aplikasi ini adalah melakukan pemetaan berbasis *tag* terhadap soal ujian. Fitur *tag* pada aplikasi ini adalah pengelompokan ujian siswa berdasarkan *tag* atau bab, sehingga siswa dapat melihat rekap nilai dengan lebih rinci dan dapat mengetahui materi ajar yang sudah dipahami ataupun belum. Karena bab pada satu materi ajar sangat banyak, sehingga diperlukan berbagai macam *tag* yang berkaitan dengan soal ujian. Tugas admin adalah mengelola *tag* yang digunakan. Rancangan antarmuka ini merupakan halaman Daftar *Tag*, pada halaman ini terdapat daftar semua *tag* yang digunakan pada aplikasi. Seperti halaman Daftar Siswa dan Daftar Guru, halaman ini juga dilengkapi dengan fitur *search*, *sorting*, dan *paging*. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.13**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)
Daftar Tag (Button)	Daftar Mata Pelajaran (Button)			
Tambah Tag (label, icon)				
Show (Label)	Kolom Angka _____	Entries (Label)	Search (Label)	Kolom Search _____
Mata Pelajaran (label)	Tag (Label)	Tanggal Upload (label)		
			Edit (Label)	Hapus (Label)
			Previous (Label)	Next (Label)

**Gambar 3.13** Rancangan antarmuka halaman pengguna

#### 5. Rancangan Antarmuka *Reset Password*.

Gambaran rancangan antarmuka halaman ini dapat dilihat pada

#### **Gambar 3.14.**

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)
Lupa Password? Masukkan Email Anda (label)				
Email (Label)		Kolom Email _____		
		Reset (Button)		

**Gambar 3.14** Rancangan antarmuka halaman *Reset Password*

Apabila siswa atau guru mengalami lupa *password* sehingga tidak bisa *login*, maka siswa atau guru tersebut harus melaporkannya kepada admin. Admin akan mengirimkan *link* untuk *reset password* ke alamat *email user* tersebut. Pada halaman ini, terdapat kolom *email* yang akan digunakan admin untuk memasukkan alamat *email user*.

#### 6. Rancangan Antarmuka Daftar Profil Guru

Rancangan antarmuka ini merupakan halaman Daftar Profil Guru yang berisi daftar semua guru yang menggunakan aplikasi. Pada halaman ini terdapat fitur *search*, *sorting*, dan *paging*. Pada halaman ini juga terdapat tombol Tambah Guru, yang akan mengarahkan admin ke halaman tambah guru, tombol Edit untuk mengarahkan ke halaman edit guru, Detail untuk melihat informasi guru yang lebih lengkap, dan tombol Hapus untuk menghapus guru. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.15**.

Nama Sekolah (logo)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)	
Daftar Pengguna (Button)	Daftar Profil Guru (Button)	Daftar Profil Siswa (Button)			
Tambah Guru (label, icon)					
Show (Label)	Kolom Angka	Entries (Label)	Search (Label)	Kolom Search	
Nama Guru (Label)	Mata Pelajaran (Label)	Sekolah (Label)	Edit (Label)	Detail (Label)	Hapus (Label)
				Previous (Label)	Next (Label)

**Gambar 3.15** Rancangan antarmuka halaman Daftar Profil Guru

#### 7. Rancangan Antarmuka Daftar Profil Siswa

Rancangan antarmuka ini merupakan halaman Daftar Siswa yang berisi daftar semua siswa yang menggunakan aplikasi. Pada halaman ini

terdapat kolom untuk melakukan *search* yang memudahkan admin dalam mencari data siswa. Halaman daftar siswa juga dilengkapi dengan fitur *sorting* dan *paging*. Fitur *sorting* dapat digunakan admin untuk mengurutkan data dari yang terkecil sampai terbesar, atau mengurutkan sesuai abjad. Admin juga dapat mengatur jumlah data yang ingin ditampilkan pada setiap halaman. Pada halaman ini juga terdapat tombol Tambah Siswa yang akan mengarahkan admin ke halaman tambah siswa. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.16**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)	
Daftar Pengguna (Button)	Daftar Profil Guru (Button)	Daftar Profil Siswa (Button)			
Tambah Siswa (label, icon)					
Show (Label)	Kolom Angka	Entries (Label)	Search( Label)	Kolom Search	
Nama Siswa (Label)	Sekolah (Label)	Kelas (Label)			
			Edit (Label)	Detail( Label)	Hapus( Label)
				Previous (Label)	Next (Label)

