**BAB III**

**ANALISIS DAN PERANCANGAN**

**3.1 Analisis**

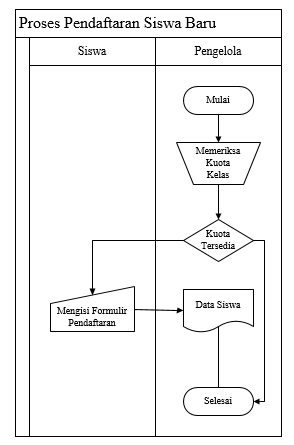
Sebelum merancang sebuah aplikasi yang akan dibangun, factor yang harus diperhatikan adalah tahap analisis. Tahap analisis merupakan tahap pemahaman dengan perangkat lunak yang akan dibangun, kebutuhan fungsional, diagram alir data, kamus alir data, dan enkripsi proses. Tahap analisis bertujuan untuk mengetahui mekanisme system, proses-proses yang terlibat dalam system, serta hubungan antara proses tersebut. Analisis adalah tahap yang sangat penting karena suatu kesalahan dalam tahap ini akan mempengaruhi pada tahap berikutnya. Penelitian membuktikan bahwa kesalahan yang diperbaiki setelah tahap analisis akan memakan biaya yang lebih besar daripada jika diperbaiki saat dilakukan analisis.

Belajar dari Para ahli yang telah mendefinisikan pengertian analisa maka sebuah analisis data, proses dan hasil dari analisa biasanya dilakukan meliputi kegiatan seperti mengorganisasikan data, mengelompokkan data, mengklasifikasi data, memaparkan data dan menarik kesimpulan dari keseluruhan data tersebut. Dari beberapa pengertian analisis diatas dapat ditarik kesimpulan bahwa analisa merupakan sekumpulan kegiatan, aktivitas dan proses yang saling berkaitan untuk memecahkan masalah atau memecahkan komponen menjadi lebih detail dan digabungkan kembali lalu ditarik kesimpulan.

**3.1.1 Analisis Sistem Yang Sedang Berjalan**

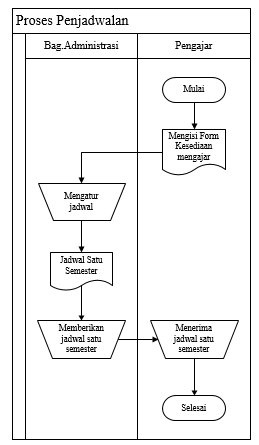
Setelah melakukan wawancara dengan bagian administrasi atau operator bimbingan belajar dapat disimpulkan bahwa, bimbingan belajar ALC masih menggunakan proses manual dalam mengelola bimbingan belajarnya. Masih banyak proses yang kurang tersusun rapi dalam pendataan maupun penyimpanan data dalam

Untuk pendaftaran siswa baru, proses yang digunakan adalah sebagai berikut :

  
Gambar 3.1 Proses Pendaftaran Siswa Baru

Dalam flowmap diatas, ketika calon siswa ingin mendaftar bimbingan belajar pada bimbingan belajar ALC, pengelola harus mengecek kuota kelas yang tersedia. Apabila kuota kelas untuk kelas yang akan didaftarkan masih tersedia, maka pengelola menyerahkan formulir pendaftaran untuk diisi lalu akan memberitahu jadwal bimbingan belajar sesuai kelas siswa tersebut.

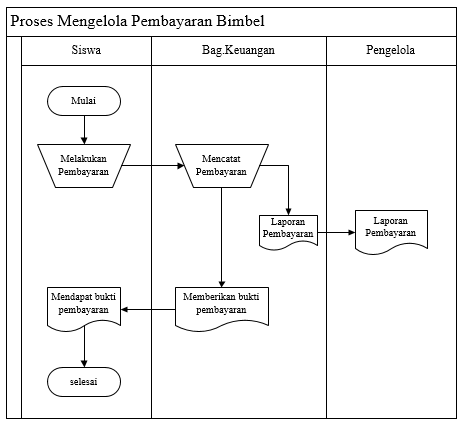
Proses Penjadwalan adalah sebagai berikut :



Gambar 3.2 Proses Penjadwalan

Untuk proses penjadwalan, pengajar mengisi terlebih dahulu form kesedian mengajar yang telah disediakan oleh bagian administrasi bimbingan belajar. Setelah itu bagian administrasi akan mengatur jadwal sesuai dengan form kesedian pengajar dengan kelas yang ada dan jadwal yang tersedia. Setelah itu bagian administrasi akan membuat dokumen jadwal satu semester.

Proses mengelola pembayaran adalah sebagai berikut :



Gambar 3.3 Proses Pembayaran Bimbel

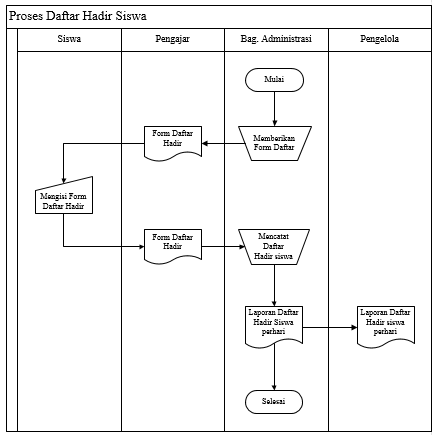
Untuk proses pembayaran biaya bimbingan belajar siswa membayar biaya kepada bagaian keuangan lalu bagian keuangan mencatat pada sebuah dokumen. Setelah itu siswa mendapatkan bukti pembayaran berupa kuitansi. Selanjutnya bagian keuangan memberikan laporan pembayaran kepada pengelola.

Berikut ini adalah daftar jenis pembayaran dan biaya bimbingan belajar ALC:

Tabel 3.1 Daftar Harga Bimbel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Jenis Pembayaran | Kelas | Biaya | |
| Bulanan | Tahunan |
| 1 | Reguler SD | I-VI | Rp. 275.000 | - |
| 2 | Ujian SD | VI | Rp. 300.000 | Rp. 3.100.000 |
| 3 | Reguler SMP | VII-IX | Rp. 325.000 | - |
| 4 | Ujian SMP | IX | Rp. 350.000 | Rp.3.600.000 |
| 5 | Reguler SMA | X-XII | Rp. 375.000 | - |
| 6 | Ujian SMA | XII | Rp. 400.000 | Rp. 4.200.000 |

Proses daftar hadir siswa adalah sebagai berikut :



Gambar 3.4 Proses Daftar hadir siswa

Untuk proses daftar hadir siswa, bagian administrasi memberikan form daftar hadir kepada pengajar. Setelah didalam kelas pengajar akan memberikan daftar hadir pada siswa untuk diisi. Setelah form daftar hadir diisi pengajar memberikan kembali kepada bagian administrasi untuk dicatat kedalam sebuah dokumen. Setelah itu bagian administrasi membuat laporan daftar hadir siswa per hari untuk kemudian diserahkan kepada pengelola.

**3.1.2 Analisis Sistem Yang akan Dibangun**

Berdasarkan proses manual yang sedang berjalan di bimbingan belajar ALC, maka akan dibangun sistem informasi dengan fitur-fitur yang dibutuhkan untuk proses administrasi yang ada pada bimbingan belajar ALC. Adapun fitur yang akan dibangun adalah sebagai berikut.

