

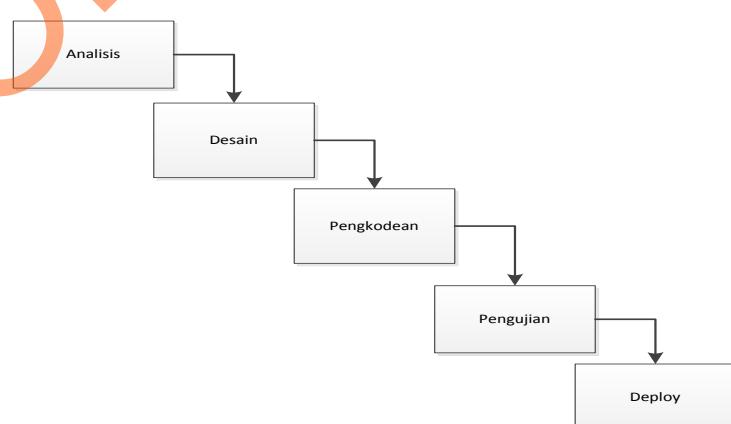
BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa sistem

Pada bab ini akan dibahas mengenai perancangan sistem yang akan dibangun. Untuk membangun sistem dilakukanlah analisa sistem yang ada pada saat ini untuk mengetahui kekurangan sistem yang ada pada saat ini. Selanjutnya dari analisa kekurangan sistem tersebut digunakan untuk membangun sistem yang baru. Pada pengembangan sistem terdapat banyak model pengembangan sistem salah satunya adalah model *waterfall*. Pada pengembangan sistem ini akan menggunakan model *waterfall*.

Menurut Sholiq(2010:33) pada model pengembangan perangkat lunak dengan *waterfall*, ada beberapa langkah yang dilakukan antara lain: (i) analisis kebutuhan, (ii) desain sistem, (iii) pengkodean sistem, (iv) pengujian sistem, dan (v) *deploy* sistem. Model pengembangan *waterfall* dapat digambarkan dengan diagram pada gambar 3.1 berikut ini.



Gambar 3.1 Waterfall Model

3.1.1 Identifikasi Masalah

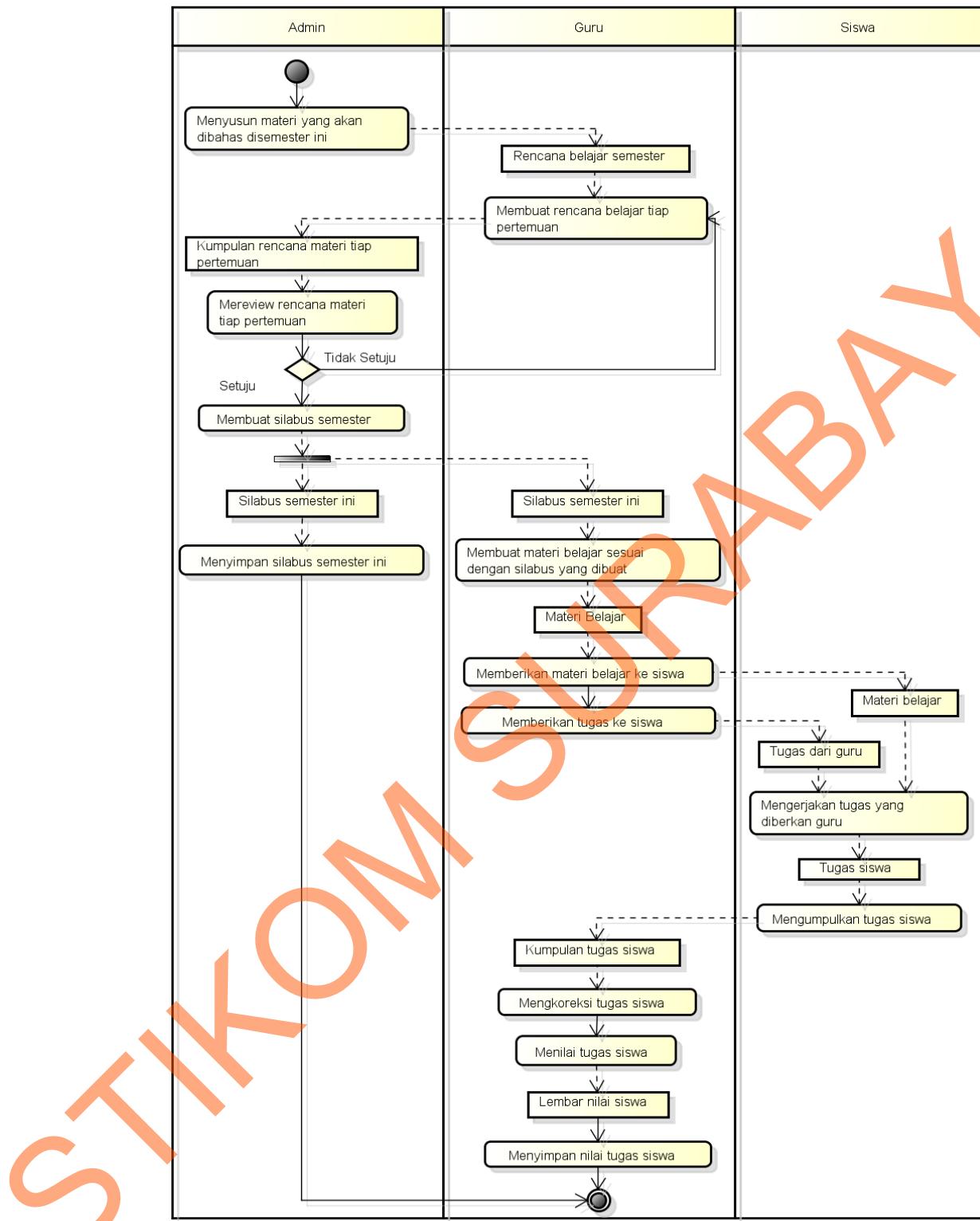
Untuk melakukan identifikasi masalah maka dilakukan observasi pada laboratorium komputer SD Al-Furqon Jember. Observasi ini dilakukan untuk menggali informasi dan menganalisa sistem yang ada pada laboratorium komputer SD Al-Furqon dengan mewawancarai kepala bagian laboratorium komputer mulai tanggal 1 sampai 30 November 2012. Pada tahapan ini informasi yang dikumpulkan mulai dari sistem pembelajaran di laboratorium komputer pada saat ini, cara mendistribusikan materi, pemberian tugas, dan penyimpanan data nilai tugas siswa, serta pembuatan silabus pada awal semester.

Pada saat ini proses belajar mengajar pada pelajaran TIK dimulai dengan memberikan teori pada separuh sesi mata pelajaran kemudian dilanjutkan dengan pemberian latihan dan tugas mandiri. Untuk mendapatkan *file* tugas dan materi belajar tersebut para siswa dapat meminta secara langsung kepada guru yang ada di kelas. Siswa mengalami kesulitan dalam mendapatkan materi belajar dikarenakan tidak ada fitur untuk *sharing materi*. Mereka mendapatkan materi dengan cara meng-copy *file* secara langsung ke guru yang ada di kelas dan meng-copy-kan materi yang telah di-copy ke dalam komputer mereka.

Pada saat menggunakan komputer para murid dapat secara langsung melihat *file-file* yang telah ada atau dikerjakan oleh anak kelas lain. Kesempatan tersebut dapat memungkinkan murid untuk menduplikasi dan menghapus tugas yang telah dikerjakan. Pada saat pengumpulan tugas murid meng-copy *file* dan mengumpulkan ke guru masing-masing, sehingga guru harus menata ulang *file* tugas para murid tersebut. Guru melakukan koreksi dan evaluasi tugas siswa tersebut di komputer guru. Namun terkadang ada guru yang terlebih dahulu

mencetak tugas-tugas muridnya kemudian melakukan koreksi. Selanjutnya nilai dicatat oleh guru pada lembar penilaian dan hasil kerja dibagikan ke masing-masing siswa. Lembar - lembar nilai tersebut rawan hilang pada saat disimpan oleh guru. Dari lembar pencatatan nilai tersebut guru tidak dapat melihat perkembangan nilai dari siswa tersebut.

Pada proses pembuatan rencana belajar guru dan koordinator guru merapatkan materi belajar yang akan digunakan untuk semester yang akan berlangsung. Selanjutnya guru-guru akan membuat rencana materi belajar tiap mata pelajaran per pertemuan. Laporan materi belajar tiap pertemuan tersebut akan disetujui oleh koordinator selanjutnya setelah semua rencana materi belajar terkumpul maka dibuatlah silabus pembelajaran semester ini. Selanjutnya silabus tersebut akan disimpan oleh koordinator dan masing-masing guru yang bersangkutan. Namun ketika semester berjalan materi-materi yang telah dibuat oleh guru pada tiap-tiap minggunya tidak teratur penempatan lokasi *file* materinya dan terkadang guru tidak membuat materi, sesuai dengan yang direncanakan di awal semester. Dari beberapa proses bisnis tersebut dapat digambarkan menjadi *activity diagram* keseluruhan untuk sistem yang ada saat ini seperti pada gambar 3.2 berikut ini.



Gambar 3.2 *Activity Diagram Learning Management System pada Laboratorium komputer SD Al-Furqon yang digunakan saat ini.*

Dari hasil identifikasi diatas, maka dapat disimpulkan bahwa laboratorium SD Al-Furqon membutuhkan sistem yang mampu mengatur materi-materi pembelajaran. Sistem harus mampu mengatur hak akses dari murid dan guru pada saat mengakses materi-materi pembelajaran. Sistem juga harus mampu menyediakan tempat bagi siswa dalam mengunduh materi pembelajaran dan hasil tugas siswa pada pertemuan sebelumnya yang telah disediakan oleh guru. Sistem juga dapat menyimpan data nilai tugas siswa. Sistem juga harus mampu mencatat dan menyimpan silabus pembelajaran pada tahap perencanaan silabus diawal semester agar materi – materi yang *di-upload* oleh guru sesuai dengan silabus. Dengan adanya sistem tersebut pada laboratorium komputer SD Al-Furqon diharapkan dapat mengatur materi belajar, membuat silabus pembelajaran, menampilkan perkembangan nilai siswa, dan menyediakan tempat bagi guru untuk sharing materi belajar .

3.1.2 Hasil Analisis

Dari hasil analisis permasalahan , didapatkan kelemahan - kelemahan dari sistem yang lama dan untuk mengatasi kelemahan – kelemahan tersebut maka akan dibuat sistem yang dapat menangani permasalahan dan sesuai dengan kebutuhan pihak laboratorium komputer SD Al-Furqon. Hasil identifikasi masalah pada laboratorium komputer SD Al-Furqon sebagai berikut:

A. Kekurangan sistem

1. Materi yang disimpan pada tiap-tiap komputer siswa tidak memiliki penempatan *file* yang teratur.

2. *File* tugas dan materi belajar dapat diakses oleh siswa lain dikarenakan pemakaian komputer secara bersama tanpa ada batasan hak akses dalam penggunaan materi.
3. Penyimpanan nilai siswa yang konvensional membuat guru tidak bisa melihat informasi perkembangan nilai siswa.
4. Siswa kesulitan dalam mendapatkan materi belajar.
5. Pembuatan silabus pembelajaran diawal semester yang dilakukan secara konvensional membuat setiap pertemuan minggunya membuat lokasi penempatan *file* tidak teratur dan guru mempunyai kesempatan untuk tidak membuat materi belajar.

B. Kebutuhan pemakai

Kebutuhan dari pengguna yaitu siswa dan guru pada laboratorium komputer SD-Al-Furqon adalah sebagai berikut :

1. Sistem dapat mengatur materi-materi belajar.
2. Sistem dapat mengatur hak akses dari tiap pengguna pada saat mengakses materi belajar.
3. Sistem dapat menyediakan tempat bagi guru untuk mendistribusikan *file* tugas kepada murid.
4. Sistem dapat menyimpan data nilai tugas dari para siswa.
5. Guru dapat membuat rencana pembelajaran dan menjadi *template* bagi guru untuk menunggah materi.
6. Sistem dapat menampilkan informasi perkembangan nilai tugas murid.

7. Sistem dapat menampilkan informasi tanggungan materi guru yang tidak didistribusikan.

3.2 Melakukan Studi Literatur

Studi literatur tersebut dilakukan untuk mendapatkan landasan teori tentang pengembangan aplikasi *learning management sistem* berbasis *android* agar pada pelaksanaan pengembangan mendapatkan acuan yang benar dan dapat memenuhi tujuan dari pengembangan sistem tersebut. Landasan teori tersebut penulis dapatkan dari membaca buku dan jurnal ilmiah mengenai pengembangan *learning management sistem* berbasis *android*. Hasil dari studi literatur yang dilakukan penulis adalah mendapatkan landasan teori tentang apa saja yang dibutuhkan dalam pembuatan sistem tersebut meliputi *learning management system*, android, Mysql, JqueryMobile, dan PhoneGap. Berikut beberapa landasan teori yang akan digunakan.

A. Learning management system

Menurut Ali (2011:3) *Learning Management System* adalah sebuah sistem yang didesain untuk menyajikan,melacak,melaporkan,dan mengatur konten pembelajaran, kemajuan siswa dan interaksi siswa. Menurut Rahman, dkk (2011:75) *Learning Management System* menyediakan *platform* untuk konten yang terintegrasi, memberikan materi ke pengguna, mendistribusikan, manajemen pembelajaran yang dapat di akses oleh beberapa pengguna antara lain, pelajar, pembuat konten, dan administrator.

Klasifikasi pengguna dari *Learning Management System*:

1. Peserta didik yang menggunakan *Learning Management System* dapat mengikuti proses belajar mengajar secara jarak jauh atau di tempat dan waktu yang sesuai dengan jadwal.
2. Pengajar dan asisten pengajar, menjadi guru yang menggunakan *Learning Management System* untuk mengajar,mengawasi,membantu dan mengevaluasi murid.
3. Administrator sistem adalah orang yang berusaha mendukung semua pengguna dari *learning management system* dan menjaga kelayakan dari operasional tersebut.

B. Android

Menurut Bhawiyuga, dkk (2011:2) *Android* adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli *Android* Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *android*, dibentuklah Open Handset Alliance konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi. Sistem perangkat lunak *android* sebenarnya merupakan sebuah stack atau tumpukan yang terdiri atas beberapa lapisan dari yang paling dekat dengan perangkat keras sampai berinteraksi langsung dengan pengguna.

C. Mysql

Menurut Nugroho (2005:181). Adalah sebuah perangkat sistem manajemen basis data SQL atau yang dikenal dengan DBMS(*Database management system*) yang dapat *multithread* dan *multi-user*.

Sebagai *database server*, MySQL dapat dikatakan lebih unggul daripada *database server* lainnya, terutama dalam kecepatan. Berikut ini keunggulan MySQL antara lain:

1. Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server dan solaris.

2. Multiuser

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

3. Security

MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta *password* terenkripsi.

4. Scalability dan limit

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris.

Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

D. JqueryMobile

Menurut Reid (2011:1) Jquery mobile adalah seperangkat jquery *plug-in* dan widget yang bertujuan untuk menyediakan *cross-platform API* untuk membuat *mobile web applications*. Pada penggunaanya jquery mobile mirip dengan jquery UI, namun untuk Jquery ui difokuskan pada *desktop applications*, jquery mobile dibangun untuk *mobile applications*.

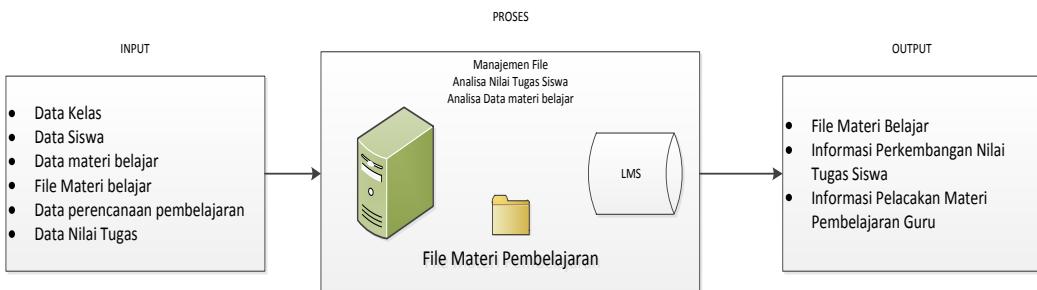
E. PhoneGap

Menurut Realitha (2012:2) phonegap adalah *cross platform mobile framework* , yaitu alat pengembang *open source* untuk membuat aplikasi *mobile* dengan menggunakan javascript. Phonegap dapat digunakan untuk membuat aplikasi *mobile* pada perangkat iPhone, Android, BlackBerry, Symbian dan Windows phone dengan memanfaatkan SDK masing-masing *mobile platform* tersebut.

Dari beberapa teori tersebut merupakan elemen penting dalam pembuatan sistem tersebut. Pembuataan sistem yang akan dibangun mengikuti kriteria LMS sesuai dengan landasan teori dan dijalankan di sistem operasi *android*. Dalam pembuatannya penulis memanfaatkan Mysql sebagai tempat penyimpanan data data yang digunakan oleh sistem LMS. Untuk aplikasi penulis menggunakan JqueryMobile dan PhoneGap untuk membuat aplikasi tersebut dapat berjalan di *platform android*. Dari pemahaman teori-teori yang mendukung dalam pembuatan *learning management system* tersebut diharapkan aplikasi yang dihasilkan tidak melenceng jauh dari pengertian *learning management system*.

3.3 Perancangan Sistem

Tahap perancangan sistem merupakan tahap pengembangan setalah melakukan analisis sistem. *Learning management System* (LMS) yang akan dibangun merupakan LMS yang melibatkan 3 pengguna yaitu guru, siswa, dan admin. Pada pengembangan LMS tersebut membutuhkan beberapa data seperti terlihat pada blok diagram di bawah ini.



Gambar 3.3 Blok diagram LMS

Dari blok diagram pada gambar 3.7 dapat memberikan informasi tujuan dibuatnya sistem LMS ini dengan rincian peran dan tanggung jawab pengguna sistem sebagai berikut:

1. Guru

Pada sistem LMS guru berperan sebagai pengajar dan melakukan proses memberikan materi belajar, memberikan tugas belajar, melakukan penilaian tugas siswa. Dari proses tersebut guru pada sistem LMS ini menggunakan dan mengelolah data dan *file* materi belajar, data nilai tugas dari siswa. Guru juga dapat melihat perkembangan nilai tugas dari siswa.

2. Murid

Pada sistem LMS murid memiliki kegiatan belajar dan mengerjakan tugas yang diberikan oleh guru. Untuk belajar murid dapat mengunduh materi belajar yang telah disediakan oleh guru. Dari proses tersebut murid pada sistem LMS ini menggunakan *file* materi belajar dan *file* tugas.

3. Administrator

Untuk Administrator pada sistem LMS ini dapat mengelolah data siswa, data kelas dan membuat perencanaan belajar pada satu semester. Administrator

juga dapat melihat informasi guru yang tidak memberikan materi sesuai dengan perencanaan silabus pada tiap minggunya.

3.3.1 Model Pengembangan Sistem

Pada model pengembangan sistem ini dimulai dengan mengumpulkan beberapa data yang digunakan sebagai *input-an* dari sistem. Data data tersebut meliputi Data kelas,data siswa, data silabus, data materi belajar dan file materi belajar selanjutnya data tersebut akan diproses untuk menghasilkan informasi sesuai dengan tujuan pengembangan sistem.

Sesuai dengan teori tentang LMS menurut Ali (2011) menyebutkan bahwa *Learning Management System* adalah sebuah sistem yang didesain untuk menyajikan, melacak, melaporkan, dan mengatur konten pembelajaran, kemajuan siswa dan interaksi siswa. Selanjutnya fitur aplikasi yang akan dikembangkan memiliki fungsi – fungsi seperti:

1. Sistem dapat mengatur materi-materi belajar sesuai dengan fungsi LMS mengatur konten pembelajaran
2. Sistem dapat mengatur hak akses dari tiap pengguna pada saat mengakses materi belajar.
3. Sistem dapat menyediakan tempat bagi guru untuk mendistribusikan file tugas kepada murid sesuai dengan fungsi LMS menyajikan materi.
4. Sistem dapat menyimpan data nilai tugas dari para siswa.
5. Guru dapat membuat rencana pembelajaran dan menjadi *template* bagi guru untuk menunggah materi.
6. Sistem dapat menampilkan informasi perkembangan nilai tugas murid sesuai dengan fungsi LMS melaporakan dan melihat kemajuan siswa.

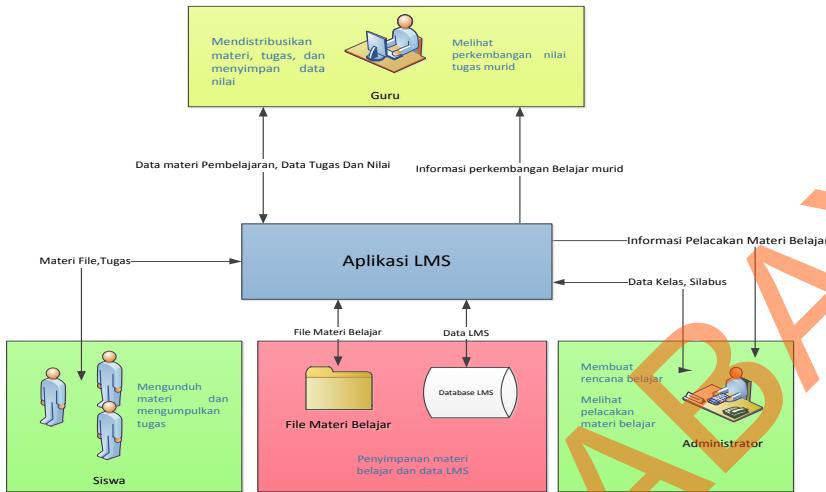
7. Sistem dapat menampilkan informasi tanggungan materi guru yang tidak didistribusikan sesuai dengan fungsi melihat tanggungan materi guru.

Pada fitur LMS yang akan dikembangkan juga terintegrasi dengan aplikasi pendukung lainnya seperti word, excel, power point, aplikasi menggambar dan aplikasi game cerdas. Pengguna nantinya akan dapat membuka materi belajar tanpa harus keluar dari aplikasi LMS. Dari penjelasan tersebut dapat diberikan tabel peran dan fungsi pengguna seperti pada tabel 3.1 peran dan tanggung jawab pengguna.

Tabel 3.1 Peran dan Tanggung jawab pengguna

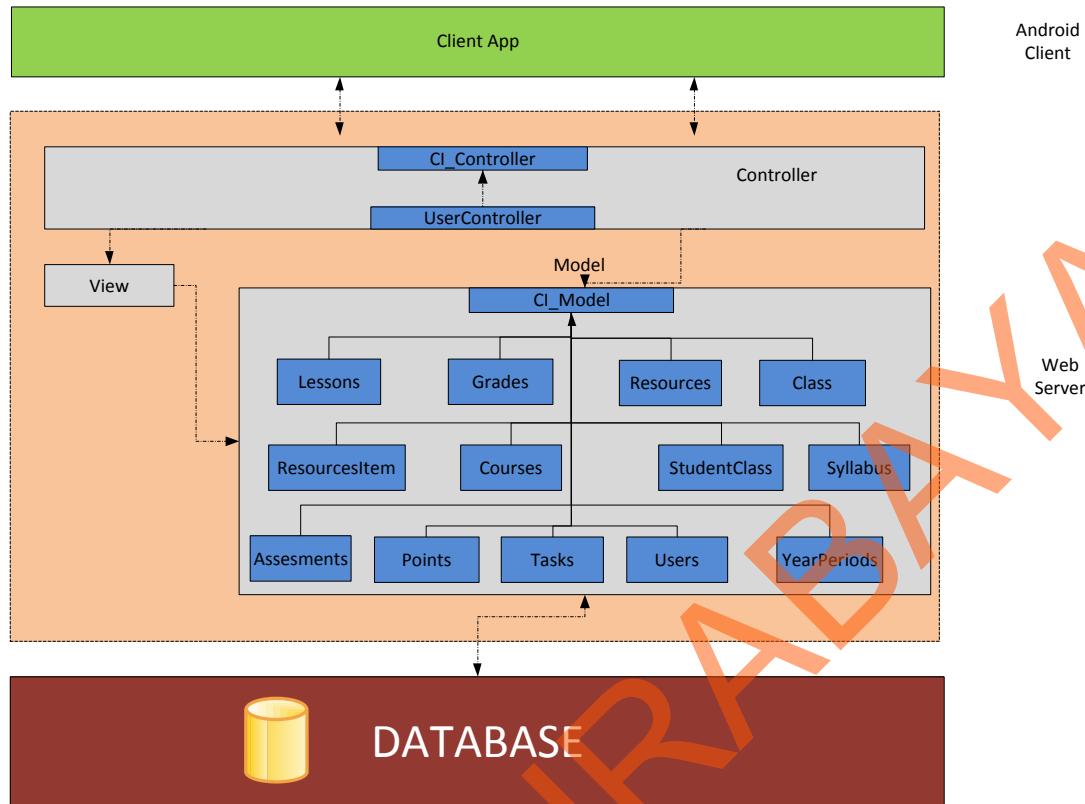
Pengguna	Tanggung jawab	Fungsi
Guru	Memberikan materi dan tugas belajar pada kegiatan belajar mengajar di tiap minggunya	Sistem dapat melakukan sharing materi dan menempatkannya pada direktori yang disediakan sesuai silabus
	Mengelolah data nilai tugas siswa	Sistem dapat melakukan pengolahan data nilai tugas, menampilkan informasi perkembangan nilai tugas siswa
Murid	Mendapatkan materi sesuai dengan mata pelajaran dan kelas	Sistem dapat menyediakan tempat untuk mengunduh materi belajar
	Mengerjakan tugas yang diberikan oleh guru	Sistem dapat mengunduh dan menunggu tugas siswa
	Mendapatkan tugas yang telah dikerjakan sebagai bahan refensi	Sistem dapat mengunduh tugas siswa yang telah di kerjakan
Administrator	Mengelolah data kelas	Sistem dapat mengelolah data kelas
	Membuat rencana belajar	Sistem dapat mengelolah data rencana belajar
	Mengontrol materi belajar tiap minggunya	Sistem dapat memberikan informasi pelackan materi belajar

Dari peran dan fungsi pengguna diatas maka dapat dibuat model pengembangan sistem seperti pada gambar



Gambar 3.8 Model Pengembangan LMS

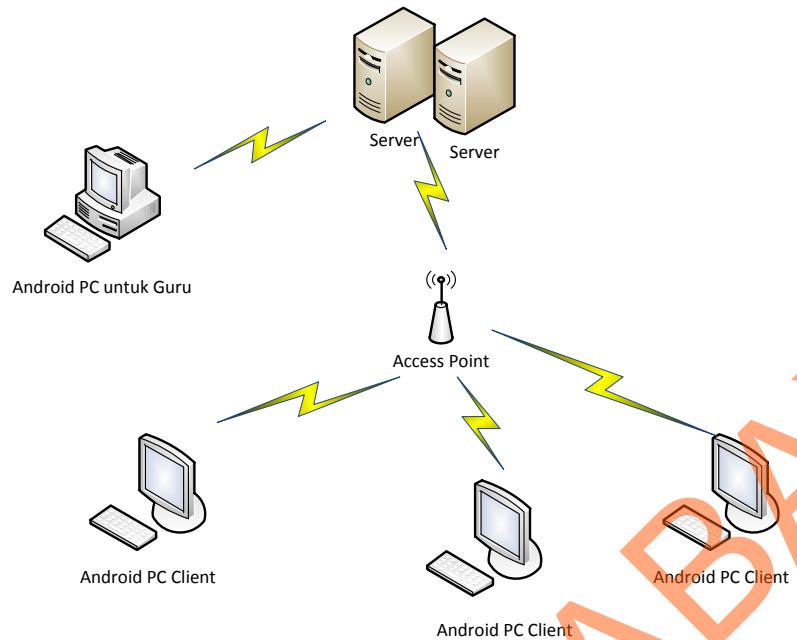
Pada pengembangan aplikasi LMS memanfaatkan teknologi *webview* yang ada pada *android*. *Webview* sendiri merupakan *public class* yang ada di API *android*. *Class* tersebut akan melakukan *request* data ke *controller* yang kemudian akan ditampilkan di-*main activity* aplikasi tersebut. Sedangkan untuk pengembangan aplikasi *webpage* yang dikembangkan dengan menggunakan *design pattern* *MVC*. Menurut Firdaus (2008:2) *MVC* merupakan pola pada pemrograman yang digunakan untuk memisahkan *data acces* dan *bussines logic* dari *data presentasion* dan *user interaction*.