**Gambar 3.16** Rancangan antarmuka halaman Daftar Profil Siswa

## 8. Rancangan Antarmuka Halaman Tambah Admin

Halaman Tambah Admin merupakan halaman yang digunakan untuk mendaftarkan admin baru. Pada halaman ini, admin harus mengisi data-data seperti: *Email*, *Password*, dan Konfirmasi *Password* agar dapat mendaftarkan admin. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.17**

Nama Sekolah (label)	Hello Admin (label)
Email (Label)	Konten Email _____
Password (Label)	Konten Password _____
Konfirmasi Password (Label)	Konten Konfirmasi Password _____
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Daftar (button)</div>	

**Gambar 3.17** Rancangan antarmuka halaman Tambah Admin

#### 9. Rancangan Antarmuka Halaman Tambah Guru

Halaman Tambah Guru merupakan halaman yang digunakan admin untuk melakukan registrasi guru baru. Pada halaman ini terdapat data – data yang harus diisi oleh admin untuk dapat melakukan registrasi. Data- data yang telah diisikan akan ditampilkan pada profil guru. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.18**.

Nama Sekolah (label)	Hello Admin (label)
Nama Guru (Label)	Kolom Nama Guru _____
Jenis Kelamin (Label)	Kolom Jenis Kelamin _____
Mata Pelajaran (Label)	Kolom Mata Pelajaran _____
Alamat (Label)	Kolom Alamat _____
Email (Label)	Kolom Email _____
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Submit (Buttom)</div>	

**Gambar 3.18** Rancangan antarmuka halaman Tambah Guru

## 10. Rancangan Antarmuka Halaman Tambah Siswa

Halaman Tambah Siswa merupakan halaman yang digunakan admin untuk melakukan registrasi terhadap siswa. Pada halaman ini terdapat data – data yang harus diisi oleh admin untuk dapat melakukan registrasi. Data- data yang telah diisi akan ditampilkan pada profil siswa. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.19**.

Nama Sekolah (label)	Hello Admin (label)
Nama Siswa (Label)	Kolom Nama Siswa
Jenis Kelamin (Label)	Kolom Jenis Kelamin
Sekolah (Label)	Kolom Sekolah
Alamat (Label)	Kolom Alamat
Email (Label)	Kolom Email
	Submit (Buttom)

**Gambar 3.19** Rancangan antarmuka halaman Tambah Siswa

## 11. Rancangan Antarmuka Halaman Edit Profil Guru

Halaman Edit Profil Guru digunakan oleh admin untuk memperbaharui data profil guru yang sudah terdaftar. Setelah memperbaharui data guru, admin dapat menyimpan data tersebut dengan menekan tombol Edit. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.20**.

Nama Sekolah (label)	Hello Admin (label)
Nama Guru (Label)	Kolom Nama Guru _____
Jenis Kelamin (Label)	Kolom Jenis Kelamin _____
Mata Pelajaran (Label)	Kolom Mata Pelajaran _____
Alamat (Label)	Kolom Alamat _____
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Edit (Buttom)</div>	

**Gambar 3.20** Rancangan antarmuka halaman Edit Profil Guru

## 12. Rancangan Antarmuka Halaman Edit Profil Siswa

Halaman Edit Profil Siswa digunakan oleh admin untuk memperbaharui data profil siswa yang sudah terdaftar. Setelah memperbaharui data siswa, admin dapat menyimpan data tersebut dengan menekan tombol Edit. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.21**.

Nama Sekolah (label)	Hello Admin (label)
Nama Siswa (Label)	Kolom Nama Siswa _____
Jenis Kelamin (Label)	Kolom Jenis Kelamin _____
Sekolah (Label)	Kolom Sekolah _____
Alamat (Label)	Kolom Alamat _____
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Edit (Buttom)</div>	