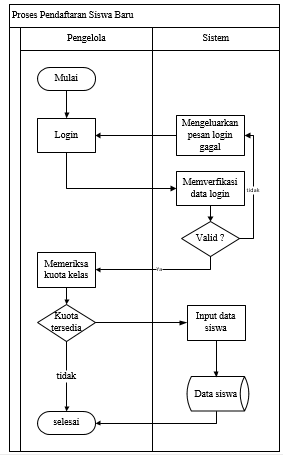
Tabel 3.2 Analisis yang akan dibangun

| Bagian Administrasi | Bagian Keuangan | Pengelola |
| --- | --- | --- |
| 1. Fitur Login 2. Fitur mengelola data siswa 3. Fitur Mengelola jadwal belajar 4. Fitur Mengelola daftar hadir siswa 5. Fitur Mengelola Data Pengajar 6. Fitur Mengelola Data Ruangan | 1. Fitur Login 2. Fitur Mengelola data pembayaran | 1. Fitur Login 2. Fitur *Input* data murid baru 3. Fitur melihat laporan keuangan 4. Fitur melihat laporan jumlah murid 5. Fitur mengirim SMS pemberitahuan 6. Fitur melihat kapasitas ruangan 7. Fitur melihat jadwal belajar 8. Fitur melihat data pengajar 9. Fitur melihat data siswa 10. Fitur mengelola kuota kelas 11. Fitur melihat daftar hadir siswa 12. fitur melihat daftar hadir pengajar |

**3.1.2.1 Flowmap Aplikasi Yang Akan Dibangun**

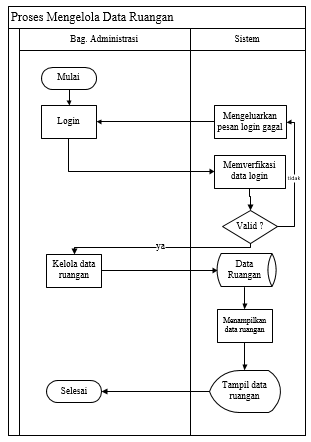
Berdasarkan latar belakang dan analisis sistem lama dalam proses administrasi, maka akan dibuat sebuah aplikasi administrasi bimbingan belajar ALC sesuai dengan fitur yang telah dijabarkan disebelumnya.

Proses pendaftaran siswa baru adalah sebagai berikut :

  
Gambar 3.5 Proses Pendaftaran Siswa Baru

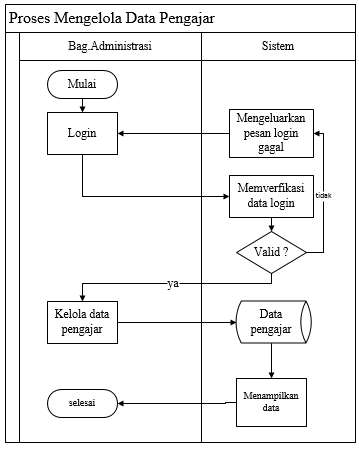
Proses pendaftaran pada bimbingan belajar ALC adalah seorang siswa harus datang langsung ke tempat bimbingan belajar ALC. Pengelola akan mengecek kuota kelas pada aplikasi apakah masih tersedia atau sudah habis. Jika masih ada siswa diberi formulir pendaftaran kemudian pengelola input data siswa dan langsung disimpan dalam database.

Proses mengelola data ruangan adalah sebagai berikut :

  
Gambar 3.6 Proses Mengelola Data Ruangan

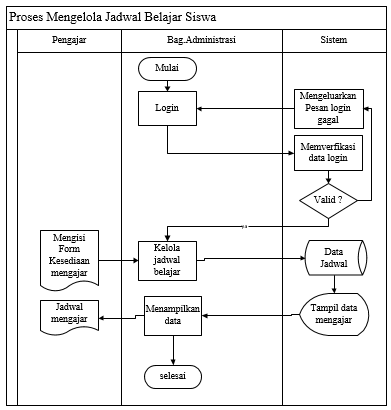
Pada *flowmap* diatas menjelaskan bagaiman bagian administrasi mengelola data ruangan. Bagian administrasi akan *login* menggunakan *password* dan *username* yang sudah diberikan sebelumnya. Setelah *login*, bagian administrasi memilih menu data ruangan, dimana bagian administrasi dapat menginputkan, memperbahuri dan menghapus data. Data akan tersimpan otomatis kedalam *database* dan data yang telah dikelola akan ditampilkan.

Proses mengelola data pengajar adalah sebagai berikut :

  
Gambar 3.7 Proses Mengelola Data Pengajar

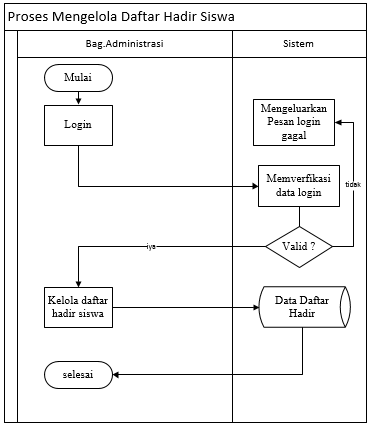
Pada *flowmap* diatas bagian administrasi dapat menginput, mengubah dan menghapus data pengajar. Data pengajar akan tersimpan secara otomatis pada *database*. Data yang telah diolah dapat dilihat oleh Bagian Administrasi.

Proses mengelola jadwal belajar siswa adalah sebagai berikut :

  
Gambar 3.8 Proses Mengelola Jadwal Belajar

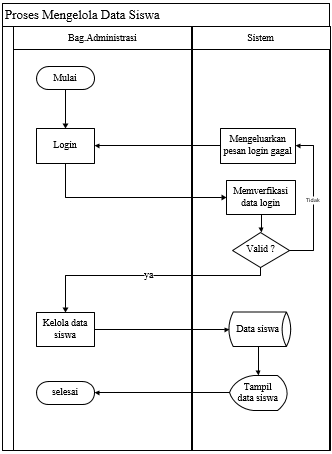
Pada proses penjadwalan, pengajar tetap mengisi form kesedian mengajar. Didalam form tersedia kolom jadwal kesedian mengajar. Setelah itu bagian administrasi memasukkan jadwal mengajar sesuai dengan form yang diisi oleh pengajar. Data yang dimasukkan adalah hari belajar, waktu belajar, ruangan serta kelas yang akan diajar. Bagian administrasi juga dapat mengubah jadwal dan menghapus jadwal apabila ada perubahan. Setelah itu hasil dari dari proses itu tersimpan didalam *database*. Pengajar mendapat jadwal berupa dokumen yang diberikan oleh bagian administrasi.