Gambar 3.9 Arsitektur aplikasi LMS berbasis *android*

Pemisahan tersebut dilakukan dengan tujuan agar setiap perubahan yang terjadi pada *presentation logic* atau *bussines logic* tidak memberikan pengaruh satu sama lainnya yang kompleks. Pada arsitektur LMS yang akan dibangun pada *controller* terdapat satu *controller* utama yang menangani *request* dari *client* yaitu *User Controller*. *UserController* digunakan untuk menangani permintaan dari *client* pada fungsi-fungsi utama setelah *login*.

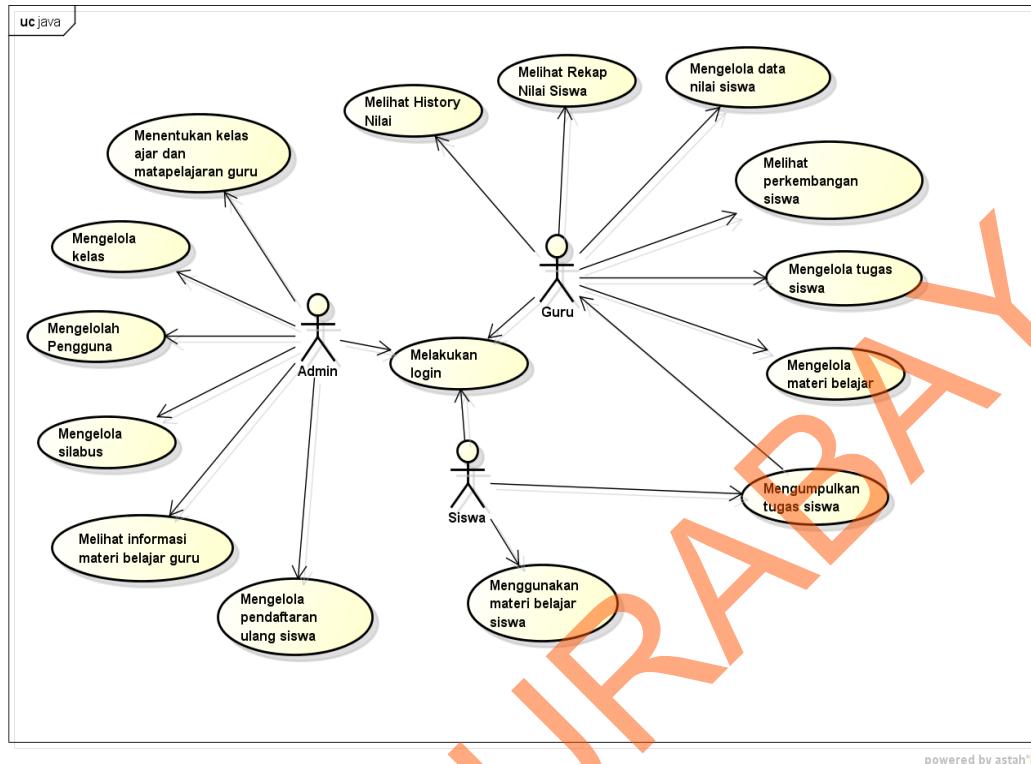
Pada pengembangan aplikasi LMS memiliki arsitektur jaringan seperti pada gambar 3.10 Arsitektur jaringan aplikasi LMS.



Gambar 3.10 Arsitektur jaringan aplikasi LMS

Aplikasi akan ditempatkan pada masing-masing *android PC di client* dan di komputer guru yang berbasis *android PC* juga. Dengan menggunakan *access point* aplikasi pada *client* baik itu computer guru dan murid akan terhubung ke *server* lokal di laboratorium. Pada *server local* tersebut terdapat *web server* yang akan menjalankan *webpage*, dimana *webpage* tersebut akan *di-render* oleh aplikasi *android* yang ada di *client*. Dengan penerapan model tersebut diharapkan aplikasi LMS dapat berjalan dengan baik di laboratorium komputer SD Al-Furqon.

3.3.2 Use Case Diagram Learning Management System



Gambar 3.11 Use Case Diagram Learning Management System

Pada gambar 3.11 teradapat 3 actor pengguna dalam sistem yang akan dibangun yaitu admin , guru dan siswa. Pada *usecase* tersebut admin memiliki peran untuk mengelola kelas, mengelola pendaftaran ulang siswa, mengelola silabus, menentukan guru mengajar dikelas mana dan melihat informasi materi belajar yang didistribusikan oleh guru. Sedangkan guru pada sistem yang akan dikembangkan memiliki fungsi untuk mengelolah nilai tugas siswa, mengelolah materi belajar, mengelola tugas untuk siswa, melihat informasi perkembangan siswa. Untuk siswa pada sistem yang akan dikembangkan dapat menggunakan materi belajar dan tugas yang disediakan guru, serta dapat mengumpulkan tugas yang diberikan guru.

3.3.3 Flow of Event

Pada usecase diagaram terdapat *flow of event* yang digunakan untuk mengetahui langkah-langkah yang dilakukan pengguna dan sistem pada tiap *usecase*. Berikut ini adalah *flow of event* dari *usecase* yang telah dibuat .

A. Flow of event *usecase* login

Flow of event login digunakan oleh semua pengguna. Pada *flow of event* ini terdapat beberapa proses antara lain melakukan input *username* dan *password*, melakukan pengecekan pengguna, validasi *input-an* pengguna seperti terlihat pada tabel berikut ini.

Tabel 3.2 *Flow of event usecase* melakukan *login*

Usecase melakukan login	
Nama Usecase	Melakukan login
Deskripsi singkat	<i>Usecase</i> ini digunakan pengguna untuk melakukan <i>login</i> ke sistem
Aktor	Siswa, Guru, Admin
Prasyarat	Pengguna belum <i>login</i> dan akun pengguna ada dalam database
Alur Utama	<p>1 Pengguna memasukkan <i>username</i> dan <i>password</i></p> <p>2 Sistem melakukan validasi <i>input-an</i> pengguna apakah sudah diisi atau belum,jika belum maka akan ke alur A1 , jika sudah akan berlanjut ke alur 3</p> <p>3 Sistem melakukan cek <i>login</i> dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>UsersModel</i> untuk melakukan <i>query</i> di <i>database</i> dan mengembalikan hasil level pengguna dari sistem, jika data pengguna ada maka akan ke alur 4 jika tidak maka akan ke langkah A2</p> <p>4 Pengguna masuk ke menu utama sesuai dengan level pengguna</p>
Alur alternatif	<p>A1 Sistem menampilkan kesalahan untuk <i>input-an</i> pengguna yang tidak valid dan kembali ke langkah 1</p> <p>A2 Sistem menampilkan pesan kesalahan <i>username</i> dan pengguna salah</p>
Kondisi sukses	Pengguna berhasil <i>login</i> sesuai dengan level pengguna

B. Flow of event usecase mengelola kelas

Flow of event ini digunakan oleh admin pada saat membuat kelas, mengubah data-data kelas yang ada saat ini, dan menghapus data kelas. Seperti terlihat pada tabel 3.3 dibawah ini.

Tabel 3.3 *Flow of event usecase mengelola kelas*

Usecase mengelola kelas																					
Nama Usecase	Mengelola kelas																				
Deskripsi singkat	<i>Usecase</i> mengelola data kelas digunakan Pengguna(admin) untuk menambahkan kelas, mengubah , menghapus data kelas																				
Aktor	Admin																				
Prasyarat	Telah melakukan <i>login</i>																				
Alur Utama	<table border="1"> <tr> <td>1</td><td>Pengguna memilih menu kelola data kelas</td></tr> <tr> <td>2</td><td>Sistem menampilkan data tingkat kelas dari <i>database</i> dengan mengirimkan perminataan ke <i>UserController</i> dan dilanjutkan ke <i>GradesModel</i> untuk mengambil data tingkat kelas</td></tr> <tr> <td>3</td><td>Pengguna memilih data tingkat kelas</td></tr> <tr> <td>4</td><td>Pengguna memasukkan nama kelas yang akan dibuat</td></tr> <tr> <td>5</td><td>Pengguna menyimpan data kelas</td></tr> <tr> <td>6</td><td>Sistem melakukan validasi <i>input</i>-an pengguna apakah data kosong atau tidak jika kosong maka akan menuju ke alur A3 dan jika benar maka akan ke alur 7</td></tr> <tr> <td>7</td><td>Sistem melakukan pengecekan apakah data sudah ada di <i>database</i>, jika data sudah ada maka akan kelangkah A4 dan jika belum maka akan ke langkah 8</td></tr> <tr> <td>8</td><td>Sistem menyimpan data kelas dengan cara mengirimkan data ke <i>usercontroller</i> kemudian diteruskan ke <i>classmodel</i> untuk disimpan di database</td></tr> <tr> <td>9</td><td>Pengguna mengexpand data tingkat kelas pada daftar tingkat kelas</td></tr> <tr> <td>10</td><td>Pengguna memilih kelas untuk melakukan <i>edit</i> atau <i>delete</i> kemudian muncul menu <i>edit</i> dan <i>delete</i> jika <i>edit</i> maka menuju langkah 9 jika <i>edit</i> maka menuju langkah 11</td></tr> </table>	1	Pengguna memilih menu kelola data kelas	2	Sistem menampilkan data tingkat kelas dari <i>database</i> dengan mengirimkan perminataan ke <i>UserController</i> dan dilanjutkan ke <i>GradesModel</i> untuk mengambil data tingkat kelas	3	Pengguna memilih data tingkat kelas	4	Pengguna memasukkan nama kelas yang akan dibuat	5	Pengguna menyimpan data kelas	6	Sistem melakukan validasi <i>input</i> -an pengguna apakah data kosong atau tidak jika kosong maka akan menuju ke alur A3 dan jika benar maka akan ke alur 7	7	Sistem melakukan pengecekan apakah data sudah ada di <i>database</i> , jika data sudah ada maka akan kelangkah A4 dan jika belum maka akan ke langkah 8	8	Sistem menyimpan data kelas dengan cara mengirimkan data ke <i>usercontroller</i> kemudian diteruskan ke <i>classmodel</i> untuk disimpan di database	9	Pengguna mengexpand data tingkat kelas pada daftar tingkat kelas	10	Pengguna memilih kelas untuk melakukan <i>edit</i> atau <i>delete</i> kemudian muncul menu <i>edit</i> dan <i>delete</i> jika <i>edit</i> maka menuju langkah 9 jika <i>edit</i> maka menuju langkah 11
1	Pengguna memilih menu kelola data kelas																				
2	Sistem menampilkan data tingkat kelas dari <i>database</i> dengan mengirimkan perminataan ke <i>UserController</i> dan dilanjutkan ke <i>GradesModel</i> untuk mengambil data tingkat kelas																				
3	Pengguna memilih data tingkat kelas																				
4	Pengguna memasukkan nama kelas yang akan dibuat																				
5	Pengguna menyimpan data kelas																				
6	Sistem melakukan validasi <i>input</i> -an pengguna apakah data kosong atau tidak jika kosong maka akan menuju ke alur A3 dan jika benar maka akan ke alur 7																				
7	Sistem melakukan pengecekan apakah data sudah ada di <i>database</i> , jika data sudah ada maka akan kelangkah A4 dan jika belum maka akan ke langkah 8																				
8	Sistem menyimpan data kelas dengan cara mengirimkan data ke <i>usercontroller</i> kemudian diteruskan ke <i>classmodel</i> untuk disimpan di database																				
9	Pengguna mengexpand data tingkat kelas pada daftar tingkat kelas																				
10	Pengguna memilih kelas untuk melakukan <i>edit</i> atau <i>delete</i> kemudian muncul menu <i>edit</i> dan <i>delete</i> jika <i>edit</i> maka menuju langkah 9 jika <i>edit</i> maka menuju langkah 11																				

Usecase mengelola kelas	
	11 Jika pengguna memilih <i>edit</i> maka pengguna data kelas akan masuk ke <i>form</i> untuk <i>edit</i> data terpilih
	12 Pengguna melakukan perubahan data kelas
	13 Sistem melakukan validasi <i>input</i> -an pengguna apakah data kosong atau tidak jika kosong maka akan menuju ke alur A3 dan jika benar maka akan ke alur 12
	14 Sistem melakukan perubahan data kelas dengan mengirimkan data ke <i>usercontroller</i> selanjutnya akan di teruskan ke <i>classmodel</i> untuk dilakukan perubahan data di <i>database</i>
	15 Jika pengguna memilih <i>delete</i> maka akan muncul <i>modal dialog</i> untuk menanyakan apakah akan menghapus data tersebut jika ya maka akan ke langkah 14 jika tidak maka akan ke langkah 15
	16 Sistem mengirimkan data ke <i>usercontroller</i> selanjutnya akan di teruskan ke <i>classmodel</i> dan dilakukan <i>delete</i> data tersebut
	17 Sistem tidak melakukan penghapusan data
Alur alternatif	A1 Jika data tingkat kelas yang ditampilkan sistem tidak ada dalam daftar tingkat kelas
	A2 Pengguna menambahkan data tingkat kelas baru pada <i>form</i> tersebut
	A3 Sistem akan mengirimkan data ke <i>usercontroller</i> dan diteruskan ke <i>gradesmodel</i> untuk dilakukan <i>insert</i> data ke <i>database</i>
	A4 Sistem menampilkan pesan kesalahan kesalahan jika kondisi sistem sedang melakukan perubahan maka kembali ke langkah 9 , namun jika kondisi sistem sedang melakukan penambahan data baru maka kembali ke langkah 3
Kondisi sukses	1 Pengguna berhasil menambahkan data kelas
	2 Pengguna berhasil mengubah data kelas
	3 Pengguna berhasil menghapus data kelas

C. Flow of event *usecase* mengelola pendaftaran ulang siswa

Flow of event *usecase* mengelola pendaftaran digunakan pada saat awal-awal semester ketika siswa melakukan daftar ulang. Pada proses ini siswa didaftarkan untuk mendapatkan kelas baru pada tahun ajaran berlangsung.

Tabel 3.4 Flow of event usecase mengelola pendaftaran ulang siswa

Usecase mengelola pendaftaran ulang siswa	
Nama Usecase	Mengelola pendaftaran ulang siswa
Deskripsi singkat	Usecase mengelolah pendaftaran ulang siswa digunakan untuk Pengguna(Admin) untuk mendaftarkan siswa pada awal semester agar mendapatkan kelas
Aktor	Admin
Prasyarat	Telah melakukan <i>login</i>
Alur Utama	<p>1 Pengguna memilih menu registrasi</p> <p>2 Sistem menampilkan data kelas dari <i>database</i> dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>ClassModel</i> untuk melakukan <i>query</i></p> <p>3 Sistem menampilkan data periode tahun ajaran dari <i>database</i> dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>YearPeriodModel</i> untuk melakukan <i>query</i></p> <p>4 Pengguna memilih periode semester</p> <p>5 Pengguna memilih kelas yang akan dimasuki oleh siswa</p> <p>6 Sistem menampilkan data siswa yang belum memiliki kelas dari <i>database</i> dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>StudentClassModel</i> jika ada selanjutnya ke langkah 7 jika tidak maka akan ke langkah A1</p> <p>7 Pengguna memilih siswa yang belum memiliki kelas pada semester ini</p> <p>8 Pengguna menyimpan data pendaftaran ulang siswa</p> <p>9 Sistem melakukan validasi <i>input</i>-an pengguna apabila data tidak valid maka akan ke langkah 10 jika valid maka akan ke langkah A2</p> <p>10 Sistem melakukan penyimpanan data pendaftaran ulang siswa dengan cara mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>StudentClassModel</i> untuk menyimpan data di <i>database</i></p> <p>11 Pengguna memilih data tahun ajaran untuk mengubah data</p> <p>12 Pengguna memilih data kelas untuk menampilkan data siswa</p>

Usecase mengelola pendaftaran ulang siswa		
	13	Sistem menampilkan data siswa dari <i>database</i> berdasarkan kelas terpilih dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>StudentClassModel</i> untuk mengambil data di <i>database</i>
	14	Pengguna memilih siswa tersebut untuk melakukan perubahan data pendaftaran siswa
	15	Pengguna mengubah data pendaftaran siswa
	16	Pengguna menyimpan perubahan data pendaftaran siswa
		Sistem melakukan validasi <i>input</i> -an pengguna apabila data tidak valid maka akan ke langkah 17 jika valid maka akan ke langkah A2
	17	Sistem melakukan perubahan data siswa tersebut dengan dengan cara mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>StudentClassModel</i> untuk merubah data terpilih di <i>database</i>
Alur alternatif	A1	Sistem tidak menampilkan daftar siswa dikarenakan siswa sudah terdaftar semua
	A2	Sistem menampilkan pesan kesalahan jika kondisi pada saat melakukan perubahan data maka akan ke langkah 11 jika melakukan penambahan data maka akan ke langkah 7
Kondisi sukses	1	Pengguna berhasil menambahkan data registrasi siswa
	2	Pengguna berhasil melakukan perubahan data registrasi siswa

D. Flow of event usecase mengelola silabus

Flow of event ini digunakan untuk mengelola silabus oleh admin. Proses tersebut dilakukan pada awal semester untuk membuat rencana pembelajaran oleh admin. Selanjutnya hasil pembuatan silabus tersebut akan dibuat *template* oleh guru dalam memberikan materi kepada siswa.

Tabel 3.5 Flow of event usecase mengelola silabus.

Usecase mengelola silabus	
Nama Usecase	Mengelola silabus
Deskripsi singkat	<i>Usecase</i> digunakan pengguna(admin) untuk mengelola data rencana belajar mata pelajaran pada tingkat kelas
Aktor	Admin
Prasyarat	Telah melakukan <i>login</i> dan memilih menu silabus
Alur Utama	<p>1 Sistem menampilkan daftar tahun ajaran dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>YearPeriodsModel</i> untuk mengambil data tahun ajar</p> <p>2 Sistem menampilkan daftar tingkat kelas dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>GradesModel</i> untuk mengambil data tingkat kelas</p> <p>3 Sistem menampilkan daftar matapelajaran dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>CoursesModel</i> untuk mengambil data matapelajaran</p> <p>4 Pengguna memilih periode tahun ajaran,tingkat kelas dan mata pelajaran</p> <p>5 Pengguna mengisi data pertemuan ke-berapa dan mengisikan garis besar materi belajar</p> <p>6 Pengguna menyimpan data silabus</p> <p>7 Sistem melakukan validasi <i>input</i>-an pengguna jika data benar maka akan ke langkah A1 jika data salah maka akan ke langkah 8</p> <p>8 Sistem melakukan pengecekan apakah silabus tersebut ada di <i>database</i> atau tidak dengan cara mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i>, jika ada maka akan ke ke langkah A2 jika tidak ada maka akan ke langkah 9</p> <p>9 Sistem menyimpan data silabus dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk dilakukan penyimpanan ke <i>database</i></p> <p>10 Pengguna memilih periode tahun ajaran dan mata pelajaran</p> <p>11 Sistem menampilkan data silabus pada matapelajaran terpilih dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk mengambil data silabus</p>

Usecase mengelola silabus		
	12	Pengguna memilih silabus yang telah dikelompokkan berdasarkan tingkat kelas dan akan muncul menu <i>edit</i> dan <i>delete</i> , jika memilih <i>edit</i> maka akan ke langkah 13 dan jika memilih <i>delete</i> maka akan ke langkah 18
	13	Pengguna melakukan perubahan pada data silabus yang dipilih
	14	Pengguna menyimpan data perubahan silabus
	15	Sistem melakukan validasi <i>input-an</i> pengguna jika data benar maka akan ke langkah A1 jika data salah maka akan ke langkah 16
	16	Sistem melakukan pengecekan apakah silabus tersebut ada di database atau tidak dengan cara mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> , jika ada maka akan ke langkah A2 jika tidak ada maka akan ke langkah 17
	17	Sistem mengubah data perubahan silabus dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk dilakukan perubahan ke <i>database</i>
	18	Pengguna menghapus data silabus, maka akan muncul modal <i>dialog</i> jika pengguna memilih ya maka ke langkah 19 jika tidak maka akan ke langkah 20
	19	Sistem akan melakukan penghapusan data dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk dilakukan penghapusan data silabus di <i>database</i>
	20	Sistem membatalkan penghapusan data
Alur alternatif	A1	Sistem menampilkan pesan kesalahan karena <i>input-an</i> pengguna tidak valid jika kondisi pada saat melakukan penambahan data maka akan kembali ke langkah 4 jika pada saat perubahan data maka akan kembali ke langkah 12
	A2	Sistem menampilkan pesan kesalahan karena data silabus sudah ada jika kondisi pada saat melakukan penambahan data maka akan kembali ke langkah 4 jika pada saat perubahan data maka akan kembali ke langkah 12
Kondisi sukses	1	Pengguna berhasil menambahkan data silabus baru
	2	Pengguna berhasil melakukan perubahan data silabus
	3	Pengguna berhasil menyimpan data silabus

E. Flow of event *usecase* mengelola materi belajar

Flow of event ini digunakan untuk melakukan pengelolaan materi berlajar pada kegiatan belajar mengajar yang dilakukan oleh guru. Seperti terlihat pada tabel 3.6 berikut ini.