**Gambar 3.21** Rancangan antarmuka halaman Edit Profil Siswa



### 13. Rancangan Antarmuka Halaman Detail Guru.

Pada halaman Detail Guru, akan ditampilkan data profil yang lebih lengkap dari guru yang sudah terdaftar. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.22**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)												
<table border="1"> <tr> <td>Nama Guru (Label)</td> <td>Kolom Nama Guru</td> </tr> <tr> <td>Jenis Kelamin (Label)</td> <td>Kolom Jenis kelamin</td> </tr> <tr> <td>Mata Pelajaran (Label)</td> <td>Kolom Mata Pelajaran</td> </tr> <tr> <td>Alamat (Label)</td> <td>Kolom Alamat</td> </tr> <tr> <td>Email (Label)</td> <td>Kolom Email</td> </tr> <tr> <td>Edit (Label)</td> <td>Back (Label)</td> </tr> </table>					Nama Guru (Label)	Kolom Nama Guru	Jenis Kelamin (Label)	Kolom Jenis kelamin	Mata Pelajaran (Label)	Kolom Mata Pelajaran	Alamat (Label)	Kolom Alamat	Email (Label)	Kolom Email	Edit (Label)	Back (Label)
Nama Guru (Label)	Kolom Nama Guru															
Jenis Kelamin (Label)	Kolom Jenis kelamin															
Mata Pelajaran (Label)	Kolom Mata Pelajaran															
Alamat (Label)	Kolom Alamat															
Email (Label)	Kolom Email															
Edit (Label)	Back (Label)															

**Gambar 3.22** Rancangan antarmuka halaman Detail Guru

### 14. Rancangan Antarmuka Halaman Detail Siswa

Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3. 23**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)												
<table border="1"> <tr> <td>Nama Siswa (Label)</td> <td>Kolom Nama Siswa</td> </tr> <tr> <td>Jenis Kelamin (Label)</td> <td>Kolom Jenis Kelamin</td> </tr> <tr> <td>Sekolah (Label)</td> <td>Kolom Sekolah</td> </tr> <tr> <td>Alamat (Label)</td> <td>Kolom Alamat</td> </tr> <tr> <td>Email (Label)</td> <td>Kolom Email</td> </tr> <tr> <td>Edit (Label)</td> <td>Back (Label)</td> </tr> </table>					Nama Siswa (Label)	Kolom Nama Siswa	Jenis Kelamin (Label)	Kolom Jenis Kelamin	Sekolah (Label)	Kolom Sekolah	Alamat (Label)	Kolom Alamat	Email (Label)	Kolom Email	Edit (Label)	Back (Label)
Nama Siswa (Label)	Kolom Nama Siswa															
Jenis Kelamin (Label)	Kolom Jenis Kelamin															
Sekolah (Label)	Kolom Sekolah															
Alamat (Label)	Kolom Alamat															
Email (Label)	Kolom Email															
Edit (Label)	Back (Label)															

**Gambar 3.23** Rancangan antarmuka halaman Detail Siswa

Pada halaman Detail Siswa akan ditampilkan data profil yang lebih lengkap dari siswa yang sudah terdaftar.

#### 15. Rancangan Antarmuka Halaman Hapus Guru

Pada halaman Hapus Guru admin dapat menghapus data guru yang tidak lagi menggunakan aplikasi. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.24**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Riset Password (label)	Hello Admin (label)
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p>Anda yakin ingin menghapus data ini ? (label)</p> <p>Nama Siswa (Label)      <u>Label Nama Guru</u></p> <p>Jenis Kelamin (Label)      <u>Label Jenis Kelamin</u></p> <p>Mata Pelajaran (Label)      <u>Label Mata Pelajaran</u></p> <p>Alamat (Label)      <u>Label Mata Alamat</u></p> <p>Email (Label)      <u>Label Email</u></p> <p>Delete (Label)             Back (Label)</p> </div>				

**Gambar 3.24** Rancangan antarmuka halaman Hapus Guru

#### 16. Rancangan Antarmuka Halaman Hapus Siswa.

Aplikasi ini hanya digunakan oleh siswa kelas XII SMA, sehingga data siswa selalu berubah setiap tahunnya. Rancangan antarmuka ini merupakan halaman Hapus yang berfungsi untuk menghapus data siswa yang tidak lagi menggunakan aplikasi. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.25**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)
<p>Anda yakin ingin menghapus data guru ini ? (label)</p> <p>Nama Siswa (Label)      Label Nama Guru</p> <p>Jenis Kelamin (Label)      Label Jenis Kelamin</p> <p>Sekolah (Label)      Label Sekolah</p> <p>Alamat (Label)      Label Alamat</p> <p>Email (Label)      Label Email</p> <p>Delete (Label)      Back (Label)</p>				