Proses mengelola daftar hadir siswa adalah sebagai berikut :

  
Gambar 3.9 Proses Mengelola daftar Hadir Siswa

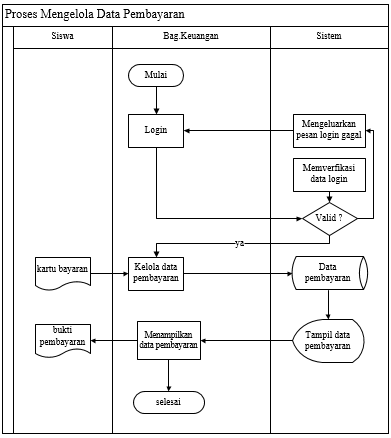
Pada *flowmap* diatas bagian administrasi mengelola data daftar hadir siswa. Setelah menerima dokumen kehadiran dari pengajar, bagian administrasi akan menginputkan data daftar hadir siswa. Data daftar hadir akan tersimpan dalam *database* daftar hadir siswa. Setelah itu bagian administrasi dapat melihat data daftar hadir yang telah diinputkan.

Proses mengelola data siswa adalah sebagai berikut :

  
Gambar 3.10 Proses Mengelola Data Siswa

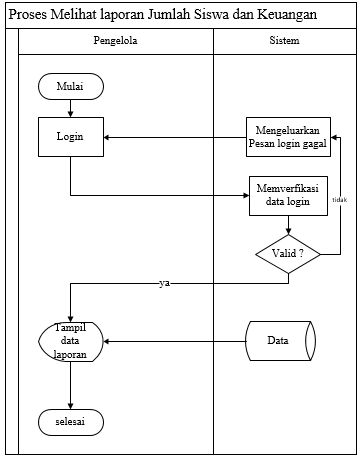
Pada *flowmap* diatas menjelaskan tentang proses bisnis pengelolaan data siswa. Bagian administrasi dapat melihat, mengubah dan menghapus data siswa. Setelah itu data masuk kedalam *database*. Data yang telah dikelola dapat dilihat oleh bagian administrasi.

Proses mengelola data pembayaran adalah sebagai berikut:

  
Gambar 3.11 Proses Mengelola Data Pembayaran

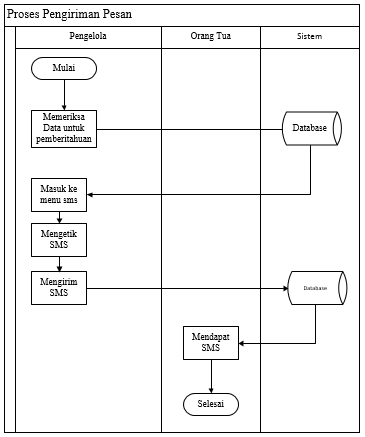
Pada proses pembayaran, bagian keuangan dapat melakukan input data pembayaran siswa. Data akan tersimpan secara otomatis didalam database. Setelah itu data yang telah dikelola dapat dilihat kembali. Dalam proses pembayaran, siswa dapat memilih salah satu paket pembayaran yang disediakan oleh bimbingan belajar ALC. Pilihan pertama siswa dapat melakukan pembayaran setiap bulan yang disebut paket bulanan, pilihan kedua yaitu paket tahunan, siswa dapat mencicil pembayaran sebanyak tiga kali pada tiga bulan pertama setelah mulai belajar. Berikut adalah daftar harga paket bulanan dan tahunan.

Proses melihat laporan adalah sebagai berikut:

  
Gambar 3.12 Proses Melihat laporan Jumlah Siswa dan Keuangan

Pengelola dapat melihat laporan jumlah siswa dan laporan keuangan yang diambil dari *database.*

Proses pengiriman pesan adalah sebagai berikut:



Gambar 3.13 Proses Pengiriman Pesan

Pengelola dapat mengirim pesan laporan ke orang tua dengan data yang ada pada *database*.

* + 1. **Deskripsi Kebutuhan Aplikasi**

Pembuatan aplikasi administrasi bimbingan belajar berbasis framework CodeIgniter dan sms gateway ini dibangun untuk membantu memudahkan administrasi pada bimbingan belajar.

Sistem yang dibuat ini memberikan hak akses yaitu :

|  |  |
| --- | --- |
| **Kategori Pengguna** | **Tugas** |
| **Pengelola**  **Administrasi**  **Bag.Keuangan** | **Mengontrol semua kegiatan di bimbingan belajar.**  **Memasukan data-data tentang kegiatan bimbingan belajar.**  **Mengurus bagian keuangan.** |

Tabel 3.3 Hak Akses Untuk Pengguna Aplikasi

**3.1.3.1 Analisis Kebutuhan Perangkat Lunak Dan Perangkat Keras**

Spesifikasi perangkat lunak yang dibutuhkan adalah sebagai berikut :

|  |  |  |
| --- | --- | --- |
| **Pengelola** | | |
| **Operating Sistem** | **:** | **Microsoft Windows 7** |
| **Development Tools** | **:** | **Sublime Text 3** |
| **Aplikasi** | **:** | **CodeIgniter, SMS Gateway** |

Tabel 3.4 spesifikasi perangkat lunak

* + - 1. **Analisis Kebutuhan Perangkat Keras yang dibutuhkan untuk Membangun Aplikasi ini**

Spesifikasi Perangkat keras yang dibutuhkan adalah sebagai berikut :

|  |  |  |
| --- | --- | --- |
| **Administrator** | | |
| **Procesor** | **:** | **Intel® Core™ i3-2328M** |
| **Memory** | **:** | **4.00 GB** |
| **Hardisk** | **:** | **1.00 GB** |
| **VGA** | **:** | **2024 MB** |