Tabel 3.6 *Flow of event usecase* mengelola materi belajar

Usecase mengelola materi belajar	
Nama Usecase	Mengelola materi belajar
Deskripsi singkat	<i>Usecase</i> mengelola materi belajar digunakan untuk menambahkan, mengubah data materi belajar dan menghapus materi belajar yang akan didistribusikan ke siswa
Aktor	Guru
Prasyarat	Telah <i>login</i> dan memilih menu materi belajar
Alur Utama	<p>1 Sistem menampilkan data kelas yang diajar oleh guru dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data guru yang mengajar</p> <p>2 Sistem menampilkan data mata pelajaran yang diajar oleh guru dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk mengambil data matapelajaran yang diajar guru</p> <p>3 Sistem menampilkan daftar pertemuan dan materi yang telah didistribusikan pada mata pelajaran tersebut berdasarkan silabus dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>ResourcesModel</i> untuk mengambil data materi belajar guru yang bersangkutan</p> <p>4 Pengguna memilih kelas dan mata pelajaran yang akan diberikan materi berdasarkan kelas</p> <p>5 Pengguna menambahkan materi yang akan didistribusikan</p> <p>6 Sistem memunculkan <i>dialog</i> untuk memilih file dan menambahkan keterangan materi yang akan didistribusikan</p> <p>7 Sistem melakukan validasi <i>input</i>-an pengguna jika data valid maka akan ke langkah 8 jika tidak maka akan ke langkah A1</p>

Usecase mengelola materi belajar		
	8	Sistem menyimpan data materi belajar yang ditambahkan pengguna dengan mengirimkan data ke <i>usercontroller</i> dan diteruskan ke <i>resourcemodel</i> untuk dilakukan penyimpanan di <i>database</i>
	9	Pengguna memilih <i>edit</i> pada daftar materi yang telah didistribusikan
	10	Sistem menampilkan keterangan materi belajar pada <i>form</i>
	11	Pengguna melakukan perubahan data (keterangan informasi materi belajar)
	12	Pengguna menyimpan perubahan data materi belajar
	13	Sistem melakukan validasi <i>input</i> -an pengguna jika data valid maka akan ke langkah 13 jika tidak maka akan ke langkah A1
	14	Sistem mengubah data materi belajar yang diubah pengguna dengan mengirimkan data ke <i>usercontroller</i> dan diteruskan ke <i>resourcemodel</i> untuk dilakukan perubahan di <i>database</i>
	15	Pengguna melakukan penghapusan data materi belajar maka akan muncul data <i>dialog</i> untuk menanyakan apakah data akan dihapus, jika pengguna memilih ya maka akan ke langkah 16, jika pengguna memilih tidak maka akan ke langkah 17
	16	Sistem akan menghapus materi belajar dengan mengirimkan data ke <i>usercontroller</i> dan diteruskan ke <i>resourcemodel</i> untuk dilakukan penghapusan di <i>database</i>
	17	Sistem batal menghapus materi belajar
Alur alternatif	A1	Sistem menampilkan pesan kesalahan karena <i>input</i> -an pengguna tidak valid jika kondisi pada saat melakukan penambahan data maka akan kembali ke langkah 4 jika pada saat perubahan data maka akan kembali ke langkah 11
Kondisi sukses	1	Pengguna berhasil melakukan penambahan data materi
	2	Pengguna berhasil melakukan perubahan data materi
	3	Pengguna berhasil melakukan penghapusan data materi

F. Flow of event usecase mengelola tugas

Flow of event mengelola tugas digunakan oleh guru pada tiap kali memberikan tugas ke siswa. Tugas-tugas yang diberitahukan kepada siswa akan dapat dilihat oleh siswa. Seperti terlihat pada tabel 3.7 berikut ini.

Tabel 3.7 *Flow of event usecase mengelola tugas.*

Usecase mengelola tugas	
Nama Usecase	Mengelola tugas
Deskripsi singkat	<i>Usecase mengelola tugas dilakukan oleh pengguna untuk menambahkan tugas, mengubah data tugas , menghapus tugas yang akan diberikan siswa</i>
Aktor	Guru
Prasyarat	Pengguna telah <i>login</i> dan memilih menu tugas siswa
Alur Utama	<p>1 Sistem menampilkan data kelas yang diajar oleh guru yang diambil dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data guru yang mengajar</p> <p>2 Pengguna memilih kelas yang diajar</p> <p>3 Sistem menampilkan data mata pelajaran yang diajar oleh guru pada kelas tersebut dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data mata pelajaran yang diajar oleh guru</p> <p>4 Pengguna memilih matapelajaran yang akan diberikan tugas</p> <p>5 Sistem menampilkan silabus pada mata pelajaran tersebut dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk mengambil data data silabus untuk guru tersebut</p> <p>6 Pengguna memilih pertemuan ke-berapa pada matapelajaran terpilih</p> <p>7 Sistem menampilkan menu tambahkan tugas dan lihat tugas, jika pengguna memilih menu tambahkan tugas maka akan ke langkah 8 jika pengguna memilih menu lihat tugas maka akan ke langkah 12</p> <p>8 Pengguna menambahkan tugas baru</p> <p>9 Pengguna menyimpan data tugas yang baru</p>

Usecase mengelola tugas		
	10	Sistem melakukan validasi <i>input</i> -an tugas yang baru ditambahkan jika data valid maka akan ke langkah 10 jika tidak maka akan ke langkah A1
	11	Sistem menyimpan data tugas yang baru dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>TaskModel</i> untuk disimpan di <i>database</i>
	12	Pengguna memilih tugas pada daftar tugas untuk pertemuan yang terpilih
	13	Sistem menampilkan menu <i>edit</i> dan <i>delete</i> pada daftar tugas tersebut
	14	Jika pengguna memilih menu <i>delete</i> maka akan ke langkah 21 jika pengguna memilih menu <i>edit</i> maka akan ke langkah 15
	15	Sistem akan menampilkan keterangan tugas pada <i>form</i> untuk melakukan perubahan data
	16	Pengguna melakukan perubahan data tugas
	17	Pengguna menyimpan perubahan data tugas terpilih
	18	Sistem melakukan validasi <i>input</i> -an tugas yang baru ditambahkan jika data valid maka akan ke langkah 19 jika tidak maka akan ke langkah A1
	19	Sistem mengubah data tugas tersebut dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>TaskModel</i> untuk melakukan perubahan di <i>database</i>
	20	Pengguna menghapus tugas tersebut
	21	Sistem memunculkan <i>dialog</i> untuk menghapus data tugas tersebut, jika pengguna memilih ya maka akan ke langkah 22 jika mengguna memilih tidak maka akan ke langkah 23
	22	Sistem akan menghapus data tersebut dengan mengirimkan data ke <i>UserController</i> dan diteruskan ke <i>TaskModel</i> untuk melakukan penghapusan di <i>database</i>
	23	Sistem batal menghapus data tersebut
Alur alternatif	A1	Sistem menampilkan pesan kesalahan karena <i>input</i> -an pengguna tidak valid jika kondisi pada saat melakukan penambahan data maka akan kembali ke langkah 6 jika pada saat perubahan data maka akan kembali ke langkah 14
Kondisi sukses	1	Pengguna berhasil menambahkan data tugas baru
	2	Pengguna berhasil melakukan perubahan data tugas
	3	Pengguna berhasil menghapus data tugas

G. Flow of event *usecase* mengelola data nilai siswa

Flow of event ini digunakan oleh guru pada saat guru melakukan penilaian terhadap tugas –tugas siswa pada setiap harinya. Pada proses tersebut guru dapat mengunduh tugas siswa yang telah dikumpulkan.

Tabel 3.8 Flow of event *usecase* mengelola data nilai siswa.

Usecase mengelola data nilai siswa	
Nama Usecase	Mengelola data nilai siswa
Deskripsi singkat	<i>Usecase</i> mengelola data nilai siswa digunakan untuk melakukan penilaian data tugas siswa
Aktor	Guru
Prasyarat	Pengguna telah <i>login</i> dan memilih menu input nilai siswa
Alur Utama	<p>1 Sistem menampilkan data mata pelajaran dan kelas yang diajar oleh guru dari <i>databased</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data guru yang mengajar</p> <p>2 Pengguna memilih mata pelajaran dan kelas untuk menampilkan data tugas siswa</p> <p>3 Sistem menampilkan data tugas berdasarkan data matapelajaran terpilih dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>TasksModel</i> untuk mengambil data tugas yang diberikan guru</p> <p>4 Pengguna memilih daftar tugas</p> <p>5 Sistem akan menampilkan siswa pada kelas tersebut beserta tugas yang dikumpulkan untuk tugas tersebut dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>AssesmentsModel</i> untuk mengambil data tugas yang telah dikerjakan siswa</p> <p>6 Jika sistem telah memiliki nilai maka pengguna dapat merubah nilai tersebut pada langkah 11 jika nilai belum ada maka pengguna dapat mengisikan nilai tugas tersebut pada langkah 7</p> <p>7 Pengguna mengisi nilai siswa tersebut</p> <p>8 Pengguna menyimpan data nilai siswa tersebut</p> <p>9 Sistem melakukan validasi <i>input</i>-an pengguna jika data benar maka akan ke langkah 10 jika data salah maka akan ke langkah A1</p>

Usecase mengelola data nilai siswa	
	10 Sistem menyimpan data nilai tugas tersebut dengan mengirimkan data ke <i>Usercontroller</i> selanjutnya di teruskan ke <i>PointModel</i> untuk dilakukan penyimpanan data di <i>database</i>
	11 Pengguna melakukan perubahan terhadap data nilai siswa tersebut
	12 Sistem melakukan validasi <i>input-an</i> pengguna jika data benar maka akan ke langkah 13 jika data salah maka akan ke langkah A1
	13 Sistem melakukan perubahan data nilai tersebut dengan mengirimkan data ke <i>Usercontroller</i> selanjutnya di teruskan ke <i>PointModel</i> untuk dilakukan perubahan data di <i>database</i>
	14 Pengguna dapat menunduh tugas yang dikumpulkan siswa apabila ada tugas yang dikumpulkan siswa pada daftar nilai tugas siswa jika pada saat mengunduh tugas siswa gagal maka akan ke langkah A2
Alur alternatif	A1 Sistem menampilkan pesan kesalahan karena <i>input-an</i> pengguna tidak valid jika kondisi pada saat melakukan penambahan data maka akan kembali ke langkah 7 jika pada saat perubahan data maka akan kembali ke langkah 11
	A2 Sistem menampilkan gagal <i>download</i>
Kondisi sukses	1 Pengguna berhasil menambahkan data nilai baru
	2 Pengguna berhasil melakukan perubahan data nilai

H. Flow of event *usecase* melihat perkembangan nilai siswa

Flow of event ini digunakan oleh guru untuk melihat perkembangan nilai siswa anak didiknya. Laporan yang ditampilkan berupa grafik nilai siswa pada tiap tugasnya. Seperti terlihat pada tabel 3.9 berikut ini.

Tabel 3.9 Flow of event *usecase* melihat perkembangan nilai siswa.

Usecase melihat perkembangan nilai siswa	
Nama Usecase	Melihat perkembangan nilai siswa
Deskripsi singkat	<i>Usecase</i> melihat perkembangan nilai siswa digunakan untuk guru melihat hasil perkembangan nilai anak didiknya

Usecase melihat perkembangan nilai siswa		
Aktor	Guru	
Prasyarat	Pengguna telah <i>login</i> dan memilih menu laporan nilai siswa	
Alur Utama	1	Sistem menampilkan data kelas yang diajar oleh guru dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data guru yang mengajar
	2	Pengguna memilih data kelas yang di ajar
	3	Sistem menampilkan matapelajaran yang diajar oleh pengguna pada kelas terpilih dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data mata pelajaran yang diajar oleh guru
	4	Pengguna memilih mata pelajaran pada daftar mata pelajaran
	5	Sistem menampilkan data siswa pada matapelajaran tersebut dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>StudentClassModel</i> untuk mengambil data siswa yang diajar oleh guru di kelas tersebut
	6	Pengguna memilih siswa tersebut
	7	Sistem menampilkan grafik perkembangan nilai tugas siswa yang didapat dari mengirimkan data ke <i>UserController</i> diteruskan ke <i>PointModel</i> untuk mengambil data nilai siswa
Alur alternatif	A1	Sistem tidak menampilkan grafik dikarenakan belum ada tugas yang dibuat oleh guru kemudian sistem menampilkan pesan kesalahan selanjutnya sistem kembali pada langkah 3
Kondisi sukses	1	Pengguna berhasil menampilkan laporan grafik nilai perkembangan siswa

I. Flow of event usecase mengumpulkan tugas

Flow of event ini digunakan pada proses siswa mengumpulkan tugas siswa melalui komputer. Selanjutnya tugas tersebut akan dinilai oleh guru. Seperti terlihat pada tabel 3.10 berikut ini.

Tabel 3.10 Flow of event usecase mengumpulkan tugas.

Usecase mengumpulkan tugas	
Nama Usecase	Mengumpulkan tugas siswa
Deskripsi singkat	<i>Usecase</i> ini digunakan untuk mengumpulkan tugas siswa , tugas disini bisa tugas yang dikumpulkan oleh siswa melalui komputer atau tugas yang langsung dikumpulkan saat selesai mengerjakan tugas ditempat siswa
Aktor Utama	Siswa
Aktor sekunder	Guru
Prasyarat	Pengguna telah <i>login</i>
Alur Utama	<p>1 Sistem menampilkan data tugas yang diberikan guru untuk mata pelajaran hari ini dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>TasksModel</i> untuk mengambil data tugas siswa yang diberikan guru</p> <p>2 Sistem menampilkan daftar tugas yang telah dikumpulkan pengguna berdasarkan mata pelajaran dari <i>databased</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>AssesmentModel</i> untuk mengambil data tugas siswa</p> <p>3 Pengguna memilih tugas yang akan dikumpulkan</p> <p>4 Jika belum ada tugas maka pengguna dapat mengklik <i>refresh</i> untuk menampilkan ulang tugas hari dan kembali ke langkah 1 jika sudah ada melanjutkan ke langkah 5</p> <p>5 Pengguna mengisi keterangan dari tugas dan memilih <i>file</i> tugas tersebut</p> <p>6 Pengguna mengumpulkan tugas tersebut untuk tugas yang dikumpulkan melalui komputer, untuk tugas yang dinilai langsung guru akan mendatangi komputer siswa bersangkutan</p> <p>7 Sistem melakukan validasi terhadap <i>input-an</i> pengguna jika data valid maka akan ke langkah 8 jika data tidak valid maka akan ke langkah A1</p> <p>8 Sistem menyimpan data tugas pengguna dengan cara mengirimkan dana ke <i>Usercontroller</i> selanjutnya diteruskan ke <i>AssesmentModel</i> untuk dilakukan penyimpanan data di <i>database</i></p> <p>9 Sistem menampilkan ulang daftar tugas yang telah dikumpulkan pengguna berdasarkan mata pelajaran dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>AssesmentsModel</i> untuk mengambil data tugas siswa</p>

Usecase mengumpulkan tugas		
	10	Pengguna memilih tugas yang telah dikumpulkan
	11	Pengguna menekan <i>button</i> buka untuk membuka tugas yang telah dikumpulkan
	12	Sistem akan mendownload tugas tersebut
Alur alternatif	A1	Sistem menampilkan pesan kesalahan untuk <i>input</i> -an pengguna yang salah selanjutnya sistem kembali pada langkah 3
Kondisi sukses	1	Pengguna berhasil menyimpan data tugas siswa
	2	Pengguna berhasil melihat data tugas yang telah dikumpulkan

J. Flow of event *usecase* menggunakan materi belajar siswa

Flow of event ini digunakan oleh siswa setiap kali siswa akan menggunakan materi belajar yang disediakan guru. Materi belajar guru dapat diunduh oleh siswa dan siswa dapat membuka materi tersebut tanpa harus keluar dari aplikasi LMS .

Tabel 3.11 *Flow of event usecase* menggunakan materi belajar.

Usecase menggunakan materi belajar siswa		
Nama Usecase		Menggunakan materi belajar siswa
Deskripsi singkat		<i>Usecase</i> ini digunakan pengguna untuk mengunduh dan membuka materi belajar yang disediakan oleh guru
Aktor		Siswa
Prasyarat		Pengguna telah melakukan login
Alur Utama	1	Sistem menampilkan daftar materi yang akan digunakan guru pada hari ini dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>ResourcesModel</i> untuk mengambil data materi belajar hari ini
	2	Sistem menampilkan daftar materi keseluruhan yang disediakan guru pengajar dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>ResourcesModel</i> untuk mengambil data materi belajar hari ini
	3	Pengguna memilih materi yang digunakan hari ini

Usecase menggunakan materi belajar siswa		
	4	Pengguna mengunduh materi hari ini
	5	Sistem akan melakukan <i>download</i> untuk materi hari ini jika materi bertipe aplikasi maka akan ke langkah A1
	6	Pengguna dapat membuka materi yang telah diunduh
	7	Sistem akan mengecek tipe <i>file</i> pada materi tersebut akan dibuka dengan menggunakan aplikasi pendukung yang cocok jika aplikasi pendukung belum terinstall maka akan kelangkah A2 jika sudah ada maka akan kelangkah 8
	8	Sistem akan membuka aplikasi tersebut
	9	Pengguna memilih materi minggu-minggu yang lalu
	10	Selanjutnya sistem akan berlanjut ke langkah no 5
	A1	Sistem menampilkan pesan bahwa materi yang di unduh harus diinstal untuk tipe materi yang bertipe aplikasi
	A2	Sistem akan menginstall aplikasi tersebut
	Kondisi sukses	1 Pengguna berhasil menggunakan materi belajar
	Kondisi gagal	1 Terjadi kesalahan dalam <i>database</i>

K. Flow of event *usecase* melihat informasi materi belajar guru

Flow of event ini digunakan oleh admin untuk melihat guru-guru siapa saja yang tidak mendistribusikan materi. Materi – materi yang belum didistribusikan guru akan terlihat pada tiap pertemuan yang tidak memiliki materi. Seperti terlihat pada tabel 3.12 berikut ini.

Tabel 3.12 Flow of event *usecase* melihat informasi materi belajar.

Usecase melihat informasi materi belajar guru	
Nama Usecase	Melihat informasi materi belajar guru
Deskripsi singkat	<i>Usecase</i> ini digunakan admin untuk melihat materi belajar guru
Aktor	Admin
Prasyarat	Pengguna telah melakukan login dan memilih menu informasi perkembangan guru

Usecase melihat informasi materi belajar guru		
Alur Utama	1	Sistem menampilkan data tahun ajaran belajar dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>YearPeriodModel</i> untuk melakukan <i>query</i>
	2	Pengguna memilih periode belajar pada daftar tahun ajar
	3	Sistem menampilkan data silabus tiap pertemuan yang belum memiliki materi yang didistribusikan dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk melakukan <i>query</i> di <i>database</i>
	4	Sistem menampilkan daftar guru yang belum mengumpulkan materi belajar dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>SyllabusModel</i> untuk mengambil data guru yang belum membuat materi sesuai silabus yang ditetapkan
	5	Pengguna dapat melihat informasi materi belajar guru yang belum terisi jika pengguna ingin memfilter berdasarkan nama guru maka akan ke langkah A1
Alur alternatif	A1	Pengguna melakukan <i>filter</i> berdasarkan nama guru dan berlanjut ke langkah 4
Kondisi sukses	1	Pengguna berhasil menampilkan informasi materi yang belum diisi oleh guru
Kondisi gagal	1	Terjadi kesalahan dalam <i>database</i>

L. Flow of event *usecase* menentukan kelas ajar dan matapelajaran yang diajar guru

Flow of event ini digunakan oleh admin pada awal-awal semester untuk menentukan kelas dan matapelajaran yang diajar oleh guru. Seperti terlihat pada tabel 3.13 berikut ini.

Tabel 3.13 Flow of event usecase menentukan kelas ajar dan matapelajaran yang diajar guru.

Usecase menentukan kelas ajar dan matapelajaran yang diajar guru	
Nama Usecase	Menentukan kelas ajar dan mata pelajaran yang diajar guru
Deskripsi singkat	Usecase ini digunakan admin untuk melihat materi belajar guru
Aktor	Admin
Prasyarat	Pengguna telah melakukan <i>login</i> dan memilih menu informasi perkembangan guru
Alur Utama	<p>1 Sistem menampilkan daftar tahun ajaran dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>YearPeriodModel</i> untuk mengambil data tahun ajar di <i>database</i></p> <p>2 Sistem menampilkan daftar kelas dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>ClassModel</i> untuk mengambil data kelas di <i>database</i></p> <p>4 Sistem menampilkan daftar mata pelajaran dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>CourseModel</i> untuk mengambil data tahun ajar di <i>database</i></p> <p>5 Sistem menampilkan daftar guru dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>UserModel</i> untuk mengambil data guru di <i>database</i></p> <p>6 Pengguna memilih tahun ajaran pada daftar tahun ajaran</p> <p>7 Pengguna memilih kelas yang akan diajar oleh guru terpilih</p> <p>8 Pengguna memilih matapelajaran</p> <p>9 Pengguna memilih guru yang akan mengajar dikelas tersebut</p> <p>10 Pengguna menyimpan pendaftaran guru tersebut</p> <p>11 Sistem melakukan validasi <i>input</i>-an pengguna jika data valid maka akan ke langkah 12 jika tidak maka akan ke langkah A2</p> <p>12 Sistem akan menyimpan data pendaftaran tersebut dengan mengirimkan data ke <i>UserController</i> dan meneruskan ke <i>LessonModel</i> untuk menyimpan data guru mengajar</p> <p>13 Pengguna dapat memilih tahun ajaran</p> <p>14 Pengguna memilih kelas</p>

Usecase menentukan kelas ajar dan matapelajaran yang diajar guru	
15	Sistem akan menampilkan daftar mata pelajaran beserta guru yang mengajar dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data guru mengajar di <i>database</i>
16	Pengguna dapat memilih daftar mata pelajaran dan guru yang mengajar tersebut kemudian sistem akan menampilkan menu <i>edit</i> atau <i>delete</i>
17	Pengguna untuk memilih menu edit maka akan menuju ke langkah 18 jika pengguna memilih <i>delete</i> maka menuju langkah 24
18	Sistem akan menampilkan guru dan mata pelajaran yang diajar terpilih ke <i>form</i> untuk dilakukan perubahan data
19	Pengguna melakukan perubahan data guru mengajar
20	Pengguna menyimpan perubahan data guru mengajar
21	Sistem melakukan validasi <i>input</i> -an pengguna jika data valid maka akan ke langkah 22 jika tidak maka akan ke langkah A2
22	Sistem akan mengubah data perubahan data guru mengajar
23	Pengguna menghapus data mata pelajaran dan kelas yang diajar oleh guru
24	Sistem akan menampilkan pesan <i>dialog</i> jika pengguna memilih ya maka akan ke langkah 23 jika pengguna memilih tidak maka akan ke langkah 24
25	Sistem akan melakukan penghapusan data guru mengajar
26	Sistem batal menghapus guru mengajar
Alur alternatif	Pengguna menambahkan mata pelajaran baru karena mata pelajaran tidak ada kemudian sistem akan menyimpan data matapelajaran baru dan kembali ke langkah 8
A1	Sistem menampilkan pesan kesalahan <i>input</i> -an pengguna jika pada saat posisi mengubah data maka akan ke langkah 6 jika saat posisi mengubah data maka akan ke langkah 19
Kondisi sukses	<p>1 Pengguna berhasil melakukan menambahkan data guru mengajar</p> <p>2 Pengguna berhasil melakukan perubahan data guru mengajar</p> <p>3 Pengguna berhasil melakukan penghapusan data guru mengajar</p>
Kondisi gagal	1 Terjadi kesalahan dalam <i>database</i>

M. Flow of event *usecase* mengelola data pengguna

Flow of event ini digunakan oleh admin pada awal-awal semester untuk memasukkan pengguna baru dari aplikasi LMS yang akan dibangun .Pada fungsi tersebut admin dapat melakukan penambahan data pengguna baru, perubahan data pengguna , pencarian data pengguna, dan penghapusan data pengguna yang ada didalam *database* seperti terlihat pada tabel 3.14 berikut ini.

Tabel 3.14 *Flow of event usecase* mengelolah pengguna.

Usecase mengelola pengguna	
Nama Usecase	Mengelola pengguna
Deskripsi singkat	<i>Usecase</i> ini digunakan admin untuk mengelola data pengguna
Aktor	Admin
Prasyarat	Pengguna telah melakukan login dan memilih menu informasi perkembangan guru
Alur Utama	<p>1 Sistem menampilkan data pengguna dengan cara mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>UserModel</i> untuk mengambil data pengguna di <i>database</i></p> <p>2 Pengguna memasukkan data pengguna</p> <p>3 Pengguna menyimpan data pengguna baru tersebut</p> <p>4 Sistem melakukan validasi <i>input</i>-an pengguna jika data valid maka akan ke langkah 4 jika tidak maka akan ke langkah A1</p> <p>5 Sistem akan menyimpan data pengguna tersebut dengan mengirimkan data ke <i>UserController</i> dan meneruskan ke <i>UserModel</i> untuk menyimpan data pengguna</p> <p>6 Pengguna dapat memilih pengguna pada daftar pengguna</p> <p>7 Sistem akan menampilkan menu <i>edit</i> atau <i>delete</i> , jika pengguna memilih menu <i>edit</i> maka akan ke langkah 8, jika pengguna memilih menu <i>delete</i> maka akan ke langkah 13</p> <p>8 Sistem akan menampilkan data pengguna terpilih ke <i>form</i> untuk dilakukan perubahan data</p> <p>9 Pengguna melakukan perubahan data pengguna</p>

Usecase mengelola pengguna		
	10	Pengguna menyimpan perubahan data pengguna
	11	Sistem melakukan validasi <i>input</i> -an pengguna jika data valid maka akan ke langkah 12 jika tidak maka akan ke langkah A1
	12	Sistem akan mengubah data pengguna
	13	Pengguna menghapus data pengguna
	14	Sistem akan menampilkan pesan <i>dialog</i> jika pengguna memilih ya maka akan ke langkah 15 jika pengguna memilih tidak maka akan ke langkah 16
	15	Sistem akan melakukan penghapusan data guru mengajar
	16	Sistem batal menghapus guru mengajar
	17	Pengguna juga dapat melakukan pencarian , dengan mengetikan kata kunci pencarian berdasarkan nama , no induk selanjutnya sistem akan kembali ke langkah 1 untuk mengambil data di <i>database</i> jika data ada maka akan ditampilkan jika tidak maka akan kelangkah A2
Alur alternatif	A1	Sistem menampilkan pesan kesalahan <i>input</i> -an pengguna jika pada saat posisi mengubah data maka akan ke langkah 6 jika saat posisi mengubah data maka akan kelangkah 19
	A2	Sistem menampilkan pesan kesalahan data yang dicari tidak ada
Kondisi sukses	1	Pengguna berhasil melakukan menambahkan data guru mengajar
	2	Pengguna berhasil melakukan perubahan data guru mengajar
	3	Pengguna berhasil melakukan penghapusan data guru mengajar

N. Flow of event usecase menampilkan laporan rekap nilai

Flow of event ini digunakan oleh guru untuk melihat rekap nilai yang telah diinputkan oleh guru dalam bentuk tabel seperti terlihat pada tabel 3.15 berikut ini.