**Gambar 3.25** Rancangan antarmuka halaman Hapus Siswa  
17. Rancangan Antarmuka Halaman Tambah *Tag*.

Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.26**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)
<p>Mata Pelajaran (Label)      Kolom Mata Pelajaran</p> <p>Tag (Label)      Kolom Tag</p> <p>Tanggal Upload (Label)      Kolom Tanggal</p> <p>Submit (Buttom)</p> <p>Back to list (Label)</p>				

**Gambar 3.26** Rancangan antarmuka halaman Tambah *Tag*

Rancangan antarmuka pada **Gambar 3.26** adalah halaman Tambah *Tag*. Pada halaman ini, admin harus memasukkan nama *tag* dan tanggal diunggahnya *tag*.

#### 18. Rancangan Antarmuka Halaman Edit *Tag*.

Sama halnya dengan halaman Edit Siswa dan Edit Guru , halaman ini digunakan oleh admin untuk memperbaharui *tag* yang sudah terdaftar.

Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.27**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)
<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>Mata Pelajaran (Label)</p> <p>Tag (Label)</p> <p>Tanggal Upload (Label)</p> </div> <div style="text-align: center;"> <p>Kolom Mata Pelajaran</p> <hr/> <p>Kolom Tag</p> <hr/> <p>Kolom Tanggal</p> <hr/> <div style="border: 1px solid black; padding: 5px; display: inline-block;">Edit (Buttom)</div> </div> </div> <p style="margin-top: 20px;">Back to list (Label)</p>				

**Gambar 3.27** Rancangan antarmuka halaman Edit *Tag*

#### 19. Rancangan Antarmuka Halaman Hapus *Tag*.

Halaman ini akan muncul saat *user* menekan tombol hapus pada halaman daftar *tag*. Pada halaman ini admin dapat menghapus *tag* yang tidak lagi digunakan. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.28**.

Nama Sekolah (label)	Pengguna (label)	Tag (label)	Reset Password (label)	Hello Admin (label)
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p>Anda yakin ingin menghapus tag ini ? (label)</p> <div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> <p>Mata Pelajaran (Label)</p> <p>Tag (Label)</p> <p>Tanggal Upload (Label)</p> </div> <div style="text-align: center;"> <p>List Mata Pelajaran</p> <hr/> <p>List Tag</p> <hr/> <p>List Tanggal</p> <hr/> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <span>Delete (Label)</span> <span>Back (Label)</span> </div> </div>				

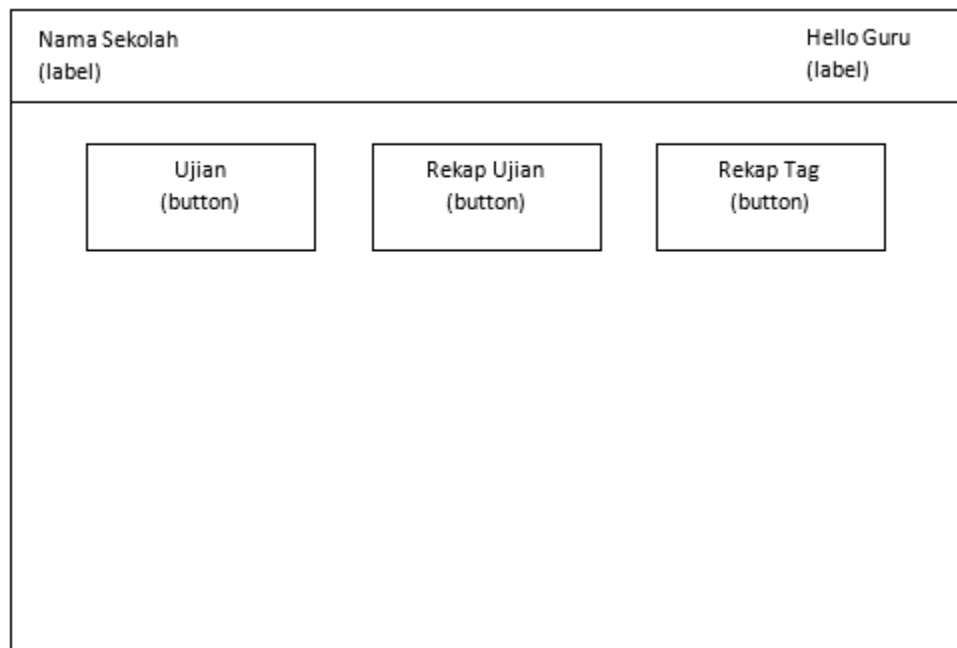
**Gambar 3.28** Rancangan antarmuka halaman Hapus *Tag*