Tabel 3.5 Spesifikasi Perangkat Keras

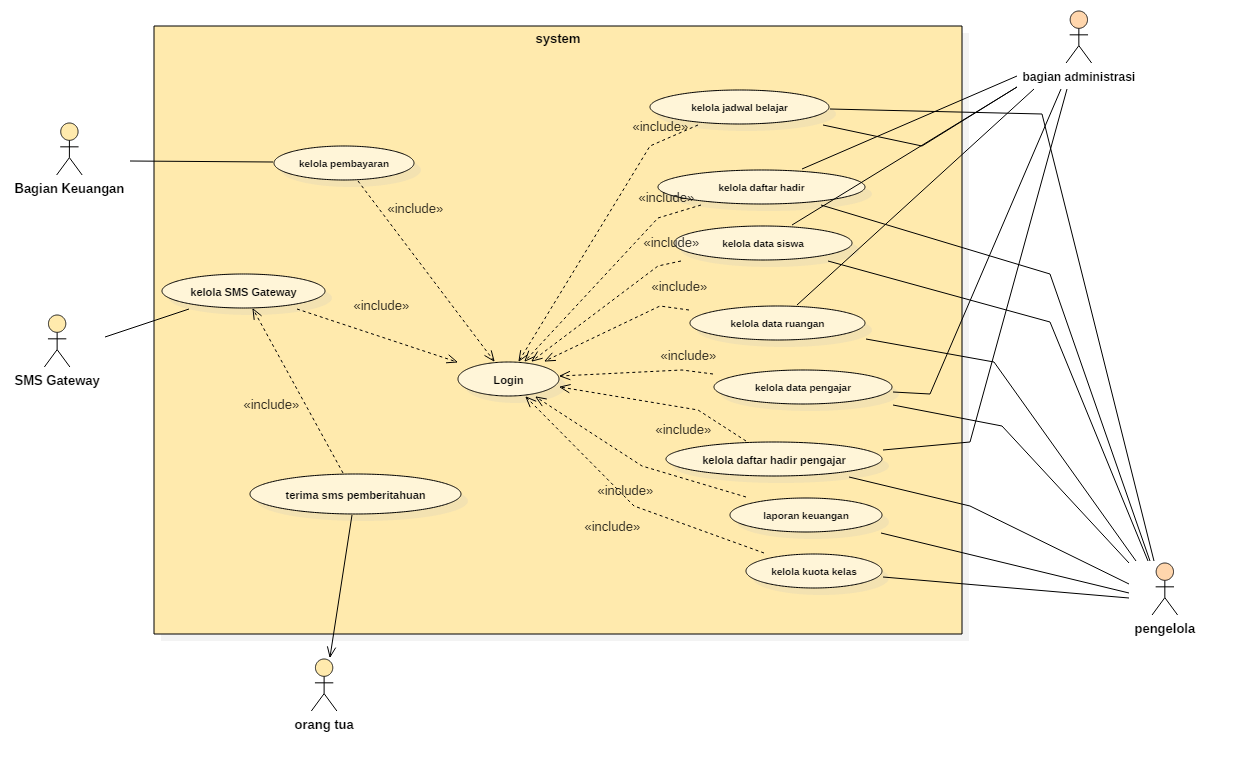
**3.2 Perancangan**

Berikut ini adalah suatu gambar analisis sistem pada perancangan dan Pembuatan aplikasi administrasi bimbingan belajar berbasis framework CodeIgniter dan sms gateway menggunakan UML, yang terdiri dari usecase Diagram, class Diagram, sequence diagram, collaboration diagram, activity diagram, statechart diagram, component diagram, dan deployment diagram.

**3.2.1 Usecase Diagram**

**3.2.1.1 Definisi Use Case Diagram**

Use Case diagram merupakan model diagaram yang akan digunakan untuk menggambarkan requirement fungsional dari sistem yang akan dibuat. Pada use case diagram ini akan menekankan “siapa” melakukan “apa” dalam sistem yang akan dibuat.

****

*Gambar 3.14* Usecase Sistem Yang Akan Dibangun

**Defini Aktor :**

1. Pemilik : Aktor ini merupakan pengelola bimbingan belajar ALC. Aktor ini mempunyai hak untuk menambahkan data siswa baru, melihat data siswa, melihat jadwal belajar, melihat data pengajar, melihat kapasitas ruangan, melihat laporan keuangan dan melihat laporan jumlah murid.
2. Keuangan : Aktor ini dapat mengelola data pembayaran siswa.
3. Administrasi : Aktor ini bertindak juga sebagai admin. Aktor ini mempunyai hak untuk mengelola data siswa, mengelola data pengajar, mengelola data ruangan, mengelola daftar hadir siswa, mengelola daftar hadir pengajar dan mengelola jadwal belajar siswa.
4. SMS Gateway : Aktor ini untuk mengelola pengiriman SMS pemberitahuan dengan SMS Gateway
5. Orang Tua : Aktor pasif untuk menerima SMS pemberitahuan.

### 3.2.1.2 Skenario *Usecase*

#### Usecase Login

Nama *Usecase* : Login

Aktor : Pengelola, Bagian Keuangan, Bagian Administrasi

Tujuan : Untuk mengakses halaman aplikasi

Deskripsi : Proses ini adalah kegiatan yang diawali oleh aktor untuk menjalankan sistem

Pre-condition : Aktor sudah berada di halaman login

Post-condition : Menampilkan halaman awal aplikasi

Tabel 3.6 Skenario *Usecase* *Login*

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar (*basic flow*) | |
| 1.Memasukkan data *login* berupa *username* dan *password* pada *form login.* |  |
| 2.Menekan tombol “login”. |  |
|  | 3.Konfirmasi data di database. |
|  | 4.Jika *username* dan *password valid* maka akan menampilkan menu utama. |
| Alur alternatif I | |
|  | 4a. Jika data *login* yang dimasukkan salah maka akan muncul pesan “*username* dan *password* yang anda masukkan salah*”.* |
| Aksi Aktor | Reaksi Siswa |
|  | 5. Kembali ke *form login.* |

#### Usecase Kelola Data Ruangan

Nama *Usecase* : Kelola Data Ruangan

Aktor : Bagian Administrasi

Tujuan : Untuk memasukkan, mengubah, mengecek dan menghapus data ruangan

Deskripsi : Proses ini adalah kegiatan untuk mengelola data ruangan yang ada pada bimbingan belajar ini

Pre-condition : Bagian Administrasi sudah *login*, bagian administrasi sudah berada di menu ruangan

Post-condition : Data ruangan sudah tersimpan di *database*

Tabel 3.7 Skenario *Usecase* Mengelola Data Ruangan

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar (*basic flow)* | |
| 1. Memilih menu kelola ruangan. |  |
|  | 1. Menampilkan halaman ruangan. |
| 1. Menekan tombol “tambah”. |  |
|  | 1. Menampilkan *form* tambah ruangan. |
| 1. Mengisikan data ruangan. |  |
| 1. Menekan tombol “simpan”. |  |
| Aksi Aktor | Reaksi Sistem |
|  | 1. Memeriksa validitas data yang dimasukkan. |
|  | 1. Apabila *valid* maka data akan disimpan di *database* ruangan. |
|  | 1. Menampilkan daftar ruangan. |
| Alur Alternatif I | |
|  | 8a. Apabila data tidak *valid* maka akan muncul pesan *error.* |
|  | 9a. Tetap dihalaman *form* tambah ruangan. |
| Alur Alternatif II | |
| 6a. Menekan tombol “batal”. |  |
|  | 7a. Menampilkan daftar ruangan. |
| Alur alternatif III | |
| 3a. Menekan tombol “ubah”. |  |
|  | 4a. Menampilkan halaman *edit* ruangan. |
| 5a. memperbaharui data ruangan. Beberapa pilihan data yang dapat diperbaharui kapasitas ruangan dan nama ruangan. |  |
| 6b. Menekan tombol “simpan”. |  |
|  | 7b. memeriksa validitas data yang diperbaharui. |
| Aksi Aktor | Reaksi Sistem |
|  | 8b. apabila *valid* maka data akan disimpan di *database* ruangan. |
|  | 9b. Menampilkan daftar ruangan. |
| Alur alternatif IV | |
|  | 8c. Apabila tidak *valid* maka akan menampilkan pesan *error*. |
|  | 9c. Kembali ke *form edit* ruangan. |
| Alur alternatif V | |
| 6c. Menekan tombol “batal” |  |
|  | 7c. Menampilkan halaman daftar ruangan. |
| Alur alternatif VI | |
| 3a. menekan tombol “hapus”. |  |
|  | 4b. Menampilkan pesan “yakin menghapus data ini?”. |
| 5b. Menekan tombol “*Ok*”. |  |
|  | 6c. data ruangan otomatis terhapus dari *database*. |
| Alur alternatif VII | |
| 5c. Menekan tombol “*Cancel*”. |  |
|  | 5d. Kembali ke daftar ruangan. |