Tabel 3.15 *Flow of event usecase* menampilkan laporan rekap nilai.

Usecase melihat rekap nilai siswa	
Nama Usecase	Melihat rekap nilai siswa
Deskripsi singkat	<i>Usecase</i> melihat rekap nilai siswa digunakan untuk melihat hasil rekap nilai pada mata pelajaran
Aktor	Guru
Prasyarat	Pengguna telah <i>login</i> dan memilih menu rekap nilai siswa
Alur Utama	<p>1 Sistem menampilkan data kelas dan mata pelajaran yang diajar oleh guru dari <i>database</i> dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>LessonsModel</i> untuk mengambil data guru yang mengajar</p> <p>2 Pengguna memilih data kelas dan mata pelajaran yang diajar</p> <p>5 Sistem menampilkan data nilai siswa pada matapelajaran tersebut dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>StudentClassModel</i> dan <i>PointModel</i> untuk mengambil data nilai siswa yang diajar oleh guru di kelas tersebut</p> <p>7 Sistem menampilkan data rekap nilai tugas siswa yang didapat dari mengirimkan data ke <i>UserController</i> diteruskan ke <i>PointModel</i> untuk mengambil data nilai siswa</p>
Alur alternatif	A1 Sistem tidak menampilkan data nilai dikarenakan belum ada nilai yang inputkan oleh guru kemudian sistem menampilkan pesan kesalahan selanjutnya sistem kembali pada langkah 3
Kondisi sukses	1 Pengguna berhasil menampilkan laporan data nilai siswa

O. *Flow of event usecase* menampilkan history nilai

Flow of event ini digunakan oleh guru untuk melihat *history* nilai siswa, pada fitur *history* nilai ini pengguna terlebih dahulu mengetikkan nama siswa untuk mencari siswa data siswa tersebut. Setelah itu sistem akan menampilkan data siswa yang dicari. Selanjutnya pengguna memilih siswa tersebut untuk ditampilkan data *history* nilai siswa tersebut seperti terlihat pada gambar 3.16.

Tabel 3.16 Flow of event usecase menampilkan history nilai.

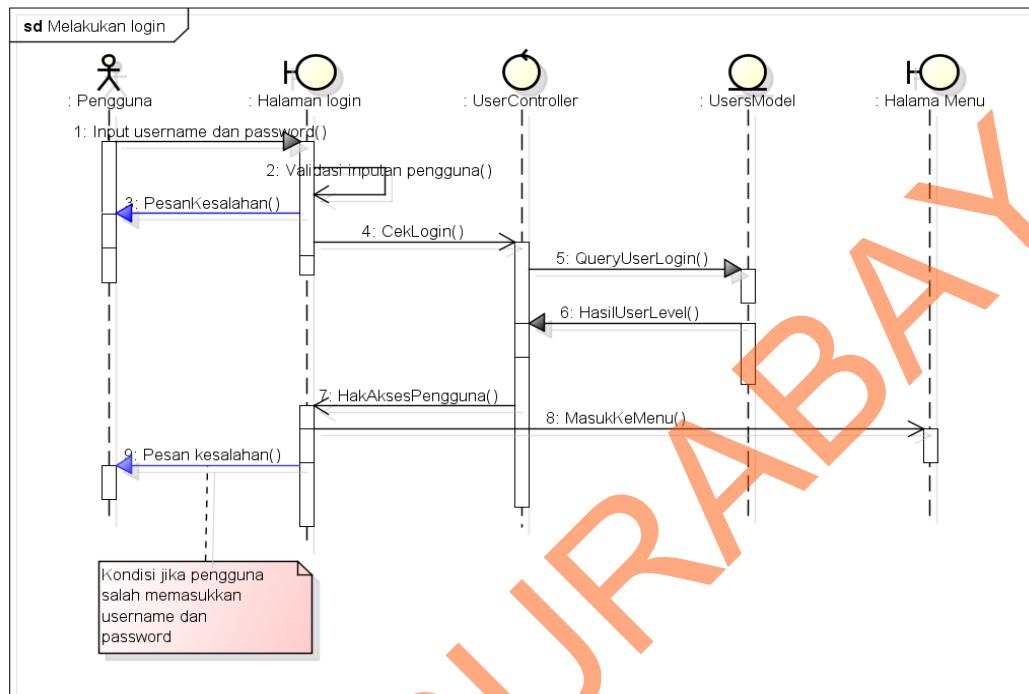
Usecase melihat rekap nilai siswa		
Nama Usecase		Melihat history nilai siswa
Deskripsi singkat		<i>Usecase</i> melihat history nilai siswa digunakan untuk menampilkan daftar nilai yang pernah didapat oleh siswa
Aktor		Guru
Prasyarat		Pengguna telah <i>logindan</i> memilih history nilai <i>siswa</i>
Alur Utama	1	Pengguna mengisikan nama dari siswa
	2	Sistem menampilkan data siswa yang dicari dengan mengirimkan permintaan ke <i>UserController</i> diteruskan ke <i>StudentClassModel</i> untuk mengambil data siswa
Alur alternatif	3	Pengguna memilih siswa
	4	Sistem menampilkan data <i>history</i> nilai siswa tersebut dengan mengirimkan permintaan ke <i>UserController</i> dan diteruskan ke <i>PointModel</i> untuk mengambil data history nilai siswa jika data belum ada maka sistem akan ke alur A1
Alur alternatif	A1	Sistem tidak menampilkan data nilai dikarenakan belum ada nilai yang inputkan oleh guru kemudian sistem menampilkan pesan kesalahan selanjutnya sistem kembali pada langkah 3
Kondisi sukses	1	Pengguna berhasil menampilkan laporan data nilai siswa

3.3.4 Sequence diagram

A. Sequence diagram melakukan login

Pada proses melakukan *login* pengguna menginputkan *username* dan *password* pada aplikasi selanjutnya dari halaman *login* aplikasi akan mengirimkan data ke *controller* untuk melakukan cek *login*. Pada *Usercontroller* data tersebut diteruskan ke *Usermodel* untuk dilakukan pencarian data yang sesuai dengan *username* dan *password* pengguna di *database*. Setelah melakukan *query* hasil akan dikembalikan *Usercontroller*. Pada *Usercontroller* level pengguna tersebut dicek dan diberi keterangan. Selanjutnya sistem akan menentukan apakah berhasil

atau tidak. Jika berhasil maka akan ke halaman menu jika tidak maka akan muncul pesan kesalahan pada pengguna.

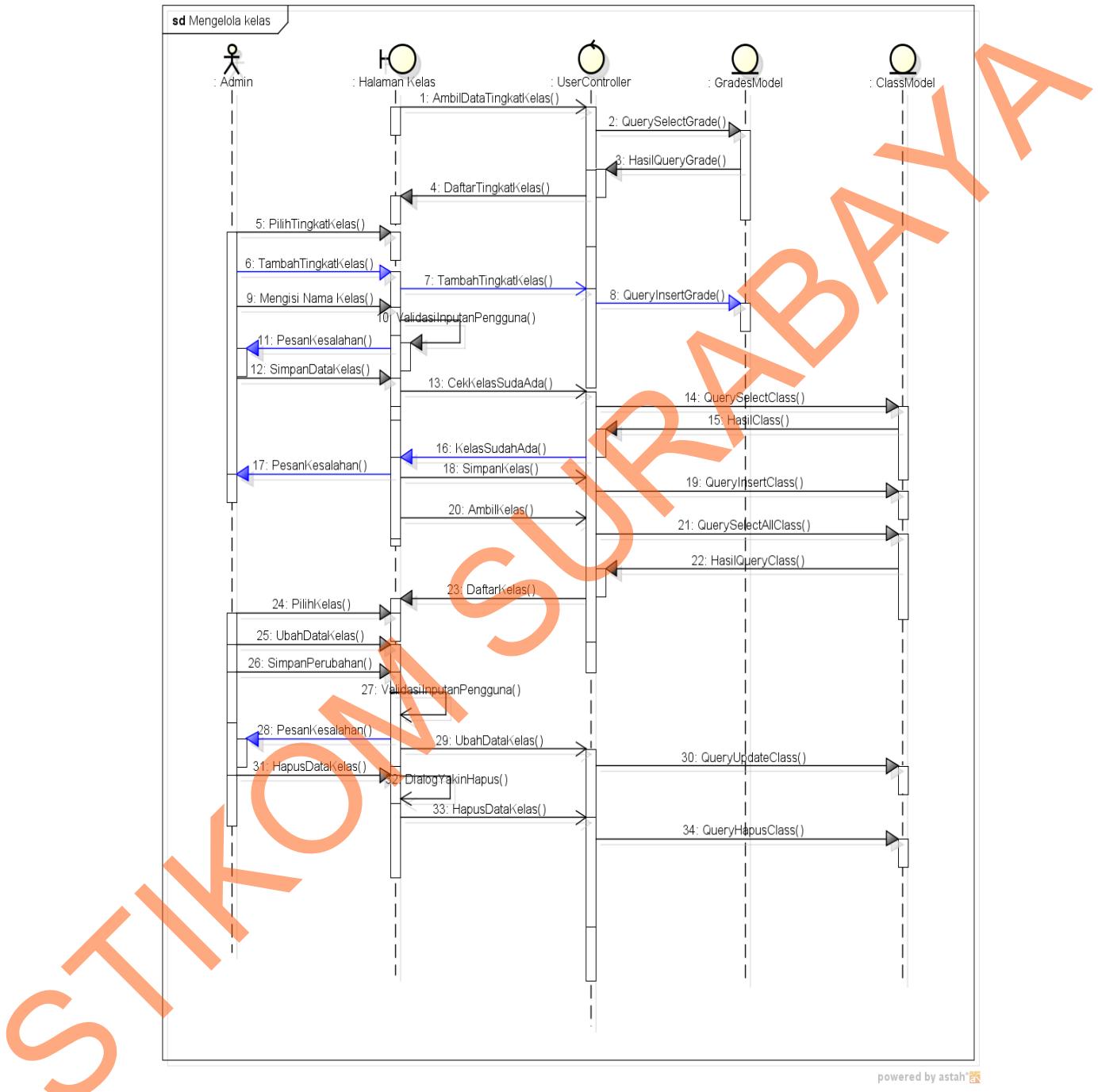


Gambar 3.12 Sequence diagram melakukan login

B. Sequence diagram mengelola kelas

Pada tahapan ini sistem akan menampilkan data tingkat kelas pada awal masuk ke halaman ini. Selanjutnya pengguna dapat memilih data tingkat kelas dan apabila data tingkat kelas tidak ada pengguna dapat menambahkanya. Pada saat ditambahkan sistem akan menyimpan data tingkat kelas yang baru pada *database* kemudian pengguna dapat memilihnya. Selanjutnya pengguna mengisikan nama kelas dan dapat disimpan oleh pengguna. Pada saat menyimpan terjadi pengecekan di *UserController* hal ini dilakukan agar tidak ada nama kelas yang sama pada dan tingkat kelas yang sama jika ada data yang sama maka akan

muncul pesan kesalahan jika tidak ada maka *UserController* akan melanjutkan pengiriman data ke *ClassModel* untuk dilakukan *insert* data ke *database*.



Gambar 3.13 Sequence diagram mengelola kelas

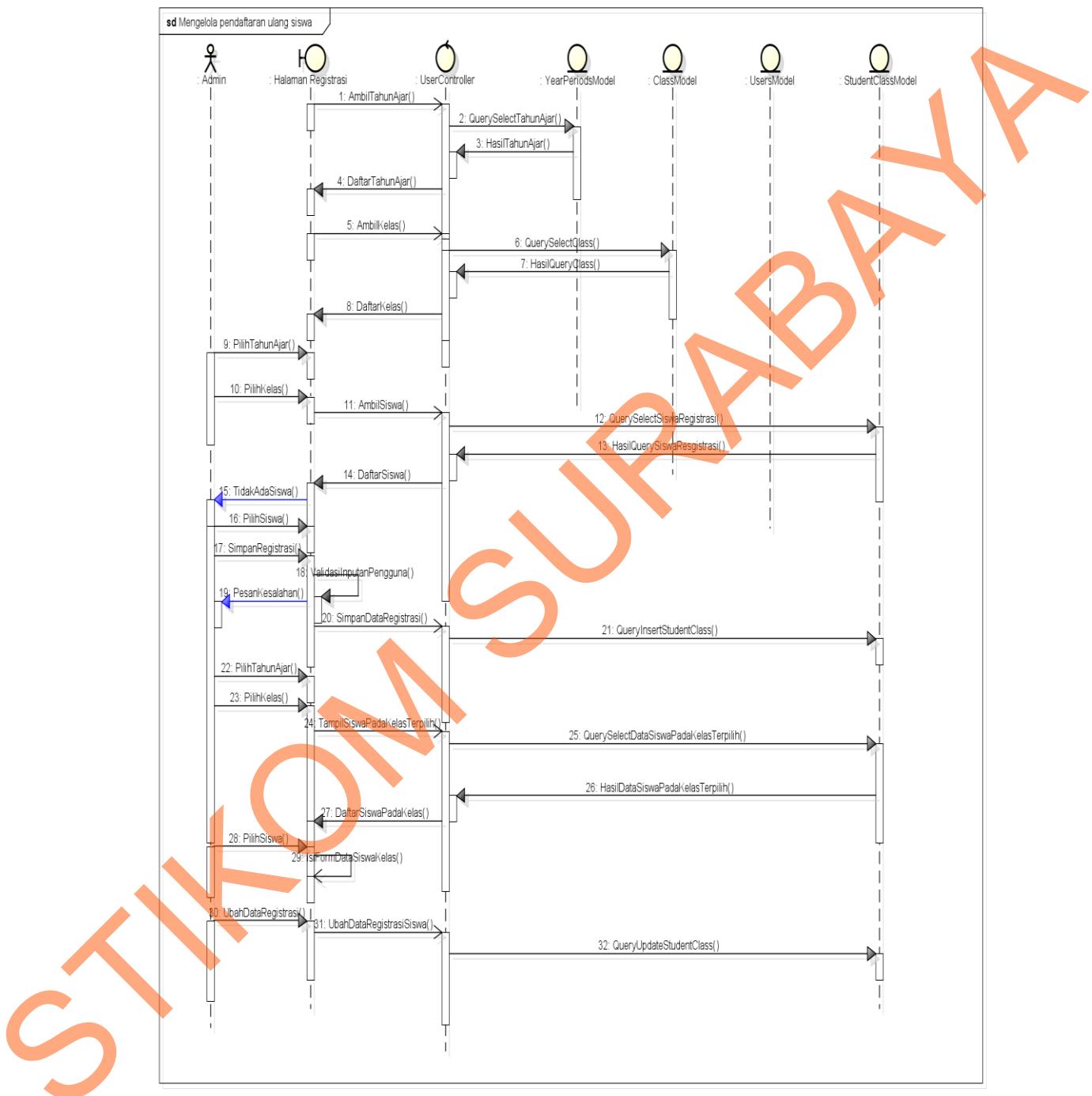
Selanjutnya setelah berhasil melakukan penambahan data kelas sistem akan menampilkan daftar data kelas. Untuk melakukan ubah atau hapus data pengguna

dapat memilih daftar kelas tersebut kemudian muncul menu ubah atau hapus. Untuk ubah pengguna melakukan perubahan di *form* yang disediakan kemudian sistem akan mengirimkan data perubahan tersebut ke *UserController* untuk dilakukan pengecekan apakah data yang dirubah tersebut ada yang sama di *database* jika ada maka akan dikembalikan ke *client* dan muncul pesan kesalahan dari sistem, jika benar maka *UserController* akan meneruskan ke *ClassModel* untuk melakukan perubahan data di *database* pada kelas terpilih. Jika pada menu pengguna memilih hapus maka akan muncul *dialog* untuk pemberitahuan yakin dihapus atau tidak jika memilih ya maka akan dihapus jika tidak sistem tidak melakukan apa-apa.

C. Sequence diagram mengelola pendaftaran ulang siswa

Pada saat pengguna masuk ke menu registrasi sistem akan menampilkan data tahun ajaran dan daftar kelas dari *database*. Selanjutnya pengguna memilih tahun ajar dan kelas yang akan dimasuki oleh siswa. Pada saat tersebut sistem akan melakukan permintaan ke *UserController* untuk mengambil data siswa yang belum memiliki kelas di tahun ajaran tersebut melalui *StudentClassModel* yang dilanjutkan melakukan *query* di *database*. Selanjutnya data tersebut akan dikembalikan ke *StudentClassmodel* dan diteruskan ke *UserController* dan ke *client* untuk di tampilkan. Jika tidak ada siswa karena semua siswa sudah tersdaftar maka sistem akan menampilkan pesan tidak ada siswa. Pengguna selanjutnya dapat memilih siswa pada daftar siswa kemudian menyimpan pendaftaran siswa tersebut. Pada proses menyimpan sistem akan mengirim data ke

UserController kemudian diteruskan ke *StudentClassModel* untuk dilakukan *insert* data ke database.

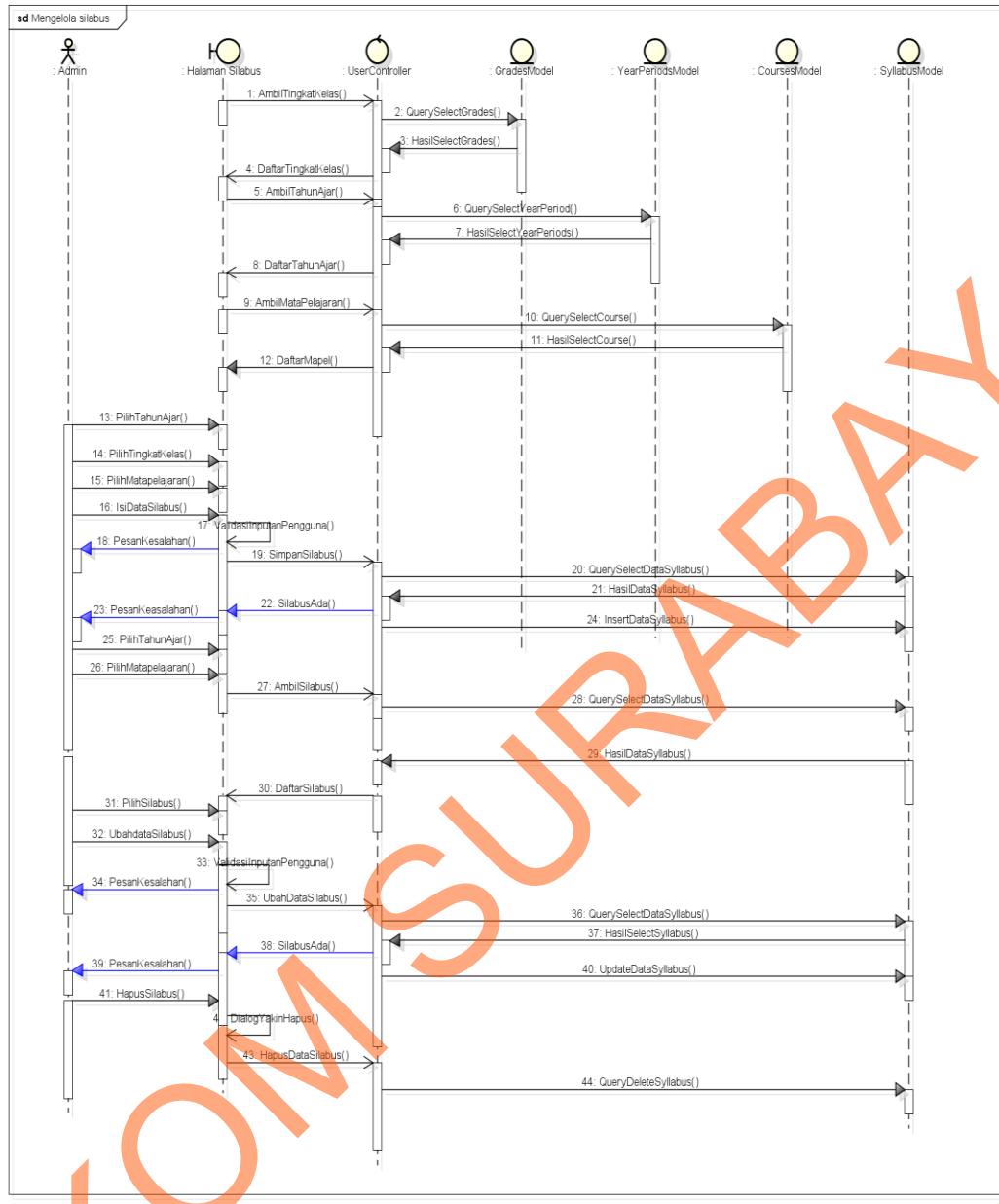


Gambar 3.14 Sequence diagram mengelola pendaftaran ulang siswa.

Untuk menampilkan data yang sudah didaftarkan pengguna dapat memilih tahun ajaran dan kelas kemudian akan muncul data siswa pada kelas tersebut. Untuk melakukan perubahan pengguna dapat memilih data siswa tersebut maka *form* untuk merubah data akan terbuka dan sudah siap dengan data yang akan diubah. Selanjutnya pengguna melakukan perubahan data tersebut. Setelah melakukan perubahan klik ubah data akan dikirimkan ke *UserController* selanjutnya diteruskan ke *StudentClassModel* untuk dilakukan perubahan data di *database*.

D. Sequence diagram mengelola silabus

Pada proses mengelolah silabus pengguna dapat memilih menu silabus. Ketika masuk pada menu tersebut sistem akan menampilkan data tahun ajaran, tingkat kelas, dan matapelajaran. Selanjutnya pengguna dapat memilih tahun ajaran, tingkat kelas dan matapelajaran tersebut serta mengisi data silabus yaitu keterangan dan pertemuan ke-berapa. Pada saat akan menyimpan dilakukan validasi *input*-an pengguna jika terdapat kesalahan maka akan menampilkan pesan kesalahan. Selanjutnya data akan dikirim ke *UserController* yang diteruskan ke *Syllabusmodel* untuk dilakukan pengecekan apakah data tersebut sudah pernah diinputkan. Setalah *query* ke *database* kemudian *Syllabusmodel* akan mengembalikan ke *UserController* jika data tersebut pernah diinputkan maka akan dikembalikan ke *client* untuk menampilkan kesalahan, jika belum akan diteruskan kembali ke *Syllabusmodel* untuk dilakukan *insert* data ke *database*.



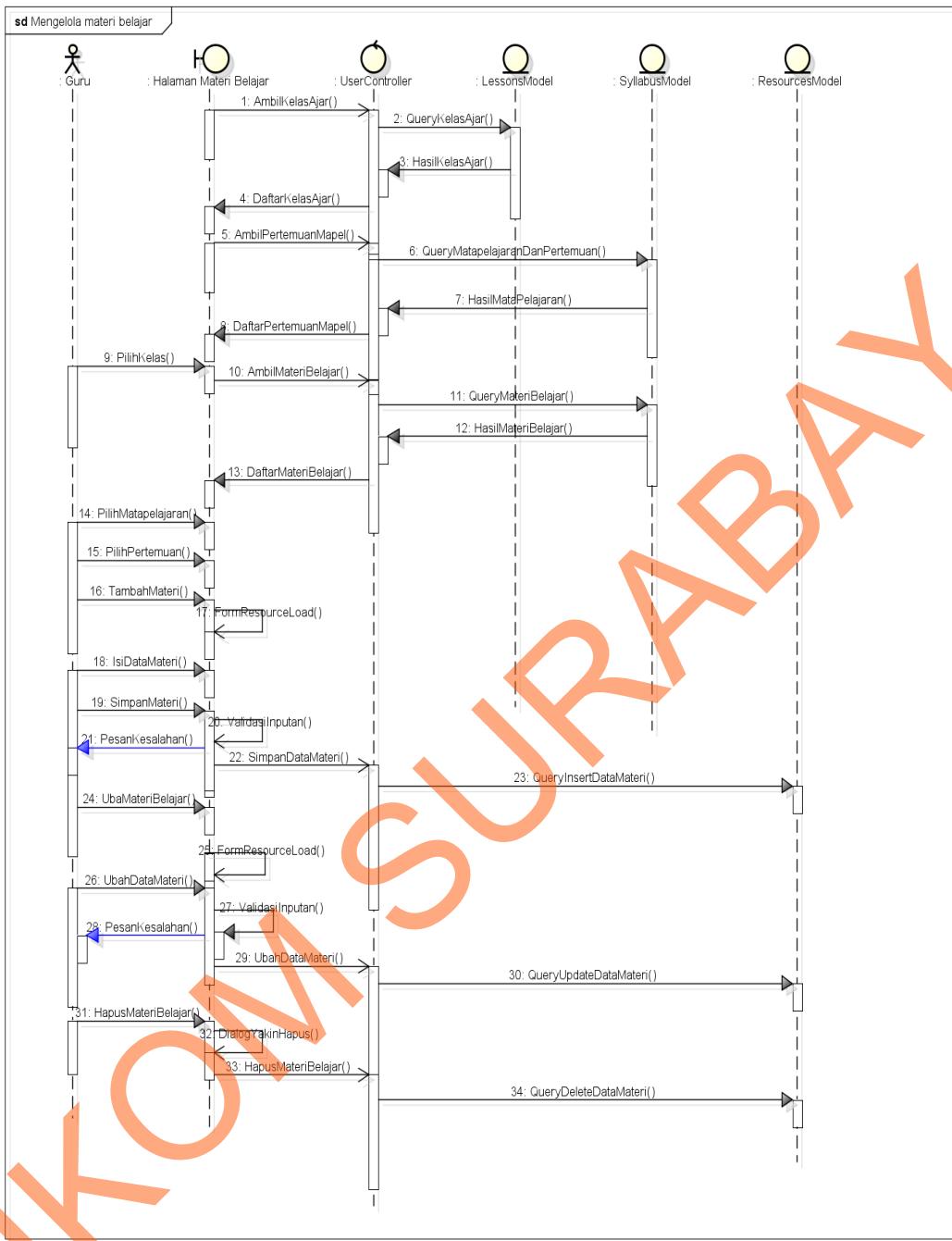
Gambar 3.15 Sequence diagram mengelola silabus.