## 20. Halaman *Dashboard* Guru

Halaman ini merupakan halaman *dashboard* yang akan tampil apabila *user* yang *login* berperan sebagai guru. Pada halaman ini terdapat fitur yang dapat digunakan guru seperti:

- a. Profil, pada saat pertama kali membuka *dashboard* guru maka user dapat melihat profil yang berisi data pribadi.
- b. Ujian, Fitur ini akan menghubungkan guru ke halaman ujian sehingga guru dapat menyediakan ujian untuk siswa.
- c. Rekap Ujian, fitur ini akan menampilkan rekap ujian secara keseluruhan.
- d. Rekap *Tag*, fitur ini akan menampilkan rekap ujian sesuai dengan *tag* atau bab.

Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.29**.

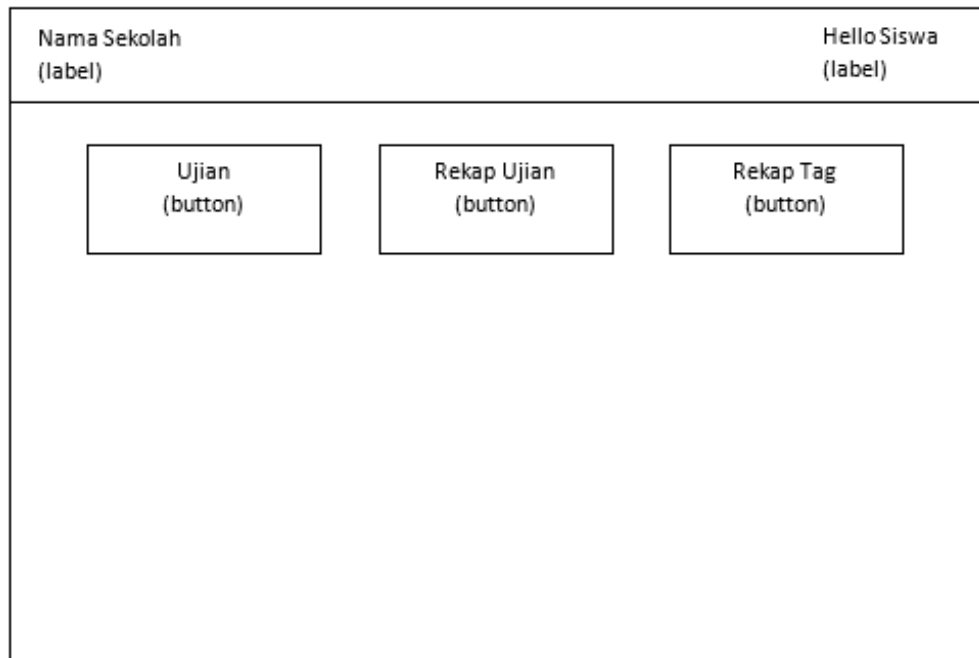


**Gambar 3.29** Rancangan antarmuka *Dashboard* Guru

## 21. Halaman *Dashboard* Siswa

Halaman ini merupakan halaman *dashboard* yang akan tampil apabila *user* yang *login* berperan sebagai siswa. Berikut ini adalah keterangan dari **Gambar 3.30**.

- a. Profil, pada saat pertama kali membuka *dashboard* siswa maka user dapat melihat profil yang berisi data pribadi.
- b. Ujian, Fitur ini akan menghubungkan siswa ke halaman ujian sehingga siswa dapat melakukan ujian yang sudah disediakan.
- c. Rekap Ujian, fitur ini akan menampilkan rekap ujian secara keseluruhan.
- d. Rekap *Tag*, fitur ini akan menampilkan rekap ujian sesuai dengan *tag* atau bab.



**Gambar 3.30** Rancangan antarmuka *Dashboard* Siswa