#### Usecase Kelola Data Pengajar

Nama Usecase : Kelola Data Pengajar

Aktor : Bagian Administrasi

Tujuan : Untuk memasukkan, mengubah dan menghapus data pengajar

Deskripsi : Proses ini adalah kegiatan untuk mengelola data pengajar yang ada pada bimbingan belajar ini

Pre-condition : Bagian Administrasi sudah berada di menu pengajar

Post-condition : Data siswa sudah tersimpan di *database*

Tabel 3.8 Skenario *Usecase* Kelola Data Pengajar

|  |  |  |
| --- | --- | --- |
| Aksi Aktor | Reaksi Sistem | |
| Alur dasar (*basic flow)* | | |
| 1. Memilih menu kelola pengajar. |  | |
|  | 1. Menampilkan halaman pengajar. | |
| 1. Menekan tombol “tambah pengajar” . |  | |
|  | 1. Menampilkan *form* tambah pengajar. | |
| 1. Mengisikan data pengajar. |  | |
| 1. Menekan tombol “simpan” |  | |
|  | 1. Memeriksa validasi data yang dimasukkan. | |
|  | 1. Apabila *valid* maka data akan disimpan di *database* ruangan. | |
| Aksi Aktor | Reaksi Sistem | |
|  | 1. Menampilkan daftar pengajar. | |
| Alur alternatif I | | |
|  | 7a. Apabila data tidak *valid* maka akan menampilkan pesan *error.* | |
|  | 8a. Tetap dihalaman *form* tambah pengajar. | |
| Alur alternatif II | | |
| 6a. Menekan tombol “batal”. |  | |
|  | 7b. Menampilkan daftar pengajar. | |
| Alur alternatif III | | |
| 3a. Menekan tombol “ubah”. | |  |
|  | | 4a. Menampilkan halaman *edit* pengajar. |
| 5a. memperbaharui data pengajar. Beberapa pilihan data yang dapat diperbaharui nama pengajar, alamat pengajar, no telepon, waktu kesediaan, status, pendidikan terakhir dan tingkat mengajar. | |  |
| 6b. Menekan tombol “simpan”. | |  |
|  | | 7b. Memeriksa validitas data yang diperbaharui. |
|  | | 8b. Apabila *valid* maka data akan disimpan di *database* pengajar. |
| Aksi Aktor | | Reaksi Sistem |
|  | | 9a. Menampilkan daftar pengajar |
| Alur alternatif IV | | |
|  | | 8c. Apabila tidak *valid* maka akan menampilkan pesan *error.* |
|  | | 9b. Kembali ke *form edit* pengajar. |
| Alur alternatif V | | |
| 3b. menekan tombol “hapus” | |  |
|  | | 4b. Menampilkan pesan “yakin menghapus data ini?” |
| 5b. Menekan tombol “*Ok*” | | 6c. data pengajar otomatis terhapus dari *database*. |
| Alur alternatif VI | | |
| 5c. Menekan tombol “*Cancel*” | |  |
|  | | 5d. Kembali ke daftar pengajar |

#### Usecase Kelola Data Siswa

Nama Usecase : Kelola Data Siswa

Aktor : Bagian Administrasi

Tujuan : Untuk mengubah dan menghapus data siswa

Deskripsi : Proses ini adalah kegiatan untuk mengelola data siswa yang ada pada bimbingan belajar ini

Pre-condition : Bagian Administrasi sudah login, bagian administrasi berada di menu siswa

Post-condition : Data siswa sudah tersimpan di *database*

Tabel 3.9 Skenario *Usecase* Kelola Data Siswa

|  |  |  |
| --- | --- | --- |
| Aksi Aktor | Reaksi Sistem | |
| Alur dasar (*basic flow)* | | |
| 1. Memilih menu siswa. |  | |
|  | 1. Menampilkan halaman siswa. | |
| 1. Memilih kelas berdasarkan data yang terdapat dalam database. Pilihan kelas terdiri dari kelas I-XII. |  | |
| 1. Menekan tombol “cari”. |  | |
|  | 1. Menampilkan data siswa sesuai dengan kelas yang dicari. | |
| 1. Menekan tombol “ubah”. |  | |
|  | 1. Menampilkan halaman *edit* siswa. | |
| 1. Memperbaharui data siswa. Beberapa pilihan data yang dapat diperbaharui nama siswa, tingkat, asal sekolah, nomor telepon siswa, kelas, paket belajar, jangka pembayaran, nama orang tua dan nomor telepon orang tua. |  | |
| 1. Menekan tombol “simpan” |  | |
| Aksi Aktor | Reaksi Sistem | |
|  | 1. Memeriksa validitas data yang diperbaharui. | |
|  | 1. Apabila *valid* maka data akan disimpan di *database* siswa. | |
|  | 1. Menampilkan halaman siswa. | |
| Alur alternatif I | | |
|  | 11a. Apabila data tidak *valid* maka akan menampilkan pesan *error*. | |
|  | 12a. Kembali ke *form* *edit* siswa | |
| Alur alternatif II | | |
| 8a. Menekan tombol “batal”. |  | |
|  | 9a. Menampilkan halaman siswa. | |
| Alur alternatif III | | |
| 6a. Menekan tombol “hapus”. | |  |
|  | | 7a. Menampilkan pesan “yakin menghapus data ini?” |
| 8b. Menekan tombol “*Ok*” | |  |
|  | | 9b. data siswa otomatis terhapus dari *database*. |
| Alur alternatif IV | | |
| 8c. Menekan tombol “*Cancel*” | |  |
|  | | 9c. Kembali kehalaman siswa. |

#### Usecase Kelola Daftar Hadir Siswa

Nama Usecase : Kelola Daftar Hadir Siswa

Aktor : Bagian Administrasi

Tujuan : Untuk memasukkan dan mengubah daftar hadir siswa setiap harinya

Deskripsi : Proses ini adalah kegiatan untuk mengelola daftar hadir siswa yang ada pada bimbingan belajar ini

Pre-condition : Bagian Administrasi sudah login, bagian administrasi sudah berada di menu daftar hadir