Untuk menampilkan data silabus yang sudah dibuat pengguna dapat memilih tahun ajaran dan matapelajaran selanjutnya sistem akan menampilkan data berdasarkan tahun ajaran dan matapelajaran terpilih. Pengguna melakukan perubahan di pada *form* selanjutnya di klik ubah maka sistem akan melakukan validasi *input-an* pengguna, jika terjadi kesalahan maka akan muncul pesan kesalahan jika tidak maka akan diteruskan ke *UserController* yang kemudian

diteruskan ke *Syllabusmodel* untuk dilakukan pengecekan apakah data tersebut sudah pernah diinputkan jika sudah maka dikembalikan dan muncul pesan kesalahan di *client* jika tidak maka data akan diubah di *database*. Untuk melakukan penghapusan data silabus pengguna dapat memilih menu hapus pada saat memilih daftar silabus kemudian sistem akan menampilkan *form* untuk menanyakan apakah data yakin dihapus, jika pengguna memilih ya maka data akan dihapus di database, jika pengguna memilih tidak sistem membatalkan penghapusan data.

E. Sequence diagram mengelola materi belajar

Pada saat pengguna masuk ke menu materi belajar sistem akan menampilkan data kelas yang diajat oleh guru beserta pertemuan yang sesuai dengan silabus. Selanjutnya pengguna akan memilih kelas dan sistem akan mengambil materi belajar yang telah di berikan pada kelas tersebut. Pengguna selanjutnya dapat meng-expand matapelajaran maka akan muncul daftar pertemuan yang sudah ditetapkan oleh silabus beserta materi yang telah disimpan dan diteruskan dengan memilih pertemuan tersebut. Untuk menambahkan materi baru pengguna dapat memilih tambahkan materi kemudia *form* akan muncul. Pengguna dapat mengisikan data materi pada *form* tersebut dan melakukan pemilihan materi yang akan di berikan dan selanjutnya pengguna menyimpan data materi tersebut. Pada saat menyimpan sistem akan memvalidasi *input-an* pengguna jika terjadi kesalahan maka akan muncul pesan kesalahan jika tidak maka data materi akan disimpan di *database*.



Gambar 3.16 Sequence diagram mengelola materi belajar.

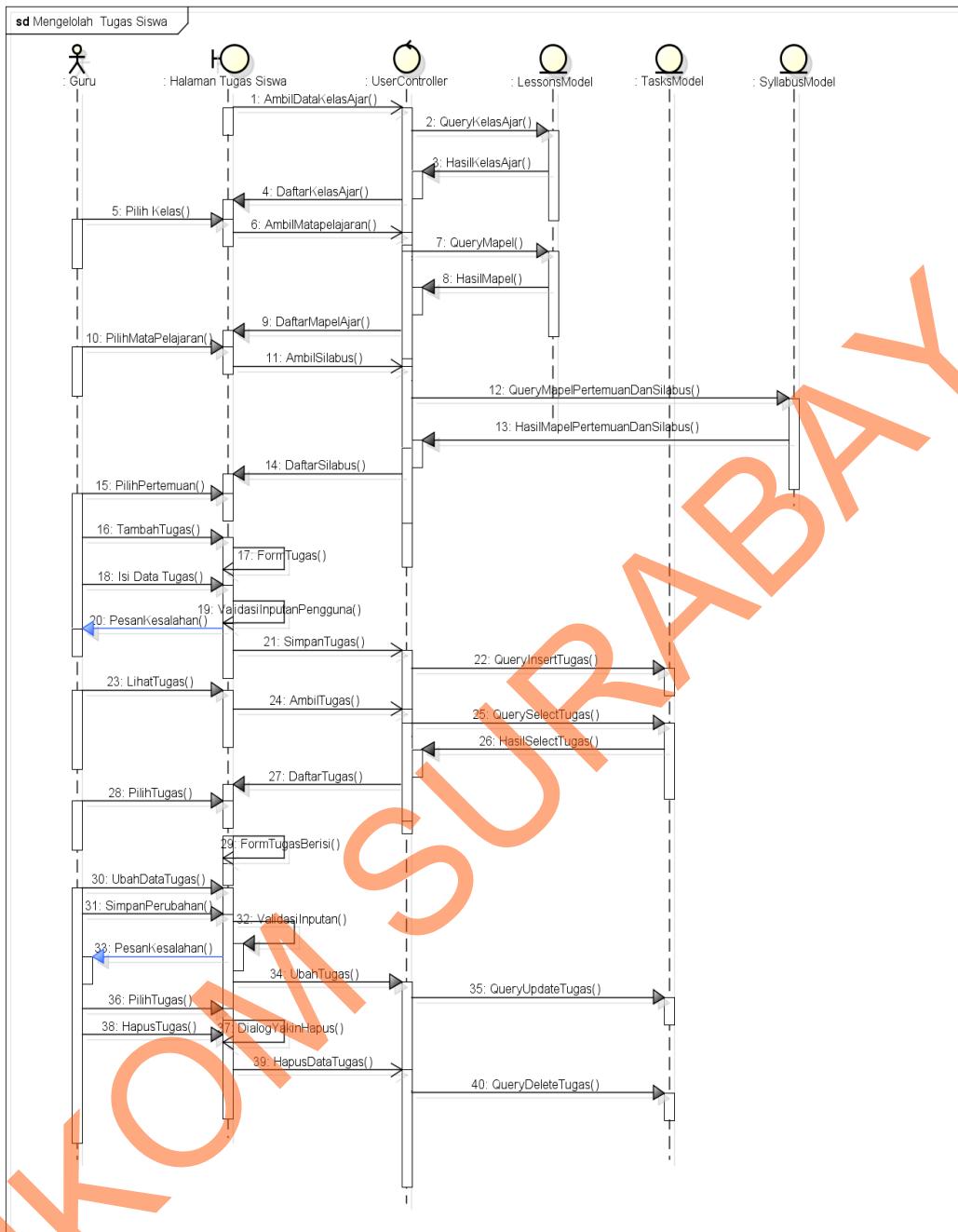
Untuk melakukan perubahan pengguna dapat memilih ubah untuk melakukan perubahan data materi. Ketika memilih ubah *form* akan muncul dan berisikan data materi, selanjutnya pengguna melakukan perubahan dan menyimpan perubahan data tersebut. Sistem akan mengirimkan data perubahan

tersebut ke *UserController* dan diteruskan ke *ResourcesModel* untuk melakukan *query* perubahan data di *database*. Untuk menghapus materi tersebut data pengguna dapat memilih hapus kemudian sistem akan menampilkan *dialog* apakah yakin untuk dihapus, jika pengguna memilih ya maka data akan dihapus jika tidak maka sistem akan membatalkan penghapusan data tersebut.

F. Sequence diagram mengelola tugas

Pada saat pengguna masuk ke menu kelola tugas sistem akan menampilkan daftar kelas yang diajar oleh pengguna. Selanjutnya pengguna dapat memilih kelas tersebut dan sistem akan menampilkan daftar mata pelajaran yang diajar oleh pengguna dikelas tersebut. Selanjutnya pengguna memilih data matapelajaran yang ada di pertemuan tersebut dan sistem akan menampilkan daftar pertemuan sesuai dengan silabus yang dibuat. Untuk menambah tugas pengguna dapat memilih tambah tugas kemudian pengguna dapat mengisi *form* untuk data tugas tersebut. Pada saat pengguna menyimpan data tugas tersebut maka sistem akan melakukan validasi *input*-an pengguna jika terdapat kesalahan maka sistem akan menampilkan pesan kesalahan. Jika tidak maka sistem akan melakukan request ke *UserController* dan diteruskan ke *TasksModel* dan melakukan penyimpanan data di *database*. Selanjutnya sistem menampilkan daftar tugas yang telah dibuat.

Untuk melakukan perubahan data tugas pengguna dapat memilih tugas di daftar tugas maka akan muncul menu untuk menghapus dan mengubah. Pengguna selanjutnya memilih mengubah kemudian sistem akan menampilkan data tugas terpilih ke *form* untuk dilakukan perubahan.



Gambar 3.17 Sequence diagram mengelola tugas.

Setalah melakukan perubahan data pengguna dapat menyimpan perubahan tersebut dan sistem akan memvalidasi *input*-an pengguna sebelum memberikan ke *UserController* lalu diteruskan ke *TasksModel* untuk dilakukan perubahan data di *database*. Untuk melakukan penghapusan data pengguna dapat memilihhapus

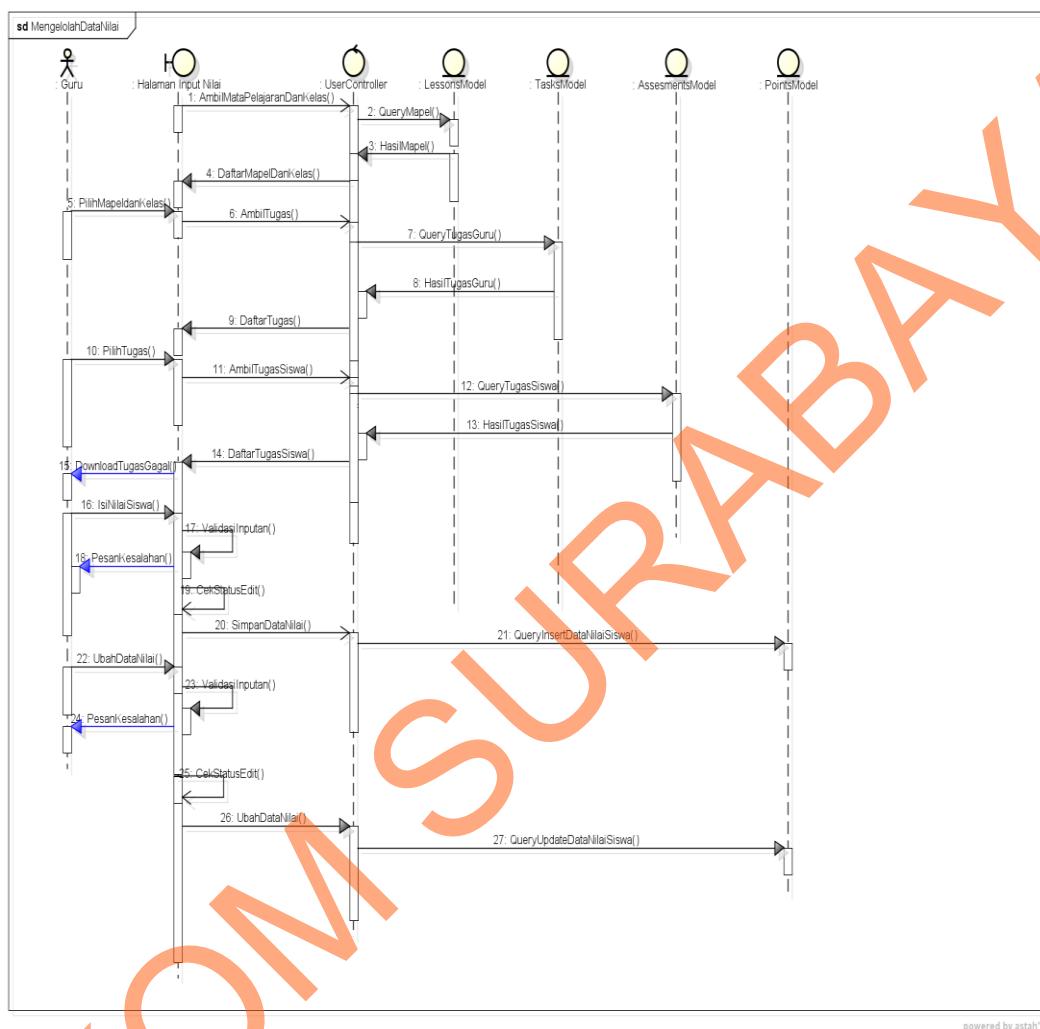
kemudian sistem akan menampilkan *dialog* apakah yakin akan menghapus data jika pengguna memilih ya maka sistem akan menghapus data tersebut jika tidak maka sistem membatalkan penghapusan data tersebut.

G. Sequence diagram mengelola data nilai siswa

Pada saat pengguna masuk ke menu kelola nilai siswa maka sistem akan menampilkan data matapelajaran yang diajar oleh guru tersebut beserta kelasnya. Selanjutnya pengguna dapat memilih mata pelajaran dan kelas tersebut untuk menampilkan tugas yang pernah diberikan oleh guru pada siswa tersebut. Selanjutnya pengguna dapat memilih tugas tersebut dan menampilkan data siswa pada kelas tersebut beserta tugas yang telah dikumpulkan oleh siswa tersebut. Jika terdapat file yang dikumpulkan siswa pengguna dapat men-*download* tugas siswa tersebut. Kemudian pengguna dapat mengisi data nilai siswa pada *input-an* nilai jika data nilai belum terisi maka sistem akan berstatus menambahkan nilai baru. Selanjutnya pengguna mengisikan data nilai tersebut. Pada saat menyimpan data nilai tersebut sistem akan melakuakn validasi *input-an* pengguna jika terdapat kesalahan maka sistem akan menampilkan kesalahan. Jika benar maka data akan dikirim ke *UserController* selanjutnya diteruskan ke *PointsModel* untuk dilakukan penyimpanan data nilai siswa.

Jika pada saat menampilkan data siswa beserta tugas sudah terdapat nilai maka sistem akan berstatus *edit* ini menunjukkan sudah ada nilai yang diinputkan oleh pengguna dan pengguna dapat melakukan perubahan data tersebut. Setelah melakukan perubahan data tersebut pengguna menyimpan perubahan datat tersebut dan sistem akan melakukan validasi pada *input-an* pengguna sebelum di

kirim data tersebut ke *UserController* dan *PointsModel* untuk dilakukan perubahan data di *database*.

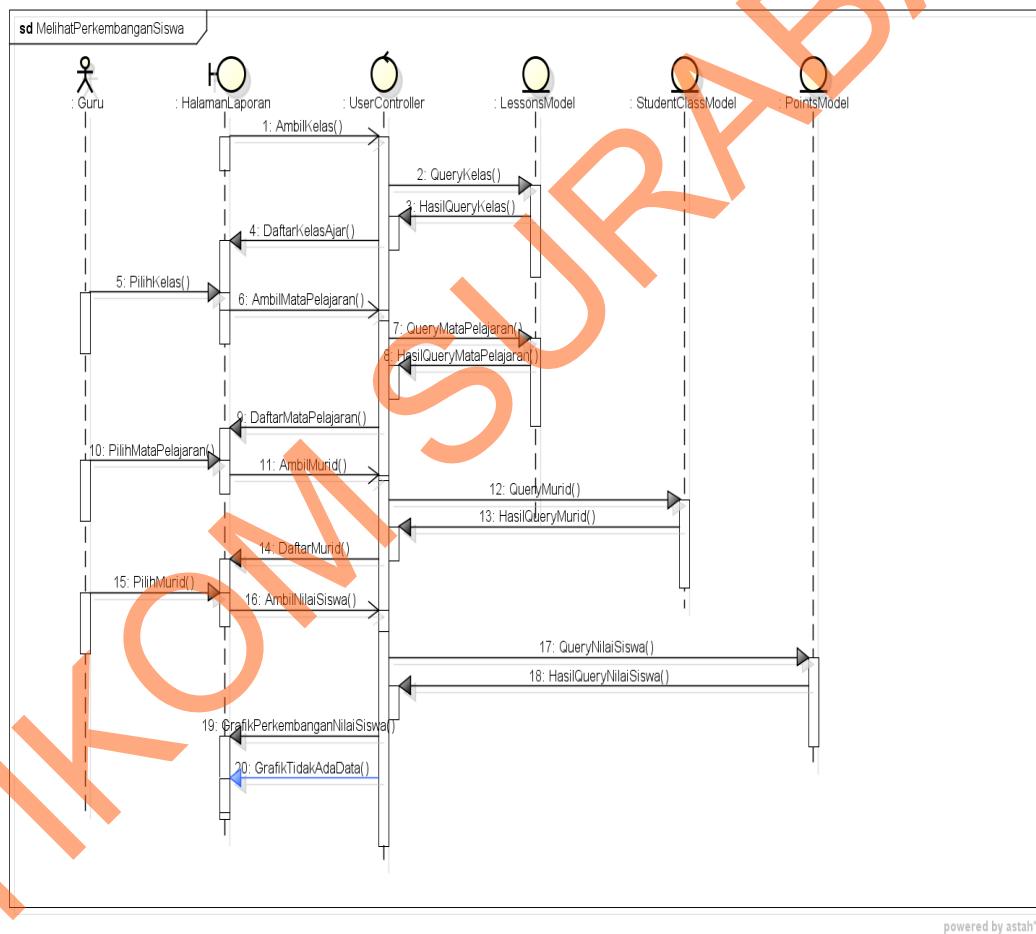


Gambar 3.18 Sequence diagram mengelola data nilai siswa.

H. Sequence diagram melihat perkembangan nilai siswa

Pada saat pengguna masuk ke menu laporan perkembangan siswa maka sistem akan menampilkan daftar kelas yang diajar oleh pengguna . Selanjutnya pengguna dapat memilih kelas tersebut dan menampilkan daftar mata pelajaran yang diajar dikelas tersebut. Pengguna dapat memilih mata pelajaran tersebut

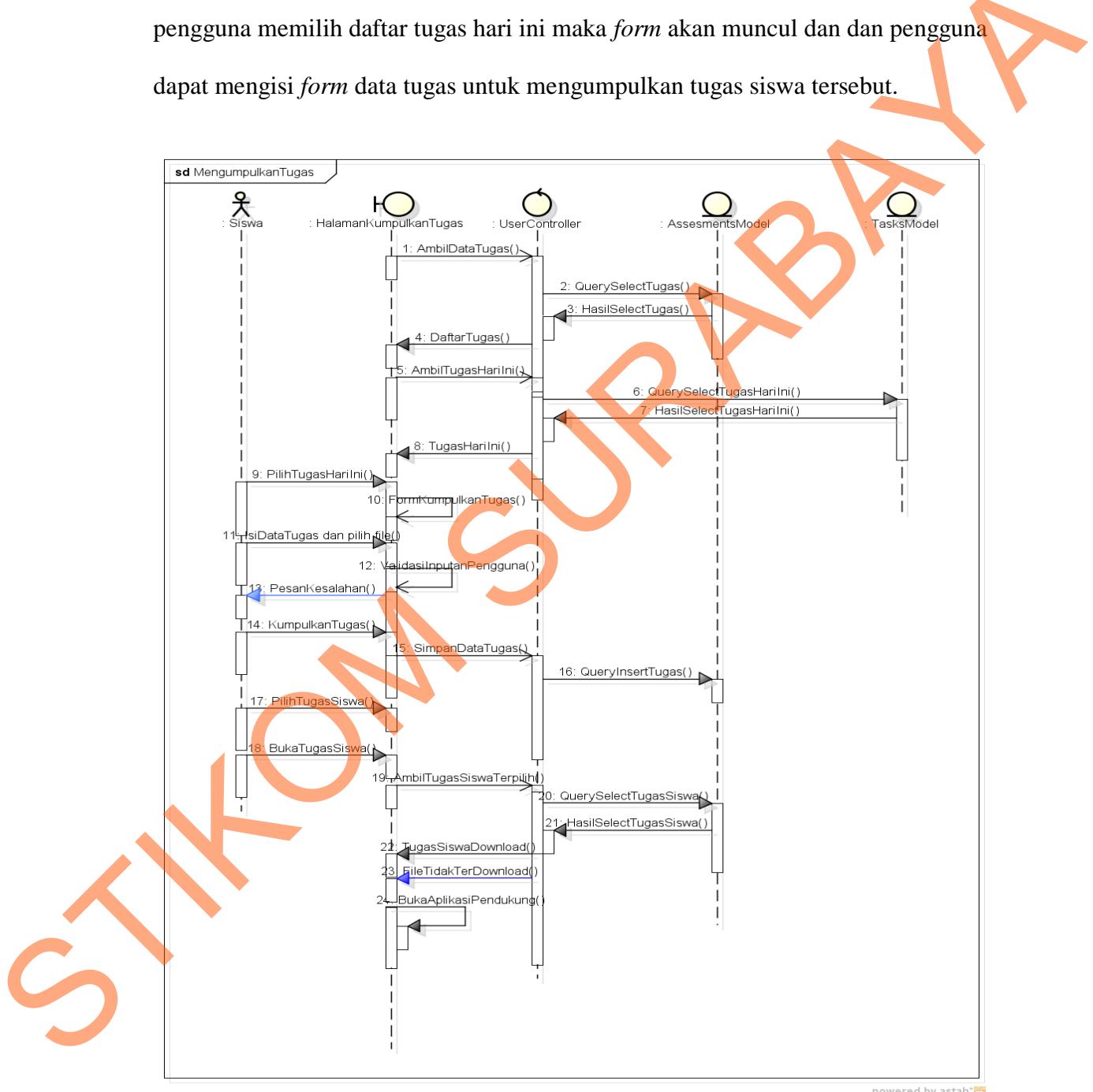
untuk menampilkan daftar siswanya dan dilanjutkan memilih siswa pada daftar siswa tersebut untuk menampilkan data nilai siswa. Pada saat pengguna memilih siswa tersebut sistem akan melakukan *request* ke *UserController* yang diteruskan ke *PointsModel* untuk menampilkan data nilai siswa, kemudia hasil dari *query* tersebut akan dikembalikan ke *client* dan ditampilkan grafik nilai berdasarkan nilai tersebut. Jika tidak ada nilai yang dimasukkan maka sistem akan menampilkan pesan grafik tidak ada data nilai.



Gambar 3.19 Sequence diagram melihat perkembangan nilai siswa.

I. Sequence diagram mengumpulkan tugas siswa

Pada saat pengguna masuk ke menu awal sistem akan menampilkan data tugas hari ini dan daftar tugas yang pernah dikumpulkan siswa. Kemudian pengguna memilih daftar tugas hari ini maka *form* akan muncul dan pengguna dapat mengisi *form* data tugas untuk mengumpulkan tugas siswa tersebut.

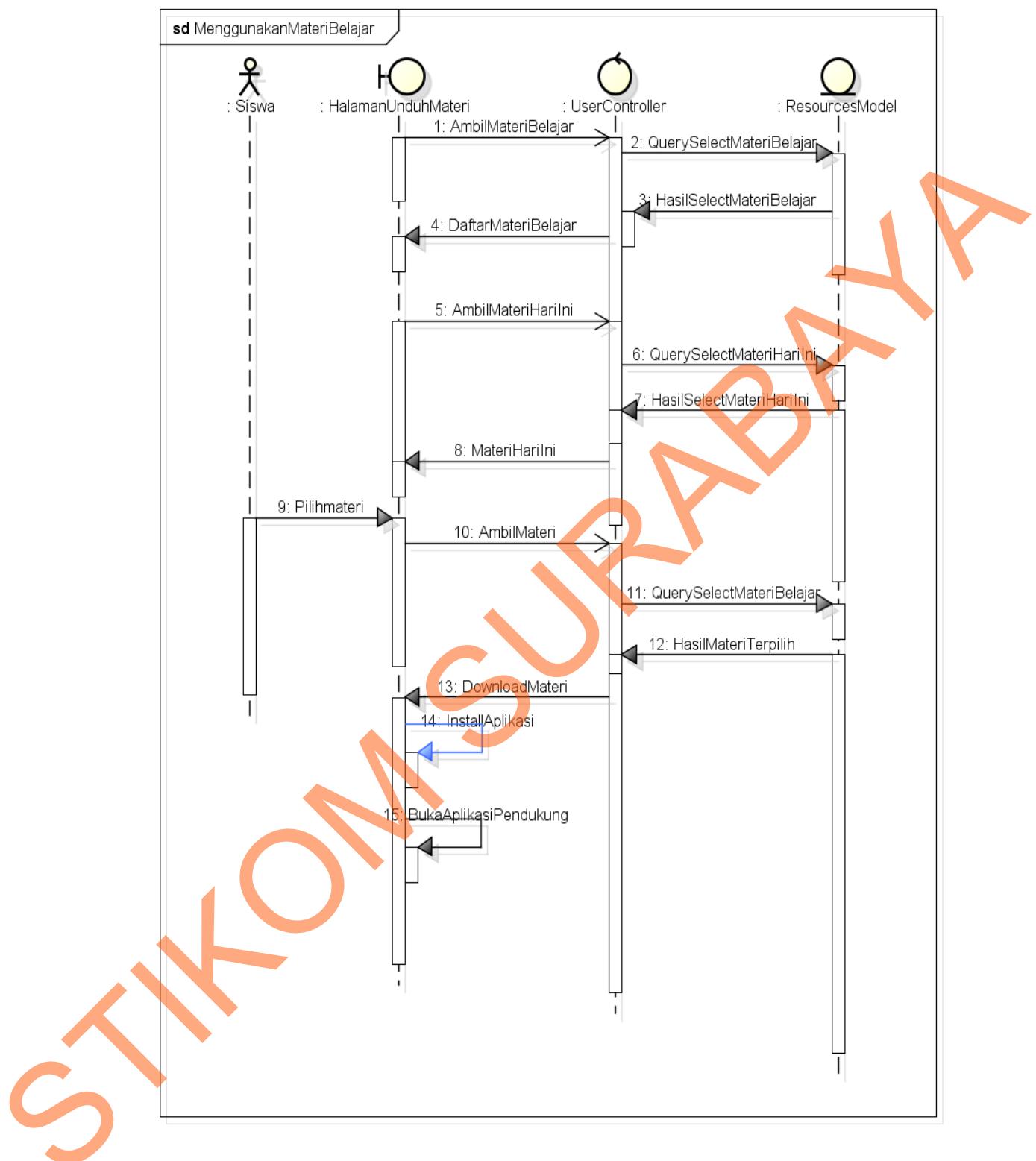


Gambar 3.20 Sequence diagram mengumpulkan tugas siswa.

Selanjutnya pengguna akan menyimpan data tugas tersebut sistem akan melakukan validasi data tugas tersebut jika terdapat pesan kesalahan maka sistem akan menampilkan pesan kesalahan. Jika *input*-an sudah benar maka sistem akan mengirim data tersebut ke *UserController* untuk diteruskan ke *AssesmentModel* dan dilakukan penyimpanan data tugas siswa di *database*. Untuk membuka tugas yang dikumpulkan pengguna dapat memilih daftar tugas yang dikumpulkan kemudian pengguna memilih buka selanjutnya sistem akan melakukan *request* ke *UserController* untuk men-download tugas tersebut selanjutnya tugas dapat dibuka sesuai dengan aplikasi pendukung pada komputer siswa.

J. Sequence diagram menggunakan materi belajar siswa

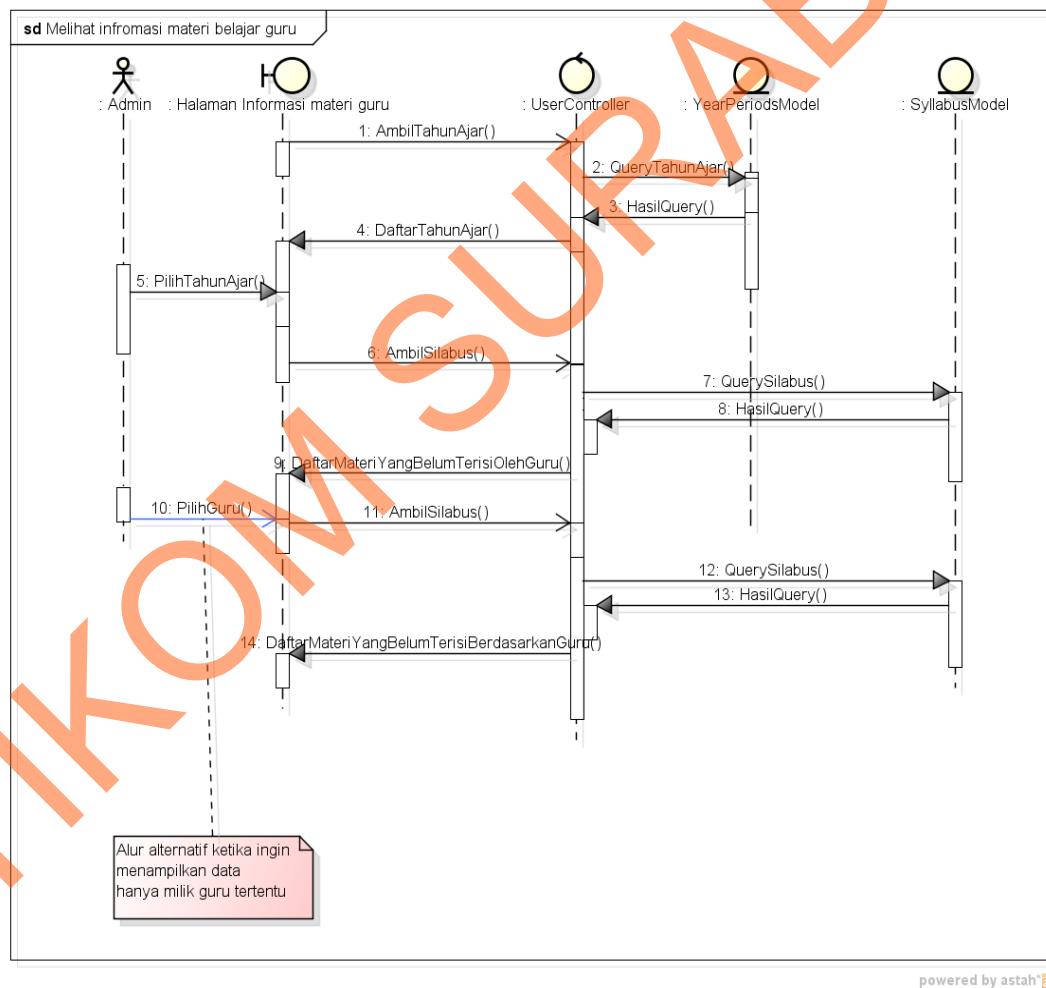
Pada saat siswa masuk ke menu utama sistem akan menampilkan daftar materi belajar keseluruhan dan materi yang dipakai untuk hari ini. Selanjutnya pengguna dapat memilih materi belajar tersebut kemudian sistem akan mengirim data ke *UserController* untuk men-download tugas tersebut setelah di *download* oleh pengguna materi tersebut dapat dibuka sesuai dengan aplikasi pendukung yang ada. Jika belum ada aplikasi pendukung maka pengguna akan melakukan instalasi untuk aplikasi pendukung tersebut. Seperti terlihat pada gambar 3.17 berikut ini.



Gambar 3.21 Sequence diagram menggunakan materi belajar siswa.

K. Sequence diagram melihat informasi materi belajar guru

Pada saat pengguna masuk pada menu melihat materi guru maka sistem akan menampilkan daftar tahun ajaran. Kemudian pengguna memilih tahun ajaran tersebut dan sistem akan mengirimkan data ke *UserController* untuk mengambil data materi yang belum dikumpulkan oleh guru berdasarkan silabus pada tahun tersebut.

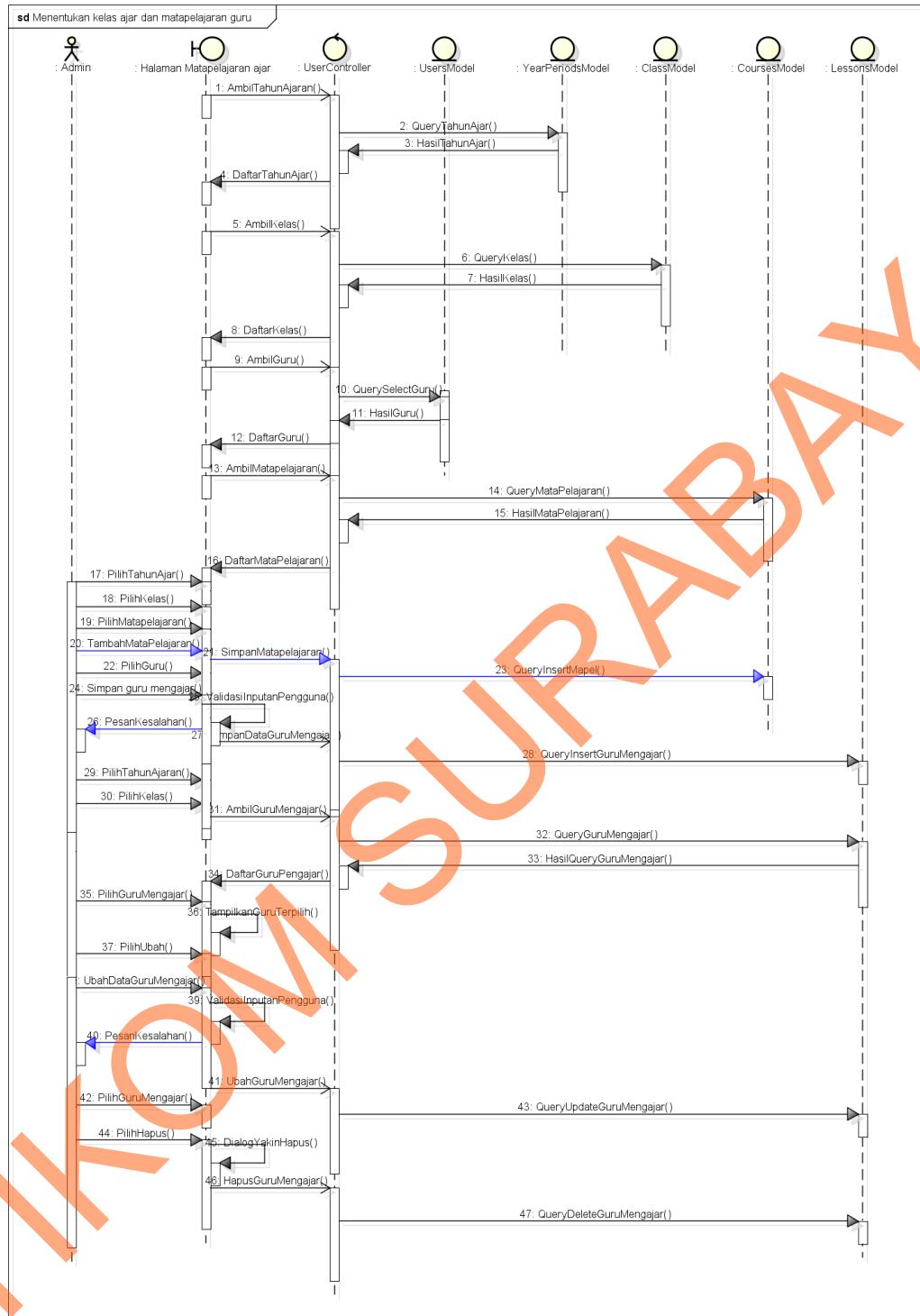


Gambar 3.22 *Sequence diagram* melihat informasi materi belajar guru.

Setelah *SyllabusModel* melakukan *query* di *database* hasilnya akan dikembalikan ke *controller* untuk dikembalikan ke *client* untuk ditampilkan. Selanjutnya pengguna dapat melakukan *filter* berdasarkan nama guru yang bersangkutan untuk menampilkan daftar materi yang tidak dibuat oleh guru tersebut.

L. Sequence diagram menentukan kelas ajar dan matapelajaran yang diajar guru

Pada saat pengguna masuk ke menu guru mengajar sistem akan menampilkan daftar tahun ajaran, data kelas, data guru, dan data mata pelajaran. Kemudian pengguna dapat memilih tahun ajaran, kelas dan guru yang akan mengajar beserta matapelajarannya. Jika matapelajaran tidak ada maka pengguna dapat menambahkan matapelajaran tersebut pada saat memilih *form* tersebut. Selanjutnya pengguna dapat menyimpan guru mengajar tersebut sistem akan mengirim data ke *UserController* dan diteruskan ke *LessonsModel* untuk menyimpan data tersebut. Untuk menampilkan data guru mengajar pengguna dapat memilih tahun ajaran dan kelas kemudian sistem akan menampilkan data guru yang mengajar pada kelas tersebut beserta matapelajaran yang diajar. Untuk mengubah data guru mengajar pengguna dapat memilih guru tersebut kemudian pilih menu *edit* dan sistem akan menampilkan *form* untuk mengubah data tersebut. Setelah melakukan perubahan tersebut sistem akan mengirimkan data ke *UserController* dan diteruskan ke *LessonsModel* untuk melakukan perubahan data.



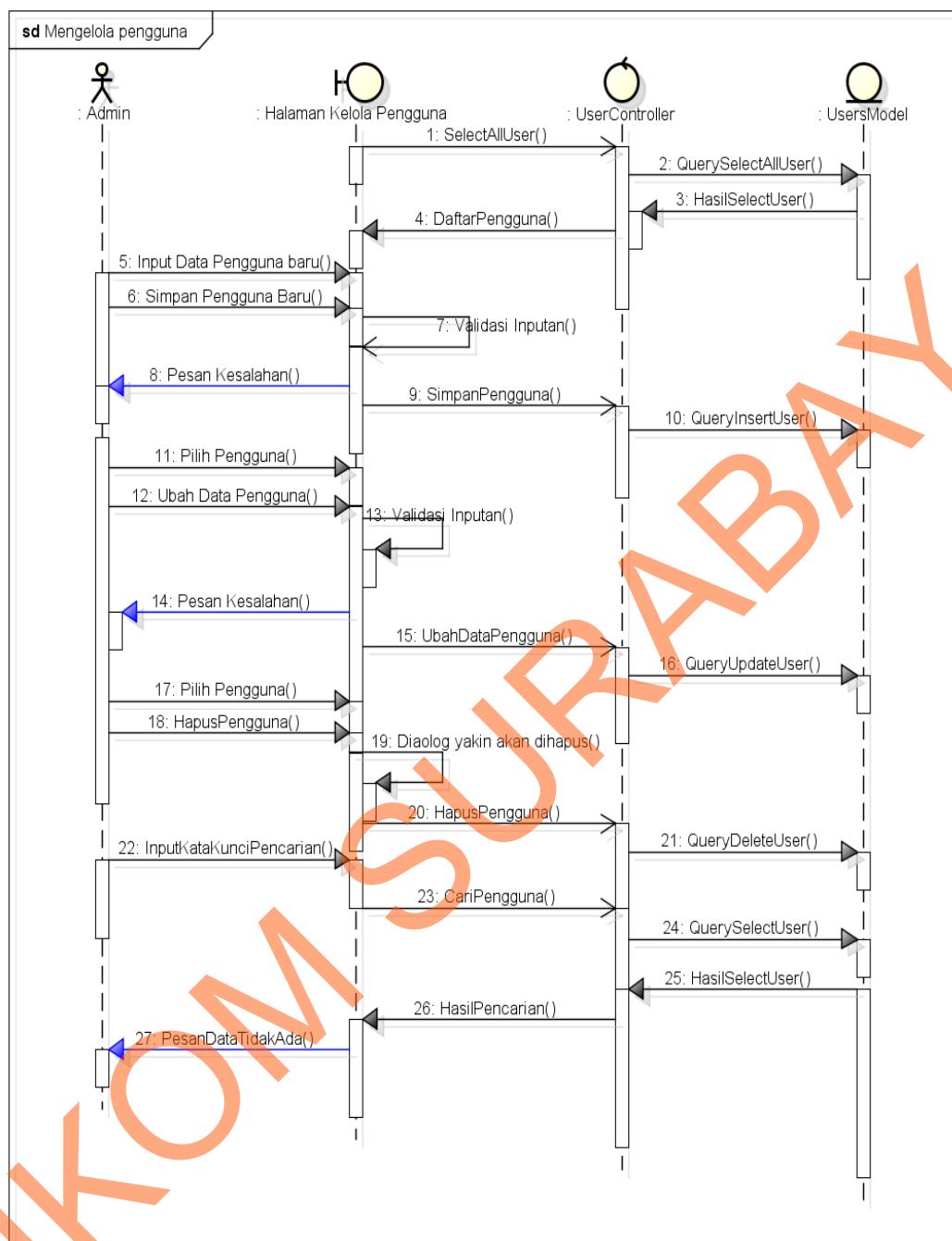
Gambar 3.23 Sequence diagram menentukan kelas ajar dan matapelajaran yang diajarkan oleh guru.

Untuk melakukan penghapusan data pengguna dapat guru tersebut dan memilih menu *delete* kemudian sistem akan memunculkan *dialog* apakah yakin

akan menghapus data tersebut, jika pengguna memilih ya maka sistem akan melakukan *request* ke *UserController* diteruskan ke *LessonsModel* dan melakukan penghapusan data namun jika pengguna memilih tidak maka sistem akan membatalkan penghapusan data tersebut.

M. Sequence diagram mengelola pengguna

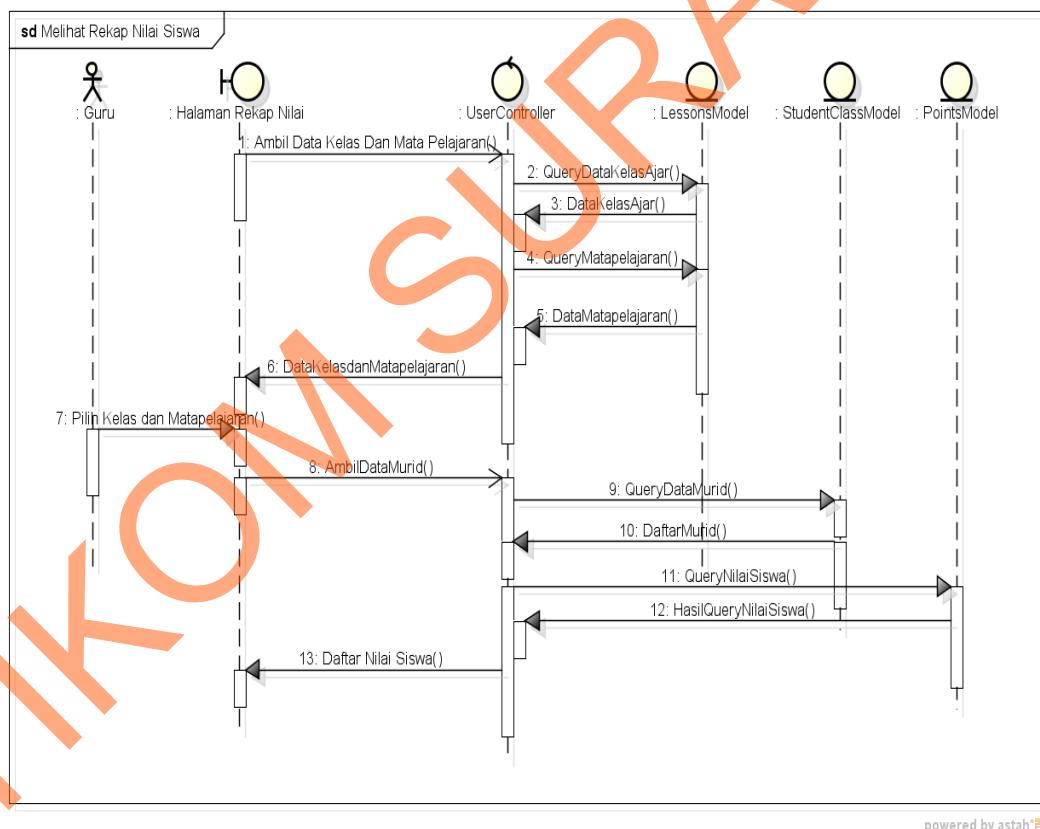
Pada saat admin masuk ke menu pengguna maka sistem akan menampilkan data pengguna dengan cara mengirimkan permintaan ke *UserController* dan diteruskan ke *UserModel* untuk melakukan *query* data pengguna di *database*. Selanjutnya setelah mendapatkan hasil *query* sistem akan menampilkan daftar pengguna. Untuk menambahkan pengguna baru admin dapat menambahkan dengan mengisikan data pengguna pada *form* input pengguna, pada saat admin akan menyimpan data pengguna baru sistem akan memvalidasi *input*-an pengguna jika terjadi kesalahan maka akan muncul pesan kesalahan dari sistem, jika tidak maka akan dilakukan penyimpanan data pengguna baru oleh sistem. Untuk melakukan perubahan data admin dapat memilih pengguna tersebut di daftar pengguna yang sudah ditampilkan. Setelah memilih data pengguna data tersebut akan masuk ke *form* untuk perubahan data. Pada saat admin akan menyimpan perubahan data tersebut sistem akan melakukan validasi pada *input*-an pengguna jika terjadi kesalahan maka sistem akan menampilkan pesan kesalahan, jika tidak maka sistem akan menyimpan data perubahan tersebut.



Gambar 3.24 Sequence diagram menampilkan rekap nilai.

N. Sequence diagram melihat rekap nilai siswa

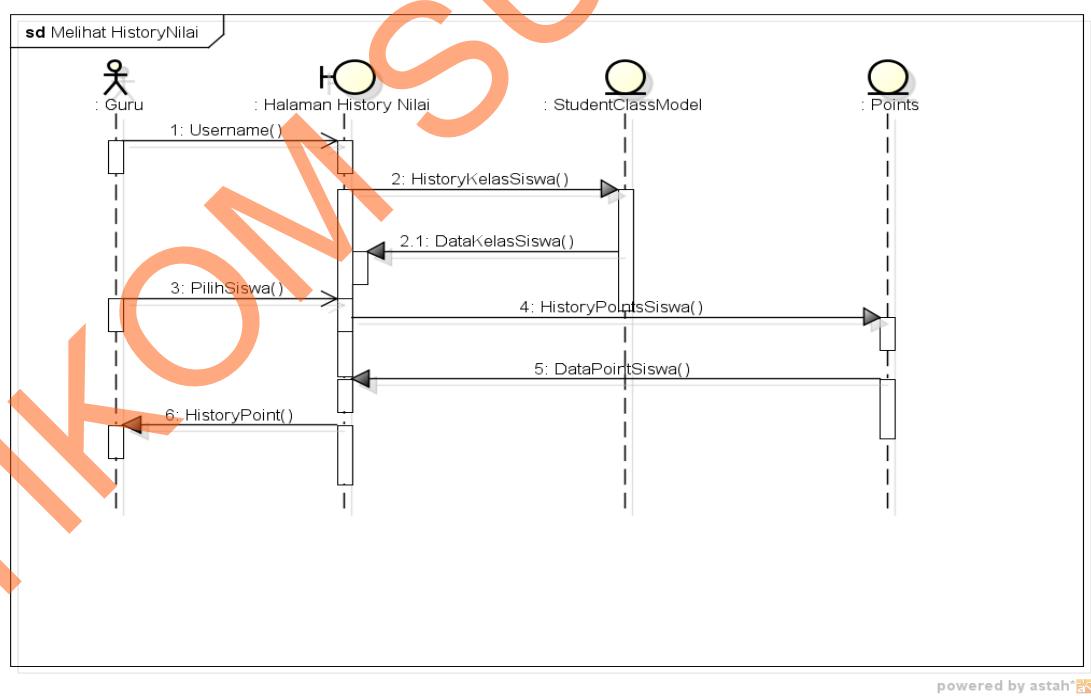
Pada saat pengguna masuk ke menu rangkuman nilai siswa maka sistem akan menampilkan daftar kelas dan mata pelajaran yang diajar oleh pengguna . Pada saat pengguna memilih matapelajaran tersebut sistem akan melakukan *request* ke *UserController* yang diteruskan ke *PointModel* untuk menampilkan data nilai siswa, kemudia hasil dari *query* tersebut akan dikembalikan ke *client* dan ditampilkan rekap nilai tersebut. Jika tidak ada nilai yang dimasukkan maka sistem akan menampilkan pesan grafik tidak ada data nilai.



Gambar 3.25 Sequence diagram menampilkan rekap nilai.

O. Sequence diagram melihat history nilai siswa

Untuk melihat *history* nilai siswa, pengguna mengetikkan nama dari siswa yang akan dilihat datanya. Pada saat pengguna mengetikkan nama sistem akan mengirimkan permintaan ke *UserController* dan diteruskan ke *StudentClassModel* untuk mengambil data siswa. Setalah data muncul pengguna dapat memilih siswa yang akan dilihat nilainya. Pada saat pengguna memilih siswa tersebut sistem akan melakukan *request* ke *UserController* yang diteruskan ke *PointModel* untuk menampilkan data *history* nilai siswa, kemudia hasil dari *query* tersebut akan dikembalikan ke *client* dan ditampilkan rekap nilai tersebut. Jika tidak ada nilai yang dimasukkan maka sistem akan menampilkan pesan grafik tidak ada data nilai.



Gambar 3.26 Sequence diagram menampilkan *history* nilai.

3.3.5 Menemukan diagram kelas

Sebelum membuat diagram kelas penulis melakukan pendaftaran objek yang akan menjadi kelas dengan cara memperhatikan *flow of event* dan diagram sekuensial. Berikut merupakan kandidat kelas yang telah diamati dari diagram sekuensial.

Tabel 3.17 kandidat kelas pada tiap diagram sekuensial

Flow of event	Kandidat kelas	Jenis
Melakukan Login	HalamanLogin	boundary
	UserController	entity
	UserModel	entity
	HalamanMenu	boundary
Melihat Informasi materi guru	Halaman Informasi materi guru	boundary
	UserController	entity
	YearPeriodsModel	entity
	SyllabusModel	entity
Melihat Perkembangan siswa	UserController	entity
	LessonModel	entity
	StudentClassModel	entity
	PointsModel	entity
	HalamanLaporan	boundary
Menentukan kelas ajar dan matapelajaran guru	UserController	entity
	UserModel	entity
	YearPeriodsModel	entity
	ClassModel	entity
	LessonModel	entity
	Halaman kelas ajar dan matapelajaran	boundary
Mengelola Data Nilai	Halaman input nilai	boundary
	UserController	entity
	LessonModel	entity
	TaskModel	entity
	AssesmentModel	entity
	PointsModel	entity
Mengelola kelas	Halaman Kelas	boundary
	UserController	entity

Flow of event	Kandidat kelas	Jenis
	GradesModel	entity
	ClassModel	entity
Mengelola Materi Belajar	UserController	entity
	LessonModel	entity
	SyllabusModel	entity
	ResourcesModel	entity
	Halaman Materi belajar	boundary
Mengelola pendaftaran ulang siswa	Halaman Registrasi	boundary
	UserController	entity
	YearPeriodsModel	entity
	ClassModel	entity
	UserModel	entity
	StudentClassModel	entity
Mengelola silabus	HalamanSilabus	boundary
	UserController	entity
	GradesModel	entity
	YearPeriodsModel	entity
	CourseModel	entity
	SyllabusModel	entity
Mengelola Tugas Siswa	Halaman Tugas Siswa	boundary
	UserController	entity
	LessonModel	entity
	TaskModel	entity
	SyllabusModel	entity
Mengelola Pengguna	Halaman kelola pengguna	boundary
	UserController	entity
	UserModel	entity
Menggunakan materi belajar	HalamanUduhMateri	boundary
	UserController	entity
	ResourcesModel	entity
Mengumpulkan tugas	UserController	entity
	AssesmentModel	entity
	TaskModel	entity
	HalamanKumpulkanTugas	boundary
Menampilkan Rekap Nilai	Halaman Rekap Nilai	boundary
	UserController	entity
	LessonModel	entity
	StudentClassModel	entity
	PointsModel	entity

3.3.6 *Class diagram*

Pada sub bab ini akan dibahas pembuatan *class diagram* yang akan digunakan dalam pembuatan sistem LMS. Pada pembuatan diagram kelas tersebut akan dipisah menjadi dua bagian, dikarenakan banyaknya kelas yang digunakan. Bagian pertama adalah bagian *Model-Controller* yang digunakan sistem dalam melakukan pengelolaan data-data dari sistem yang digunakan. Bagian kedua adalah bagian *View-Controller* yang digunakan oleh sistem untuk melakukan permintaan dari *client* ke *controller*.