Post-condition : Data daftar hadir sudah tersimpan di *database*

Tabel 3.10 Skenario *Usecase* Kelola Daftar Hadir Siswa

|  |  |  |
| --- | --- | --- |
| Aksi Aktor | Reaksi Sistem | |
| Alur dasar (*basic flow)* | | |
| 1. Memilih menu daftar hadir. |  | |
| 1. Memilih sub menu tambah daftar hadir. |  | |
|  | 1. Menampilkan halaman untuk memilih hari sesuai jadwal yang ada. | |
| 1. Memiliih hari. |  | |
| 1. Menekan tombol “cari”. |  | |
|  | 1. Menampilkan jadwal belajar sesuai dengan hari yang dipilih. | |
| 1. Menekan “Kehadiran Siswa”. |  | |
|  | 1. Menampilkan halaman untuk memasukkan kehadiran siswa. | |
| Aksi Aktor | Reaksi Sistem | |
| 1. Mengisikan data . |  | |
| 1. Menekan tombol “simpan”. |  | |
|  | 1. Memeriksa validasi data yang dimasukkan. | |
|  | 1. Apabila *valid* maka data akan disimpan di *database* kehadiran\_siswa. | |
| 1. Menekan tombol “selesai”. |  | |
|  | 1. Menampilkan halaman untuk memasukkan kehadiran siswa. | |
| Alur alternatif I | | |
|  | 12a. Apabila data tidak *valid* maka akan muncul pesan *error.* | |
|  | 13a. Kembali kehalaman untuk memasukkan data kehadiran siswa. | |
| Alur alternatif II | | |
| 1a. Memilih menu daftar hadir. |  | |
| 2a. Memilih sub menu lihat daftar hadir. |  | |
|  | 3a. Menampilkan halaman untuk memilih tanggal dan kelas. | |
| 4a. Menekan tombol “cari”. |  | |
|  | 5a. Menampilkan data berdasarkan data yang di *input*. | |
| 6a. Menekan tombol “ubah”. |  | |
| Aksi Aktor | Reaksi Sistem | |
|  | 7a. Menampilkan halaman *edit* kehadiran siswa. | |
| 8a. Memperbaharui data kehadiran siswa. Hanya status kehadiran yang dapat diperbaharui. |  | |
| 9a. Menekan tombol “simpan”. |  | |
|  | 10a. Memeriksa validasi data yang diperbaharui. | |
|  | 11a. Apabila *valid* maka data akan disimpan di *database* kehadiran\_pengajar. | |
|  | 12b. Menampilkan halaman untuk memilih tanggal dan kelas. | |
| Alur alternatif III | | |
|  | | 11b. Apabila data tidak *valid* maka akan muncul pesan *error.* |
|  | | 12c. Menampilkan halaman form edit kehadiran siswa. |
| Alur alternatif IV | | |
| 9b. Menekan tombol “batal”. | |  |
|  | | 10b. Menampilkan halaman untuk memilih tanggal dan kelas. |
| Alur alternatif V | | |
| 6b. menekan tombol “hapus”. | |  |
| Aksi Aktor | | Reaksi Sistem |
|  | | 7b. Menampilkan pesan “yakin menghapus data ini?”. |
| 8b. Menekan tombol “*Ok*” | |  |
|  | | 9c. data kehadiran otomatis terhapus dari *database*. |
| Alur alternatif VI | | |
| 8c. Menekan tombol “*Cancel*” | |  |
|  | | 9d. Kembali kehalaman untuk memilih tanggal dan kelas. |

#### Usecase Kelola Daftar Hadir Pengajar

Nama Usecase : Mengelola Daftar Hadir Pengajar

Aktor : Bagian Administrasi

Tujuan : Untuk memasukkan dan mengubah daftar hadir pengajar setiap harinya

Deskripsi : Proses ini adalah kegiatan untuk mengelola daftar hadir pengajar yang ada pada bimbingan belajar ini

Pre-condition : Bagian Administrasi sudah login, bagian administrasi sudah berada di menu daftar hadir

Post-condition : Data daftar hadir sudah tersimpan di *database*

Tabel 3.11 Skenario *Usecase* Kelola Daftar Hadir Pengajar

|  |  |  |
| --- | --- | --- |
| Aksi Aktor | Reaksi Sistem | |
| Alur dasar (*basic flow)* | | |
| 1. Memilih menu daftar hadir. |  | |
| 1. Memilih sub menu tambah daftar hadir. |  | |
|  | 1. Menampilkan halaman untuk memilih hari sesuai jadwal yang ada. | |
| 1. Memiliih hari . |  | |
| 1. Menekan tombol “cari” . |  | |
|  | 1. Menampilkan jadwal belajar sesuai dengan hari yang dipilih. | |
| 1. Mengklik “Kehadiran Pengajar”. |  | |
|  | 1. Menampilkan halaman untuk memasukkan kehadiran pengajar. | |
| 1. Mengisikan data. |  | |
| 1. Menekan tombol “simpan”. |  | |
|  | 1. Memeriksa validasi data yang dimasukkan. | |
|  | 1. Apabila *valid* maka data akan disimpan di *database* kehadiran\_pengajar. | |
|  | 1. Menampilkan halaman untuk memasukkan kehadiran pengajar. | |
| Alur alternatif I | | |
|  | 12a. Apabila data tidak *valid* maka akan muncul pesan *error.* | |
| Aksi Aktor | Reaksi Sistem | |
|  | 13a. Kembali kehalaman untuk memasukkan data kehadiran pengajar. | |
| Alur alternatif II | | |
| 10a. Menekan tombol “batal”. |  | |
|  | 11a. Kembali kehalaman untuk memasukkan hari sesuai jadwal. | |
| Alur alternatif III | | |
| 10b. Menekan tombol “Ubah”. | |  |
|  | | 11b. Menampilkan halaman *edit* kehadiran pengajar. |
| 12b. Memperbaharui data kehadiran pengajar. Status kehadiran dan mata pelajaran yang dapat diperbaharui. | |  |
| 13b. Menekkan tombol “simpan”. | |  |
|  | | 14. Memeriksa validasi data yang dimasukkan. |
|  | | 15. Apabila *valid* maka data akan disimpan di *database* kehadiran\_pengajar. |
|  | | 16. Menampilkan halaman untuk memasukkan kehadiran pengajar. |
| Alur alternatif IV | | |
| 13c. Menekan tombol “batal” | |  |
| Aksi Aktor | | Reaksi Sistem |
|  | | 14a. Menampilkan halaman untuk memasukkan kehadiran pengajar. |
| Alur alternatif V | | |
| 10c. menekan tombol “hapus” | |  |
|  | | 11c. Menampilkan pesan “yakin menghapus data ini?” |
| 12c. Menekan tombol “*Ok*” | |  |
|  | | 13d. data kehadiran otomatis terhapus dari *database*. |
| Alur alternatif VI | | |
| 12d. Menekan tombol “*Cancel*” | |  |
|  | | 13e. Kembali kehalaman untuk memasukkan data kehadiran pengajar. |

#### Usecase Kelola Jadwal Belajar Siswa

Nama Usecase : Kelola Jadwal Belajar Siswa

Aktor : Bagian Administrasi

Tujuan : Untuk memasukkan, mengubah dan menghapus jadwal belajar siswa

Deskripsi : Proses ini adalah kegiatan untuk mengelola jadwal belajar siswa yang ada pada bimbingan belajar ini

Pre-condition : Bagian Administrasi sudah login, bagian administrasi sudah berada di menu jadwal