Diagram kelas adalah alat perancangan terbaik untuk tim pengembang perangkat lunak. Diagram kelas membantu tim pengembang mendapatkan pola kelas-kelas dalam sistem, struktur sistem sebelum menuliskan kode program, dan membantu untuk memastikan bahwa sistem adalah perancangan terbaik dari beberapa alternatif rancangan(Sholiq,2010:150) .

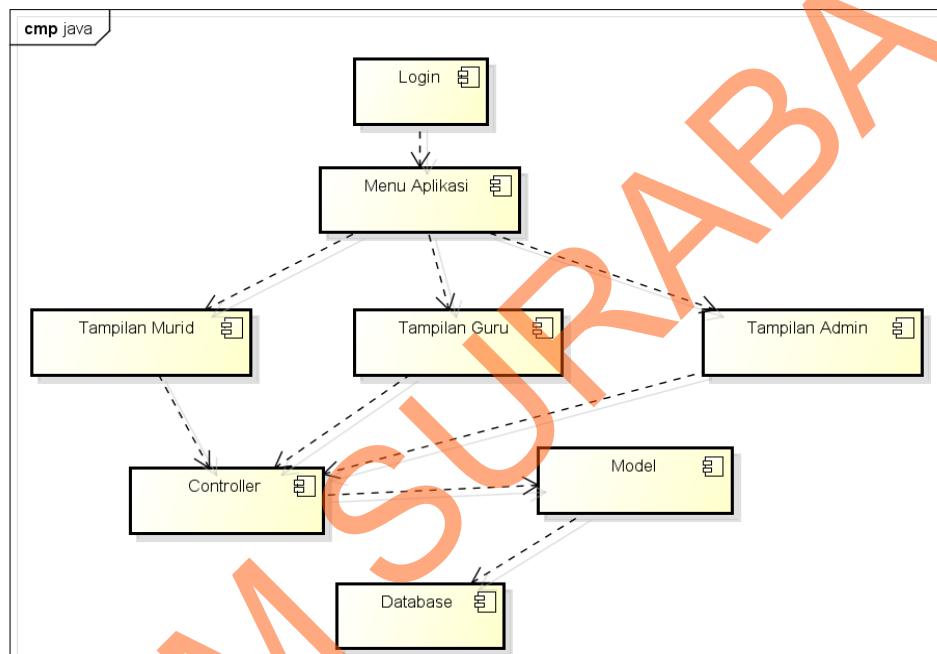
Pada perancangan diagram kelas tersebut ada 3 kelas utama yaitu *model*, *view*, *controller*. Untuk kelas *model* adalah kelas yang menangani segala bentuk operasi yang berhubungan dengan *database*, untuk kelas *view* merupakan kelas yang ada ditampilkan client yang akan melakukan *request* ke *controller*. Selanjutnya di kelas *controller* permintaan tersebut akan diproses. Berikut kelas diagram dapat dilihat di lampiran 1 dan lampiran 2.

3.3.7 *Component diagram*

Pada diagram komponen menunjukkan model secara fisik komponen perangkat lunak. Untuk pembuatan aplikasi LMS terdapat 8 bagian komponen yaitu login,menu aplikasi, tampilan admin, tampilan guru, tampilan

murid, controller, model, dan database. Semua komponen saling berinteraksi dengan memanfaatkan jaringan local area network (LAN) untuk saling berkomunikasi antar komponen.

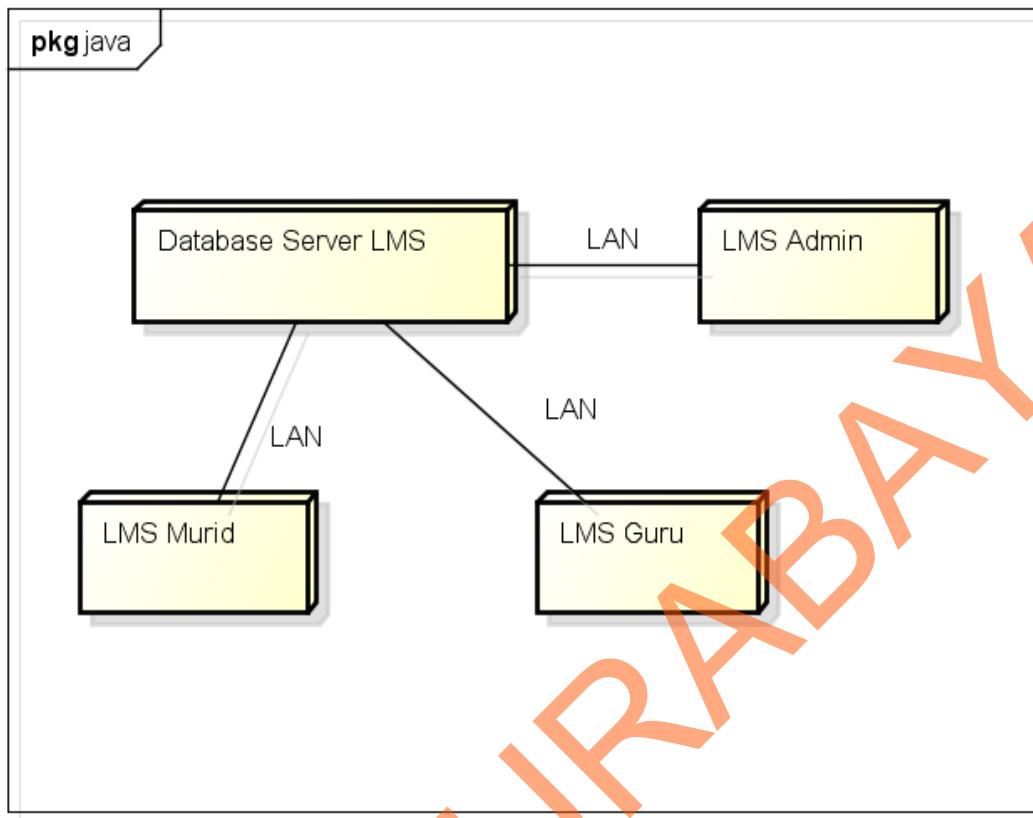
Diagram komponen adalah diagram UML yang menampilkan komponen dalam sistem dan hubungan antara mereka (Sholiq, 2010:212).



Gambar 3.26 Diagram komponen aplikasi *learning management system*.

3.3.8 Deployment diagram

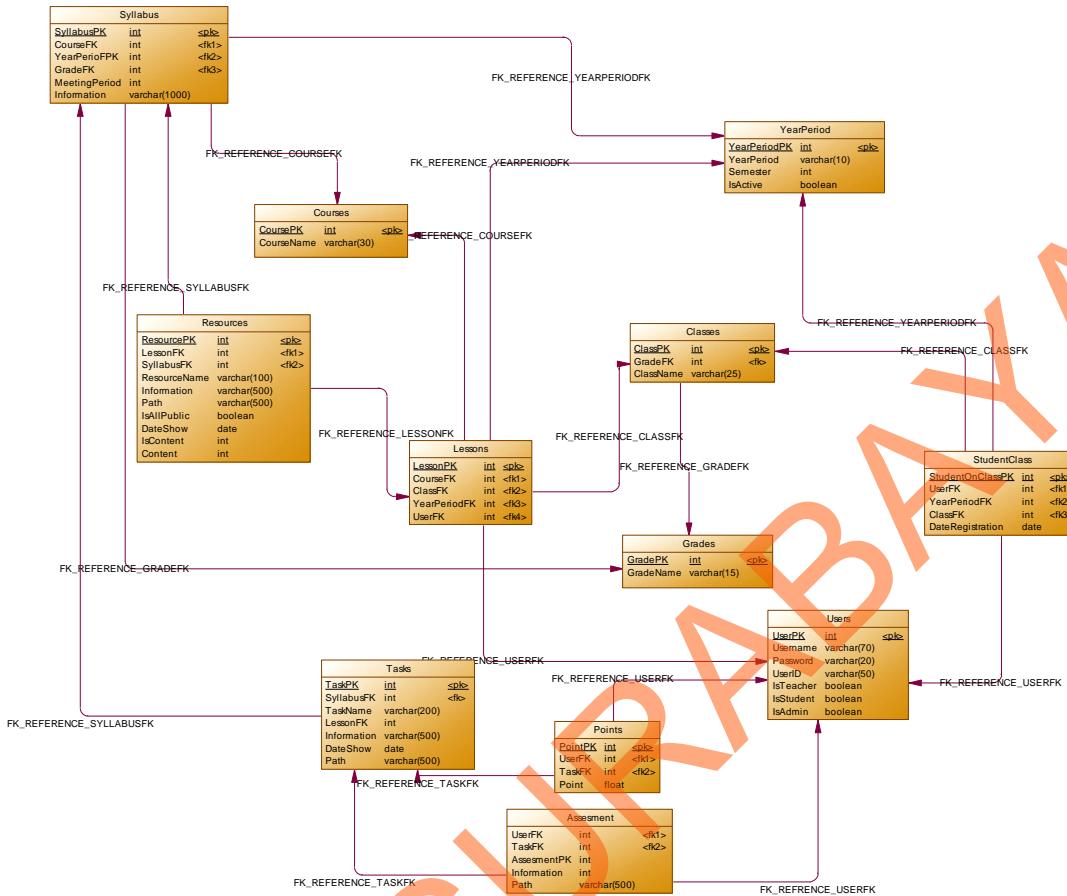
Pada gambar 3.27 merupakan diagram *deployment* yang akan dibangun. Pada gambar tersebut terdapat tiga pengguna yaitu murid, guru dan admin. Semua aplikasi yang digunakan oleh pengguna tersebut mengakses satu *database server* yang digunakan untuk menyimpan data-data dari ketiga pengguna tersebut. Untuk berkomunikasi dengan *database* tersebut sistem yang dibangun memanfaatkan jaringan *local area network* (LAN).



Gambar 3.27 Diagram *deployment* aplikasi *learning management system*.

3.3.9 Physical data model

Pada bagian ini akan digambarkan bentuk dari *physical data model* dari aplikasi *learning management system* yang akan dibangun. *Physical data model* merupakan bentuk fisik dari *database* yang akan dibangun. Dalam rancangan *physical data model* terdapat struktur dari tabel-tabel yang akan digunakan pada pembuatan aplikasi. Berikut ini gambar 3.28 adalah gambar *physical data model* dari aplikasi LMS pada SD alfurqon yang akan bangun.



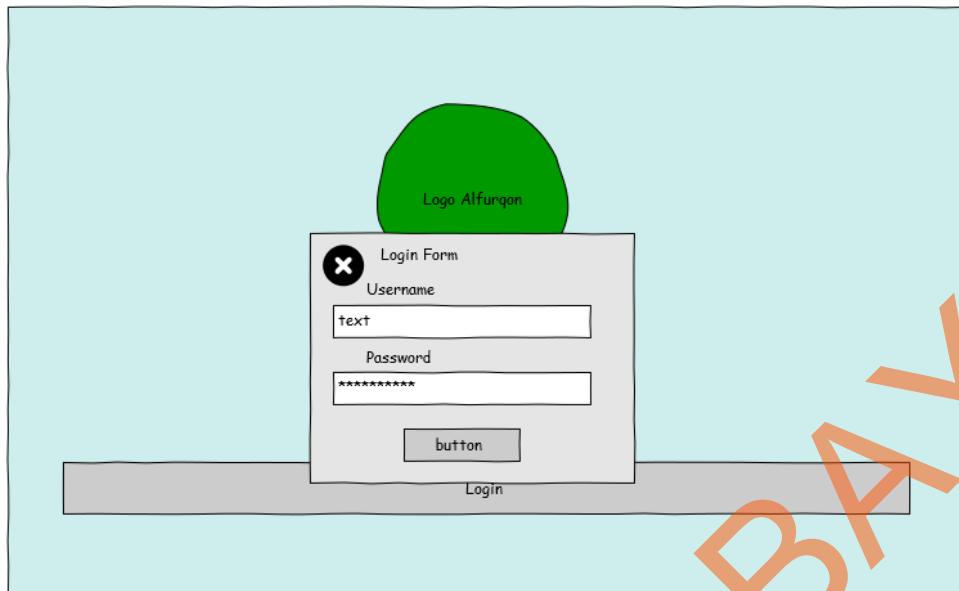
Gambar 3.28 Physical data model aplikasi LMS SD Al-Furqon.

3.3.10 Desain Interface

Pada sub bab ini akan dibahas tentang desain *interface* yang akan dibuat untuk aplikasi LMS pada SD Al-furqon agar pengguna dapat berinteraksi dengan aplikasi tersebut.

A. Desain Interface Login

Desain tampilan halaman *login* berikut ini merupakan halaman *login* untuk semua pengguna yang menggunakan aplikasi LMS. Untuk menggunakannya terdapat dua *input*-an yaitu *username* dan *password* dari pengguna. Seperti terlihat pada gambar berikut ini.

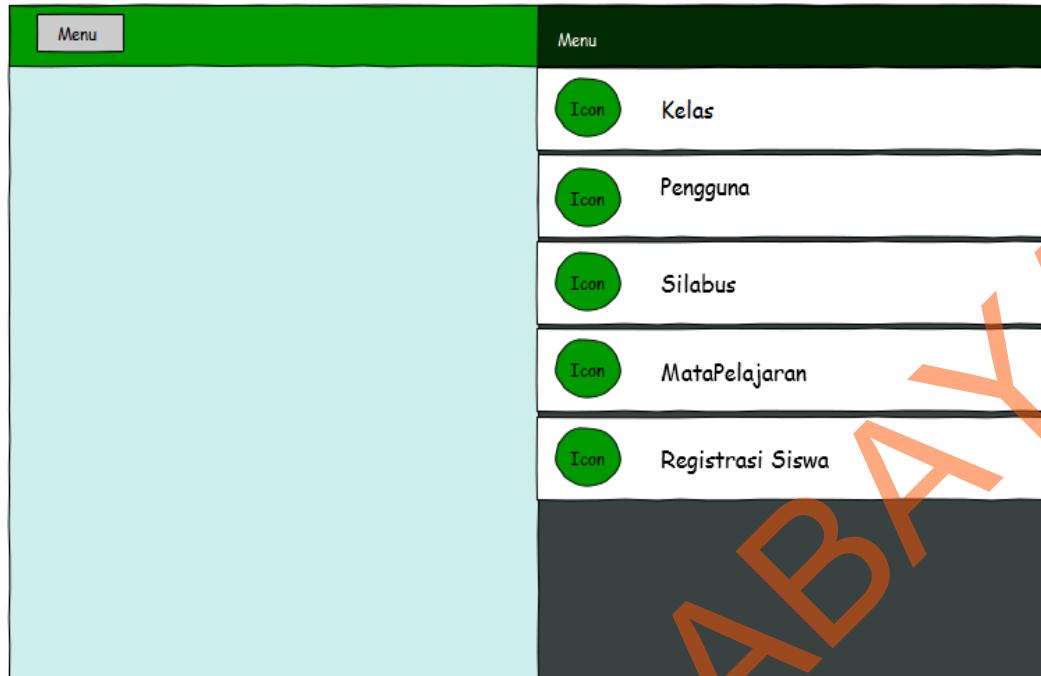


Gambar 3.29 Desain *interface* halaman *login*.

Pada halaman tersebut terdapat *button login*, ketika pengguna menekan *button* tersebut maka akan muncul *dialog* yang menampilkan *textbox username* dan *password* untuk *login* ke aplikasi.

B. Desain Interface Menu untuk Admin

Pada tampilan menu setelah melakukan *login* terdapat beberapa sub menu untuk *login* sebagai admin diantaranya menu kelola kelas, pengguna, silabus, menentukan guru mengajar dan registrasi siswa, untuk menampilkan sub menu tersebut pengguna dapat menekan tombol menu maka *panel* sub menu tersebut akan keluar.

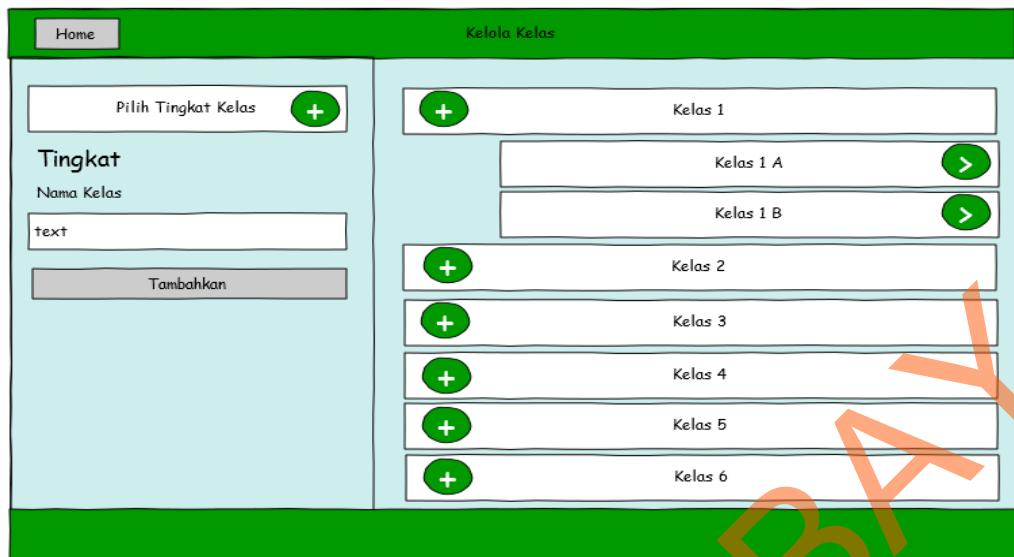


Gambar 3.30 Desain *interface* halaman menu admin.

Untuk keluar dari sistem LMS pengguna dapat menekan tombol *logout* yang ada disebelah kanan.

C. Desain Interface sub menu untuk kelola kelas

Untuk melakukan pengelolaan kelas admin dapat menggunakan menu kelola kelas. Pada menu tersebut terdapat *form* untuk melakukan penambahan dan perubahan data kelas dan menampilkan daftar kelas yang digunakan untuk melakukan pengelolaan data-data kelas. Seperti terlihat pada gambar berikut ini.



Gambar 3.31 Desain *interface* halaman kelola kelas.

D. Desain Interface sub menu untuk kelola pengguna

Desain interface untuk kelola pengguna digunakan oleh admin untuk melakukan penambahan ,perubahan data pengguna. Pada *form* tersebut terdapat *form* untuk melakukan penambahan dan perubahan serta daftar pengguna yang ditampilkan pada bagian sebelah kanan *layout*.



Gambar 3.32 Desain *interface* halaman kelola pengguna.

Pada menu ini juga terdapat pencarian data , hal ini dimunculkan karena banyaknya data pengguna yang ada ketika sistem tersebut digunakan dalam beberapa periode.

E. Desain Interface sub menu untuk kelola silabus

Desain *interface* ini digunakan oleh admin untuk melakukan pengelolaan data silabus atau rencana belajar tiap awal semester.



Gambar 3.33 Desain *interface* halaman kelola silabus.

Pada menu silabus tersebut admin dapat menambahkan data silabus melalui *form* yang ada pada sisi bagian kiri dan admin juga dapat melihat daftar silabus tiap matapelajaran yang berada di sebelah kanan.

F. Desain Interface sub menu untuk kelola guru mengajar

Desain *interface* dari kelola guru mengajar digunakan oleh admin untuk menentukan guru-guru yang mengajar matapelajaran dan kelasnya. Pada bagian ini admin dapat menambahkan dan mengubah guru mengajar yang ada di kelas.

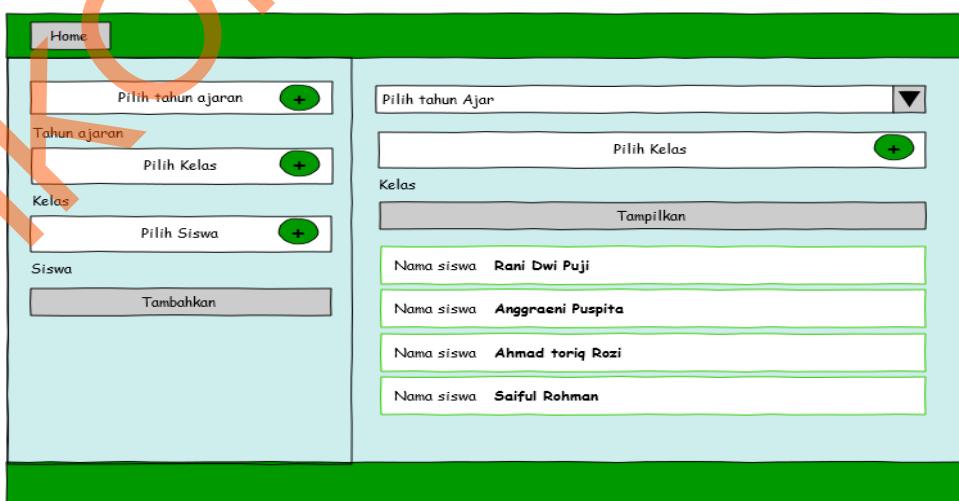
Berikut gambar dari desain *interface* yang akan digunakan.



Gambar 3.34 Desain *interface* halaman kelola guru mengajar.

G. Desain Interface sub menu untuk kelola pendaftaran siswa

Desain tampilan dari menu pendaftaran siswa digunakan oleh bagian admin pada saat mendaftarkan siswa baru di awal tahun ajaran baru. Pada menu ini admin dapat melakukan penambahan data perubahan data di bagian sebelah kiri *layout*. Untuk menampilkan daftar siswa yang telah registrasi terdapat di sebalah kanan *layout*.



Gambar 3.35 Desain *interface* halaman kelola pendaftaran siswa.

H. Desain Interface kelola materi belajar

Pada menu guru dapat menambahkan materi belajar yang akan digunakan pada saat kegiatan belajar mengajar.



Gambar 3.36 Desain *interface* halaman kelola materi belajar.

Ketika guru membuka menu tersebut sistem akan menampilkan daftar silabus yang berisikan materi yang telah diunggah oleh guru. Untuk menambahkan materi baru guru dapat menambahkan materi tersebut dengan mengklik tambahkan lalu akan muncul *form* untuk mengisi data materi baru yang akan diunggah.

Upload materi
Nama materi
<input type="text"/>
Keterangan
<input type="text"/>
Untuk Tanggal
<input type="text"/>
<input type="button" value="Pilih file"/>
<input checked="" type="checkbox"/> Dibagikan untuk kelas lain
<input type="button" value="Simpan"/>

Gambar 3.37 Desain *interface* unggah materi baru.

I. Desain Interface menu kelola tugas

Pada menu mengelola tuga guru dapat memberikan tugas yang akan diberikan kepada murid. Pada bagian kiri layout merupakan daftar pertemuan yang direncanakan.



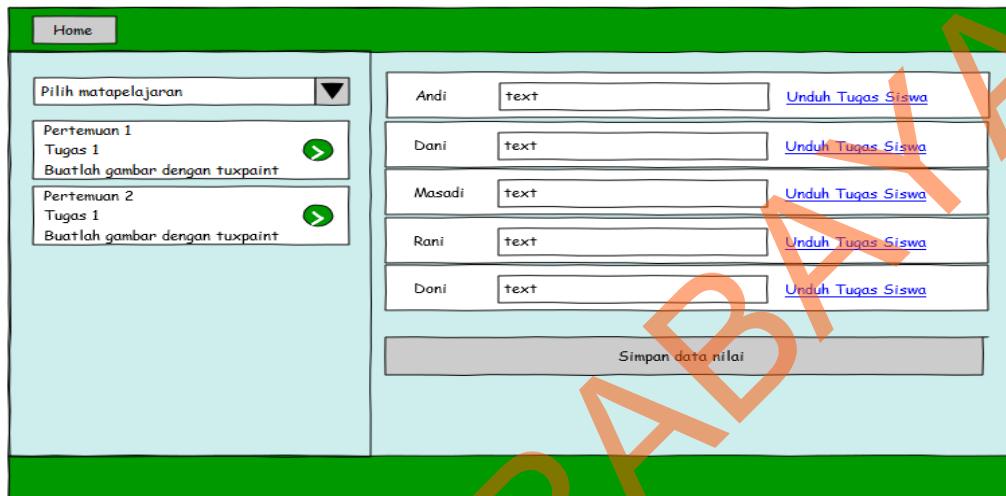
Gambar 3.38 Desain *interface* kelola tugas.

Untuk menambahkan tugas baru guru dapat menambahkan tugas pada daftar pertemuan yang terpilih dengan cara mengklik salah satu daftar pertemuan kemudian akan muncul *dialog* untuk menambahkan tugas tersebut.