Post-condition : Data jadwal sudah tersimpan di *database*

Tabel 3.12 Skenario *Usecase* Kelola Jadwal Belajar Siswa

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar (*basic flow)* | |
| 1. Memilih menu jadwal belajar. |  |
|  | 1. Menampilkan halaman jadwal belajar serta ruangan. |
| 1. Menekan tombol “Tambah Jadwal Baru”. |  |
|  | 1. Menampilkan halaman tambah jadwal. |
| 1. Mengisi data jadwal. |  |
| 1. Menekan tombol “Simpan” |  |
|  | 1. Memeriksa validasi data yang dimasukkan. |
|  | 1. Apabila *valid* maka data akan disimpan di *database* jadwal. |
|  | 1. Menampilkan halaman daftar jadwal belajar. |
| Alur Alternatif I | |
|  | 8a. Apabila tidak *valid* maka muncul pesan *error*. |
|  | 9a. Kembali ke halaman tambah jadwal. |
| Aksi Aktor | Reaksi Aktor |
| Alur Alternatif II | |
| 6a. Menekan tombol “Batal” |  |
|  | 7a. Kembali ke halaman jadwal belajar serta ruangan. |
| Alur Alternatif III | |
| 3a. Klik “Tambah Detail” |  |
|  | 4a. Menampilkan halaman tambah detail jadwal belajar. |
| 5a. Klik tombol “tambah jadwal baru”. |  |
|  | 6b. Menampilkan halaman form tambah jadwal belajar. |
| 7b. Mengisi data tambah jadwal. |  |
| 8b. Menekan tombol “Simpan”. |  |
|  | 9b. Memeriksa validasi data yang dimasukkan. |
|  | 10. Apabila *valid* maka data akan disimpan di *database* jadwal |
|  | 11. Menampilkan halaman tambah detail jadwal belajar. |
| Alur alternatif IV | |
|  | 10a. Apabila data tidak *valid* maka akan muncul pesan *error.* |
| Aksi Aktor | Reaksi Aktor |
|  | 11a. Menampilkan halaman tambah detail jadwal belajar. |
| Alur alternatif V | |
| 8c. Menekan tombol “Batal”. |  |
|  | 9c. Menampilkan halaman tambah detail jadwal belajar. |
| Alur alternatif VI | |
| 5b. klik tombol “Tambah detail”. |  |
|  | 6c. Menampilkan halaman detail jadwal belajar. |
| 7c. Mengisi form detail jadwal belajar. |  |
| 8d. Klik tombol “simpan”. |  |
|  | 9d. Memeriksa validasi data yang dimasukkan. |
|  | 10b. Apabila *valid* maka data akan disimpan di *database* jadwal ruangan. |
|  | 11b. Menampilkan halaman detail jadwal belajar. |
| Alur alternatf VII | |
|  | 10c. Apabila data tidak *valid* maka akan muncul pesan *error.* |
|  | 11c. Menampilkan halaman detail jadwal belajar. |
| Aksi Aktor | Reaksi Sistem |
| Alur alternatif VIII | |
| 8e. Klik “batal”. |  |
|  | 9e. Menampilkan halaman detail jadwal belajar. |
| Alur alternatif IX | |
| 5c. Klik tombol “ubah”. |  |
|  | 6d. Menampilkan halaman edit jadwal belajar. |
| 7d. Mengisi form edit jadwal belajar |  |
| 8f. Klik tombol “ubah” |  |
|  | 9f. Memeriksa validasi data yang dimasukkan. |
|  | 10d. Apabila *valid* maka data akan disimpan di *database* jadwal ruangan |
|  | 11d. Menampilkan halaman detail jadwal belajar. |
| Alur alternatif X | |
|  | 10e. Apabila data tidak *valid* maka akan muncul pesan *error.* |
|  | 11e. Kembali ke halaman detail jadwal belajar. |
| Alur alternatif XI | |
| 8g. Klik tombol “batal” |  |
| Aksi Aktor | Reaksi Sistem |
|  | 9g. Kembali ke halaman detail jadwal belajar. |
| Alur alternatif XII | |
| 5d. Klik tombol “hapus”. |  |
|  | 6e. Menampilkan pesan “data yakin dihapus?”. |
| 7e. Klik tombol “*Ok*” |  |
|  | 8h. Menampilkan halaman detail jadwal belajar. |
| Alur alternatif XIII | |
| 7f. Klik tombol “*Cancel*” |  |
|  | 8i. menampilkan halaman detail jadwal belajar. |
| Alur alternatif XIV | |
| 5e. Klik tombol “Ubah” |  |
|  | 6e. Menampilkan form ubah jadwal. |
| 7f. Mengisi form ubah jadwal. |  |
| 8j. Klik tombol “ubah”. |  |
|  | 9h. Memeriksa validasi data yang dimasukkan . |
|  | 10f. Apabila *valid* maka data akan disimpan di *database* jadwal. |
|  | 11f. Menampilkan halaman jadwal. |
| Aksi Aktor | Reaksi Sistem |
| Alur Alternatif XV | |
| 8k. Menekan tombol “batal”. |  |
|  | 9i. Kembali ke halaman jadwal. |
| Alur alternatif XVI | |
|  | 10g. Apabila data tidak *valid* maka akan muncul pesan data gagal diinput. |
|  | 11g. Kembali ke halaman form ubah jadwal . |
| Alur alternatif XVII | |
| 8l. Menekan tombol “hapus”. |  |
|  | 9j. Menampilkan pesan “data yakin dihapus?”. |
| 10h. Klik tombol “*Ok*” |  |
|  | 11h. Menampilkan halaman jadwal. |
| Alur alternatif XVIII | |
| 10i. Menekan tombol “*Cancel*”. |  |
|  | 11i. Menampilkan halaman jadwal. |
| Alur alternatif XIX | |
| 3b. Memilih hari . |  |
| 4b. Klik tombol “cari!”. |  |
|  | 5f. Menampilkan jadwal berdasarkan hari. |

#### Usecase Kelola Data Pembayaran Siswa

Nama Usecase : Kelola Data Pembayaran Siswa

Aktor : Bagian Keuangan

Tujuan : Untuk memasukkan, mengubah dan menghapus data pembayaran siswa

Deskripsi : Proses ini adalah kegiatan untuk mengelola data pembayaran siswa yang ada pada bimbingan belajar ini

Pre-condition : Bagian Keuangan sudah login, bagian keuangan sudah berada di menu pembayaran

Post-condition : Data pembayaran sudah tersimpan di *database*

Tabel 3.13 Skenario *Usecase* Kelola Data Pembayaran Siswa

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar (*basic flow)* | |
| 1. Memilih menu kelola pembayaran. |  |
|  | 1. Menampilkan halaman untuk mencari nama siswa. |
| 1. Mengisi kolom *search* dengan nama siswa. |  |
| 1. Menekan tombol “cari” |  |
|  | 1. Menampilkan data siswa sesuai dengan kata yang dicari. |
| 1. Memilih salah satu siswa untuk mengisi data pembayaran. |  |
|  | 1. Menampilkan halaman *input* data pembayaran sesuai dengan nama siswa yang dipilih. |
| Aksi Aktor | Reaksi Sistem |
| 1. Mengisikan data pembayaran. |  |
| 1. Menekan tombol “simpan” |  |
|  | 1. Memeriksa validasi data yang dimasukkan. |
|  | 1. Apabila *valid* maka data akan disimpan di *database* transaksi. |
|  | 1. Menampilkan halaman input data pembayaran dan menampilkan data pembayaran sesuai dengan nama siswa. |
| Alur alternatif I | |
|  | 10a. Apabila data tidak *valid* maka akan menampilkan pesan *error.* |
|  | 11a. Kembali ke halaman untuk mencari nama siswa. |
| Alur alternatif II | |
| 9a. Menekan tombol “batal”. |  |
|  | 10b. Menampilkan halaman untuk mencari nama siswa. |
| Alur alternatif III | |
| 9b. Menekan tombol “ubah”. |  |
|  | 10c. Menampilkan halaman *edit* pembayaran. |
| 11b. memperbaharui data pembayaran. Beberapa pilihan data yang dapat diperbaharui jumlah bayar dan keterangan. |  |
| Aksi Aktor | Reaksi Sistem |
| 12a. Menekan tombol “simpan”. |  |
|  | 13. memeriksa validitas data yang diperbaharui. |
|  | 14. apabila *valid* maka data akan disimpan di *database* transaksi. |
|  | 15. Menampilkan halaman data pembayaran yang datanya sudah diubah. |
| Alur Alternatif IV | |
|  | 13b. Apabila tidak *valid* maka akan menampilkan pesan *error*. |
|  | 14d. Kembali ke form *edit* pembayaran. |
| Alur Alternatif V | |
| 9c. menekan tombol “hapus”. |  |
|  | 10d. Menampilkan pesan “yakin menghapus data ini?”. |
| 11c. Menekan tombol “*Ok*”. |  |
|  | 12b. data pembayaran otomatis terhapus dari *database*. |
| Alur alternatif VI | |
| 9c. Menekan tombol “*Cancel*”. |  |
|  | 10d. Kembali ke halaman untuk mencari nama siswa. |