Gambar 3.39 Desain *interface dialog* buat tugas.

J. Desain Interface menu Input nilai

Menu input nilai digunakan oleh guru untuk menambahkan dan mengubah data nilai sesuai dengan tugas yang diberikan oleh guru tersebut.

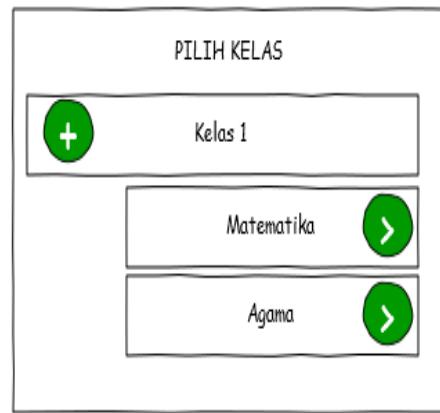


Gambar 3.40 Desain *interface* kelola nilai.

Untuk menampilkan daftar pertemuan pengguna memilih matapelajaran apa yang akan digunakan kemudian sistem akan menampilkan daftar pertemuan yang ada pada mata pelajaran tersebut. Selanjutnya pengguna memilih pertemuan kebberapa dan disebelah bagian kanan akan muncul daftar siswa kelas tersebut. Selanjutnya pengguna dapat memberikan dan menyimpan data nilai tersebut.

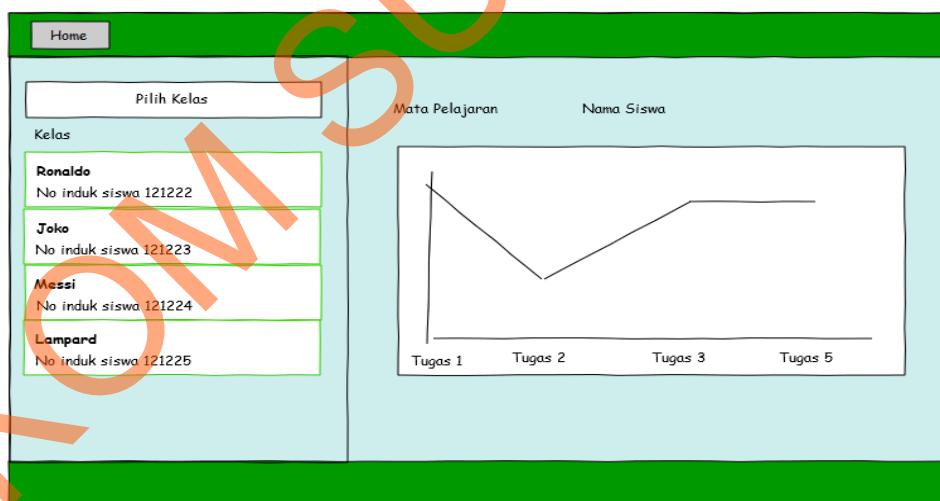
K. Desain Interface menu lihat perkembangan siswa

Untuk data perkembangan nilai guru dapat memanfaatkan menu nilai perkembangan nilai. Pada menu ini guru dapat memilih data kelas dan mata pelajaran yang diajar maka sistem akan menampilkan *dialog* mata pelajaran yang diajar.



Gambar 3.41 Desain *interface dialog* pilih matapelajaran.

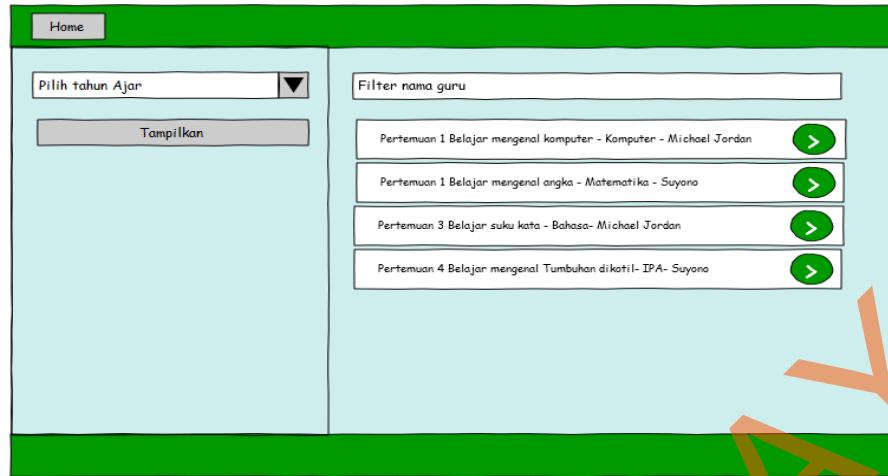
Setelah memilih selanjutnya sistem akan menampilkan daftar siswa yang ada pada kelas tersebut. Kemudian guru dapat memilih siswa pada daftar siswa untuk melihat perkembangan nilai siswa tersebut. Selanjutnya grafik akan muncul dengan keterangan nilai dan tugas.



Gambar 3.42 Desain *interface* melihat perkembangan nilai.

L. Desain Interface menu lihat materi belajar guru

Berikut ini adalah gambar desain *interface* menu melihat materi belajar guru.

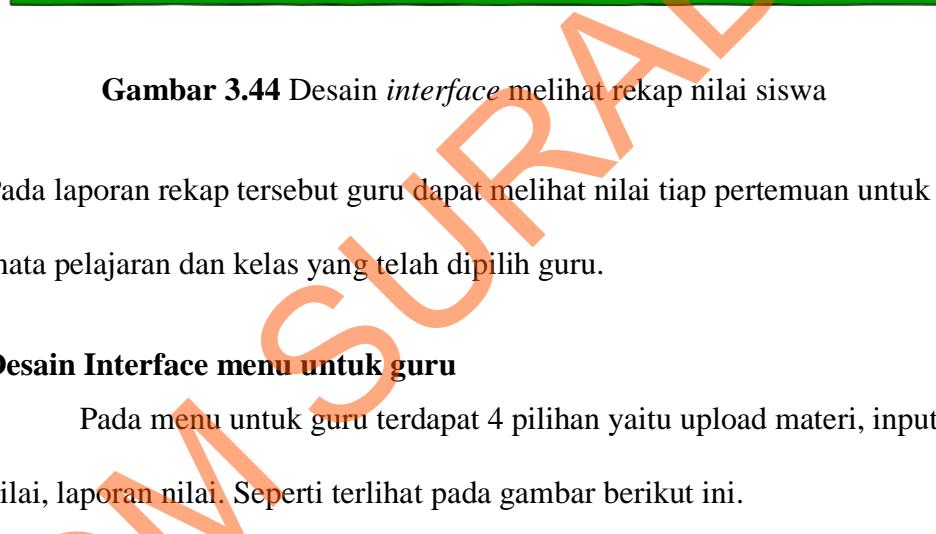


Gambar 3.43 Desain *interface* melihat materi belajar guru.

Pada menu melihat materi belajar guru digunakan oleh bagian admin untuk melihat informasi materi belajar guru yang belum mendistribusikan materinya. Untuk melihat materi tersebut admin dapat memilih tahun ajaran yang akan ditampilkan. Pengguna juga dapat melakukan filter dengan mengetikkan nama guru pada *textbox filter*.

M. Desain Interface melihat rekap nilai

Pada menu ini guru dapat melihat rekap nilai siswa tiap kelas dan mata pelajaran terpilih. Untuk menampilkan rekap nilai tersebut guru dapat memilih terlebih dahulu kelas dan matapelajaran kemudian sistem akan menampilkan rekap nilai pada kelas dan mata pelajaran terpilih tersebut.



The screenshot shows a web-based application interface. At the top left is a 'Home' button. In the center, there's a box with a green plus sign icon and the text 'Pilih Kelas dan matapelajaran'. Below this is a 'Kelas' section. To the right, there are two tabs: 'Mata Pelajaran' and 'Kelas'. Under 'Kelas', a table displays student names, their scores for three meetings (P1, P2, P3), and their average score. The table has columns for 'nama', 'P1', 'P2', 'P3', and 'Rata-rata'. The data is as follows:

nama	P1	P2	P3	Rata-rata
zainul	90	80	80	87
masadi	90	90	90	90
arinda	80	80	80	80
Dani	70	70	70	70

Gambar 3.44 Desain *interface* melihat rekap nilai siswa

Pada laporan rekap tersebut guru dapat melihat nilai tiap pertemuan untuk mata pelajaran dan kelas yang telah dipilih guru.

N. Desain Interface menu untuk guru

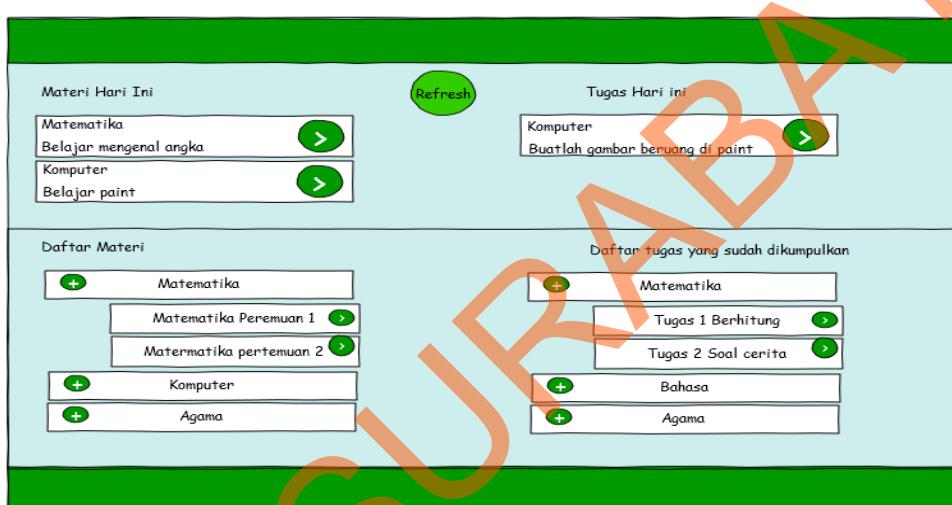
Pada menu untuk guru terdapat 4 pilihan yaitu upload materi, input nilai, laporan nilai. Seperti terlihat pada gambar berikut ini.



Gambar 3.45 Desain *interface* melihat perkembangan nilai.

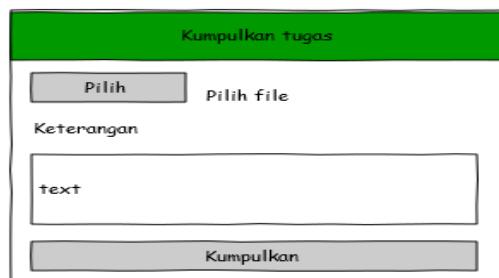
O. Desain Interface menu murid

Pada menu ini digunakan untuk siswa untuk menggunakan segala materi kegiatan belajar mulai dari materi belajar hingga tugas . Pada menu ini terdapat 4 bagian yaitu materi hari ini, tugas hari ini , daftar keseluruhan materi, dan daftar tugas yang telah dikumpulkan oleh siswa. Seperti terlihat pada gambar 3.46 berikut ini.



Gambar 3.46 Desain *interface* menu siswa.

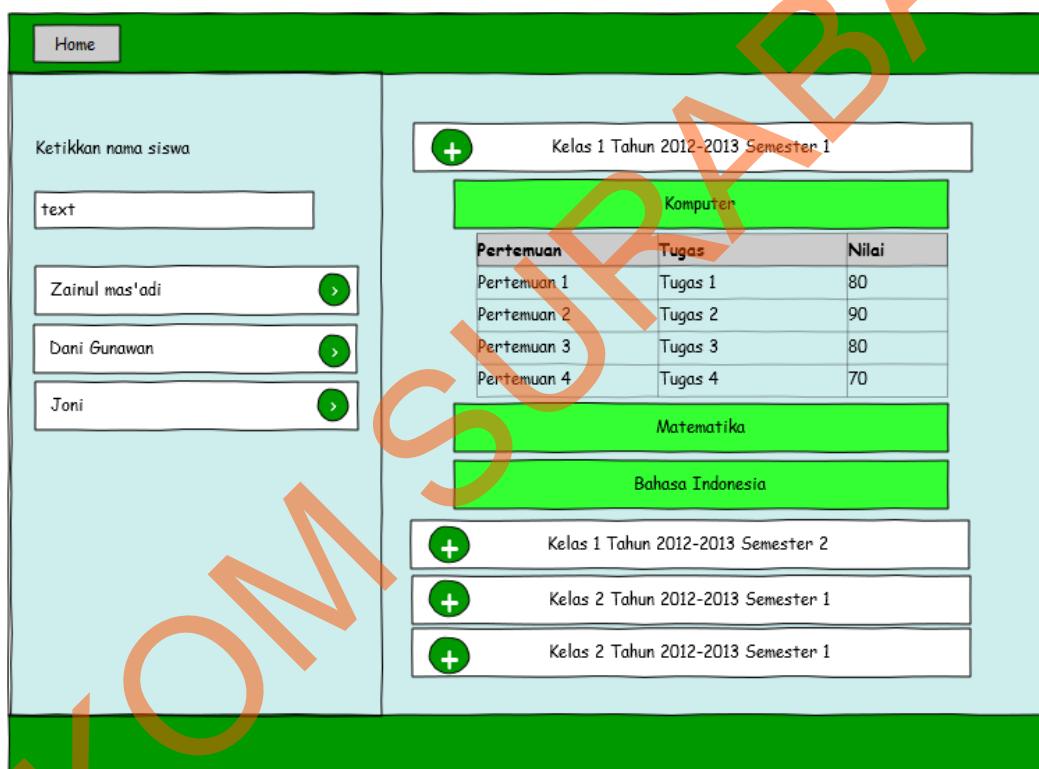
Untuk mengumpulkan tugas siswa dapat memilih di tugas hari ini kemudian mengklik maka akan muncul *dialog* untuk mengumpulkan tugas. Seperti terlihat pada gambar berikut ini.



Gambar 3.47 Desain *interface* mengumpulkan tugas siswa.

P. Desain Interface history nilai siswa

Pada menu ini digunakan untuk guru ketika ingin melihat data *history* siswa. Untuk menampilkan data tersebut pengguna perlu memasukkan nama pengguna untuk mencari data siswa tersebut. Setelah langkah tersebut sistem akan menampilkan data siswa yang siap dipilih untuk ditampilkan *history* nilai siswa yang telah dipilih.



Gambar 3.48 Desain *interface history* nilai siswa.

3.3.11 Rancangan Uji Coba Sistem

Rancangan uji coba ini digunakan untuk mengontrol dan memberikan gambaran *functional* dari sistem ini. Pengujian sistem ini dengan melakukan uji coba pada tiap modul yang ada berdasarkan hak akses dari pengguna yaitu guru, murid, dan admin.

A. Desain Uji coba untuk skenario sebagai admin

Desain uji coba untuk admin dilakukan agar fungsi-fungsi dari user admin dapat berjalan sesuai dengan yang diharapkan. Beberapa skenario antara lain melakukan *login*, mengelola kelas, mengelola pengguna, mengelola silabus, mengelola guru mengajar, mengelola pendaftaran siswa, melihat informasi materi guru. Berikut ini adalah secara detail desain uji coba untuk admin.

A.1 Melakukan Login

Uji coba ini digunakan admin ketika melakukan *login* dan hasil yang diharapkan adalah masuk kedalam halaman utama dari aplikasi, seperti terlihat pada tabel 3.18 berikut ini.

Tabel 3.18 desain uji coba proses admin login

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Berhasil masuk ke aplikasi	<i>Username</i> dan <i>password</i>	Berhasil masuk ke menu utama	

A.2 Mengelola data kelas

Uji coba ini digunakan untuk melakukan pengelolaan data kelas yang dilakukan oleh admin. Pengelolaan tersebut meliputi menambahkan , mengubah dan menghapus data kelas.

Tabel 3.19 desain uji coba proses admin mengelola data kelas

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Menampilkan data kelas	Memilih menu kelas	Sistem menampilkan daftar kelas	
2	Menambahkan data kelas	Memasukkan data kelas(nama kelas dan data tingkat kelas)	Berhasil manambahkan data kelas dan data tampil di daftar kelas	
3	Mengubah data kelas	Memasukkan data kelas(nama kelas dan data tingkat kelas)	Berhasil mengubah data kelas dan data tampil di daftar kelas	
4	Menghapus data kelas	Memilih data kelas	Berhasil menghapus data kelas	

A.3 Mengelola data pengguna

Uji coba ini digunakan untuk melakukan pengelolaan data pengguna yang dilakukan oleh admin. Pengelolaan tersebut meliputi menambahkan , mengubah dan menghapus data pengguna.

Tabel 3.20 desain uji coba proses admin mengelola data pengguna

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Menampilkan data pengguna	Memilih menu pengguna	Sistem menampilkan daftar pengguna	
2	Menambahkan data pengguna	Memasukkan data pengguna(nama ,password, no induk dan status pengguna)	Berhasil manambahkan data pengguna dan data tampil di daftar pengguna	

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
3	Mengubah data pengguna	Memasukkan data pengguna(nama ,password, no induk dan status pengguna)	Berhasil mengubah data pengguna dan data tampil di daftar pengguna	
4	Menghapus data pengguna	Memilih data pengguna didaftar pengguna kemudian memilih hapus	Berhasil menghapus data pengguna	

A.4 Mengelola data silabus

Uji coba ini digunakan untuk melakukan pengelolaan data silabus yang dilakukan oleh admin. Pengelolaan tersebut meliputi menambahkan , mengubah dan menghapus data silabus.

Tabel 3.21 desain uji coba proses admin mengelola data silabus

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Menampilkan daftar silabus	Memilih tahun ajaran dan pilih mata pelajaran	Berhasil menampilkan daftar silabus tiap tiap tingkat kelas	
2	Menambahkan data silabus	Memasukkan data silabus(Tingkat kelas,pertemuan,matapelajaran,tahun ajaran,keterangan)	Berhasil menambahkan data silabus	
3	Mengubah data silabus	Memasukkan data silabus(Tingkat kelas,pertemuan,matapelajaran,tahun ajaran,keterangan)	Berhasil mengubah data silabus	

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
4	Menghapus data silabus	Memilih silabus pada daftar silabus kemudian memilih hapus	Berhasil menghapus data silabus	

A.5 Mengelola data guru mengajar

Uji coba ini digunakan untuk melakukan pengelolaan data guru mengajar yang dilakukan oleh admin. Pengelolaan tersebut meliputi menambahkan , mengubah dan menghapus data guru mengajar.

Tabel 3.22 desain uji coba proses admin mengelola data guru mengajar

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Menampilkan daftar guru mengajar	Memilih tahun ajaran dan kelas	Berhasil menampilkan data guru mengajar	
2	Menambahkan guru mengajar	Memasukkan data pengajaran(Tahun ajar,kelas, matapelajaran,guru)	Berhasil menambahkan data guru mengajar	
3	Mengubah data guru mengajar	Memasukkan data pengajaran(Tahun ajar,kelas, matapelajaran,guru)	Berhasil mengubah data guru mengajar	
4	Menghapus guru mengajar	Memilih guru mengajar pada daftar guru mengajar kemudian memilih hapus	Berhasil menghapus data guru mengajar	

A.6 Mengelola data registrasi siswa

Uji coba ini digunakan untuk melakukan pengelolaan data registrasi siswa yang dilakukan oleh admin. Pengelolaan tersebut meliputi menambahkan , mengubah data registrasi siswa.

Tabel 3.23 desain uji coba proses admin mengelola data registrasi siswa

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	menampilkan daftar registrasi siswa	Memilih tahun ajaran dan kelas	Menampilkan daftar registrasi siswa	
2	Mendaftarkan siswa	Menambahkan data registrasi siswa(tahun ajaran,kelas,siswa)	Berhasil menambahkan data registrasi siswa	
3	Mengubah data registrasi siswa	Menambahkan data registrasi siswa(tahun ajaran,kelas,siswa)	Berhasil mengubah data registrasi siswa	

A.7 Melihat materi yang belum diunggah guru

Uji coba ini digunakan untuk melihat materi yang belum diunggah oleh guru yang dilakukan oleh admin. Sistem akan menampilkan informasi guru yang belum mengunggah materi

Tabel 3.24 desain uji coba admin melihat materi yang belum diunggah

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
21	Melihat materi belajar yang belum didistribusikan guru	Memilih periode belajar	menampilkan daftar guru yang belum mendistribusikan materi belajar	

B. Desain Uji coba pada fungsionalitas guru

Desain uji coba untuk guru dilakukan agar fungsi-fungsi dari user guru dapat berjalan sesuai dengan yang diharapkan. Beberapa fungsionalitas antara lain melakukan login, mendistribusikan materi, membuat tugas, menyimpan nilai, melihat perkembangan siswa. Berikut ini adalah secara detail desain uji coba untuk guru.

B.1 Melakukan login sebagai guru

Uji coba ini digunakan guru ketika melakukan *login* dan hasil yang diharapkan adalah masuk kedalam halaman utama dari aplikasi, seperti terlihat pada tabel 3.25 berikut ini.

Tabel 3.25 desain uji coba proses guru login

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Berhasil masuk ke aplikasi	<i>Username</i> dan <i>password</i>	Berhasil masuk ke menu utama sebagai guru	

B.2 Mengelolah materi belajar

Pada desain uji coba ini guru akan dapat menambahkan , mengubah dan menghapus materi yang didistribusikan kepada muridnya. Beberapa tahapan uji coba seperti terlihat pada tabel 3.26 di bawah ini.

Tabel 3.26 desain uji coba proses guru mendistribusikan materi

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Melakukan login	<i>Username</i> dan <i>password</i>	Dapat masuk ke halaman utama	

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
2	Menampilkan matapelajaran yang diajar beserta materi yang sudah didistribusikan	memilih menu upload materi	Muncul daftar matapelajaran yang diajar beserta materi	
3	Menambahkan materi belajar siswa	Memilih file dan mengisikan keterangan materi, nama materi	File berhasil terupload ke sistem	
4	Mengubah keterangan materi	Keterangan materi belajar, Nama Materi belajar	Materi belajar terubah kemudian data perubahan ada	
5	Menghapus materi belajar	Memilih materi belajar	Materi belajar terhapus dari sistem	

B.3 Mengelolah tugas

Pada desain uji coba ini guru akan dapat menambahkan , mengubah dan menghapus tugas yang didistribusikan kepada muridnya. Beberapa tahapan uji coba seperti terlihat pada tabel di bawah ini.

Tabel 3.27 desain uji coba proses guru mendistribusikan tugas

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Menambahkan tugas untuk siswa	Memasukkan data tugas(nama tugas dan ketarangan)	Tugas baru berhasil ditambahkan	
2	Mengubah data tugas yang diberikan kepada siswa	Memasukkan data tugas(nama tugas dan ketarangan)	Data tugas berhasil diubah	
3	Menghapus data tugas	Memilih tugas yang akan dihapus	Data tugas berhasil dihapus	
4	Menampilkan tugas yang telah dibuat	Memilih menu tugas	Muncul daftar tugas yang telah dibuat	

B.4 Mengelola nilai tugas siswa

Pada desain uji coba ini guru akan dapat menambahkan , mengubah nilai tugas siswa . Beberapa tahapan uji coba seperti terlihat pada tabel di bawah ini.

Tabel 3.28 desain uji coba proses guru mengelolah nilai tugas

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Menampilkan tugas yang telah dibuat	Memilih menu tugas	Muncul daftar tugas yang telah dibuat	
2	Menampilkan nilai tugas siswa	Memilih matapelajaran dan kelas pada menu nilai	Muncul daftar nilai untuk matapelajaran tersebut	
3	Meningputkan nilai tugas siswa	Nilai tugas siswa	Data nilai tersimpan dan tampil didalam tabel nilai	
4	Mengubah nilai tugas siswa	Nilai tugas siswa	Data nilai berhasil dirubah dan tampil didalam tabel nilai	

B.5 Melihat Perkembangan nilai

Pada desain uji coba ini guru akan dapat melihat perkembangan nilai dari siswa yang diajar untuk mata pelajaran terpilih. Beberapa tahapan uji coba seperti terlihat pada tabel 3.29 di bawah ini.

Tabel 3.29 desain uji coba proses guru melihat perkembangan nilai

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	Melihat nilai perkembangan siswa	Memilih kelas, matapelajaran dan siswa pada kelas tersebut	Muncul grafik perkembangan nilai siswa	
2	Melihat rekap nilai	Memilih kelas dan mata pelajaran	Muncul rekap nilai siswa	

3	Melihat <i>history</i> nilai	Menginputkan nama dan memilih siswa	Muncul <i>history</i> dari nilai siswa	
---	------------------------------	-------------------------------------	--	--

C. Desain Uji coba pada fungsionalitas murid

Desain uji coba untuk murid dilakukan agar fungsi-fungsi dari user murid dapat berjalan sesuai dengan yang diharapkan. Beberapa fungsionalitas antara lain melakukan login, mengunduh materi guru, mengumpulkan tugas. Berikut ini adalah secara detail desain uji coba untuk murid.

Tabel 3.30 desain uji coba proses murid.

No	Tujuan yang ingin dicapai	Input	Output yang diharapkan	Hasil
1	melakukan login	Username dan password	Berhasil masuk ke halaman utama	
2	Mengunduh materi	Memilih materi yang akan di unduh	Berhasil mengunduh materi belajar	
3	Mengunduh tugas yang telah dikerjakan	Memilih tugas hari ini	Berhasil mengunduh tugas hari ini	
4	mengumpulkan tugas	Memasukkan data tugas siswa(File terpilih, Keterangan)	Berhasil memasukkan data tugas siswa	