#### Usecase Kapasistas Ruangan

Nama Usecase : Kapasitas Ruangan

Aktor : Pengelola

Tujuan : Untuk melihat kapasitas ruangan

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat kapasitas ruangan yang telah diinputkan oleh bagian administrasi

Pre-condition : Pengelola sudah login, pengelola memilih menu ruangan, cek kapasitas ruangan

Post-condition : Sistem menampilkan data kapasitas ruangan

Tabel 3.14 Skenario *Usecase* Kapasitas Ruangan

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu ruangan. |  |
| Aksi Aktor | Reaksi Sistem |
|  | 1. Menampilkan data ruangan. |
| 1. Melihat kapasitas ruangan. |  |

#### Usecase Laporan Keuangan

Nama Usecase : Laporan Keuangan

Aktor : Pengelola

Tujuan : Untuk melihat Laporan Keuangan

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat laporan keuangan yang berasal dari data pembayaran yang sudah diinputkan oleh bagian keuangan

Pre-condition : Pengelola sudah login, pengelola memilih menu keuangan

Post-condition : Sistem menampilkan data pembayaran

Tabel 3.15 Skenario *Usecase* Laporan Keuangan

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu laporan. |  |
| 1. Memilih sub menu laporan keuangan. |  |
|  | 1. Menampilkan halaman untuk mencari data keuangan. |
| 1. Memilih bulan. |  |
| 1. Menekan tombol “cari” |  |
|  | 1. Menampilkan laporan keuangan berdasarkan bulan yang dipilih. |
| 1. Melihat laporan keuangan. |  |
| Alur Alternatif I | |
| 5a. Menekan tombol “*download*”. |  |
| Aksi Aktor | Reaksi Sistem |
|  | 6a. Mengunduh data yang sedang ditampilkan. |

#### Usecase Melihat Jumlah Siswa

Nama Usecase : Laporan Jumlah Siswa

Aktor : Pengelola

Tujuan : Untuk melihat jumlah siswa pada bimbingan belajar ini

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat jumlah siswa yang berasal dari data siswa yang telah dikelola bagian administrasi

Pre-condition : Pengelola sudah login, pengelola memilih menu siswa, laporan jumlah siswa

Post-condition : Sistem menampilkan jumlah siswa berupa grafik

Tabel 3.16 Skenario *Usecase* Laporan Jumlah Siswa

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu laporan. |  |
| 1. Memilih sub menu laporan jumlah murid. |  |
|  | 1. Menampilkan data jumlah siswa. |
| 1. Melihat laporan jumlah siswa. |  |

#### Usecase Data Pengajar

Nama Usecase : Data Pengajar

Aktor : Pengelola

Tujuan : Untuk melihat data pengajar

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat data pengajar yang terlah diinputkan bagian administrasi

Pre-condition : Pengelola memilih menu pengajar

Post-condition : Sistem menampilkan data pengajar

Tabel 3.17 Skenario *Usecase* Data Pengajar

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu pengajar. |  |
|  | 1. Menampilkan data pengajar. |
| 1. Melihat data pengajar |  |

#### Usecase Kelola Data Siswa Baru

Nama Usecase : Kelola Data Siswa Baru

Aktor : Pengelola

Tujuan : Untuk memasukkan data siswa baru

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk memasukkan data siswa setelah siswa baru mendaftar

Pre-condition : Pengelola sudah login, pengelola sudah berada didalam menu siswa, input siswa baru

Post-condition : Data siswa sudah tersimpan

Tabel 3.18 Skenario *Usecase* Kelola Data Siswa Baru

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu siswa. |  |
| 1. Memilih sub menu tambah data siswa. |  |
|  | 1. Menampilkan form tambah data siswa. |
| 1. Mengisikan data siswa. |  |
| 1. Menekan tombol “simpan”. |  |
|  | 1. Memeriksa validasi data yang dimasukkan |
|  | 1. Apabila *valid* maka data akan disimpan di *database* siswa dan akan menampilkan pesan “data berhasil disimpan” |
|  | 1. Menampilkan halaman siswa. |
| Alur alternatif I | |
|  | 6a. Apabila data tidak *valid* maka akan menampilkan pesan “data tidak berhasil disimpan” |
|  | 7a. Kembali ke form tambah data siswa. |
| Alur alternatif II | |
| 5a. Menekan tombol “batal”. |  |
| Aksi Aktor | Reaksi Sistem |
|  | 6b. Menampilkan form tambah data siswa. |
| Alur alternatif III | |
| 4a. Memilih kelas untuk cek kuota kelas |  |
|  | 5b. Menampilkan kuota kelas sesuai dengan kelas yang dipilih. |

#### Usecase Data Siswa

Nama Usecase : Data Siswa

Aktor : Pengelola

Tujuan : Untuk melihat data siswa pada bimbingan belajar ini

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat data siswa yang sudah diinputkan oleh bagian administrasi

Pre-condition : Pengelola sudah login, pengelola sudah berada didalam menu siswa, laporan jumlah siswa

Post-condition : Sistem menampilkan data siswa

Tabel 3.19 Skenario *Usecase* Data Siswa

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu siswa |  |
| 1. Memilih sub menu lihat data siswa. |  |
| Aksi Aktor | Reaksi Sistem |
|  | 1. Menampilkan halaman lihat data siswa dengan fitur *search*. |
| 1. Memasukkan nama siswa pada kolom yang sediakan. |  |
| 1. Menekan tombol “cari”. |  |
|  | 1. Menampilkan data siswa sesuai dengan nama yang dicari. |

#### Usecase Jadwal Belajar

Nama Usecase : Jadwal Belajar

Aktor : Pengelola

Tujuan : Untuk melihat jadwal belajar pada bimbingan belajar ini

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat jadwal belajar siswa yang telah diinputkan oleh bagian administrasi

Pre-condition : Pengelola sudah login, pengelola sudah berada didalam menu jadwal

Post-condition : Sistem menampilkan data jadwal

Tabel 3.20 Skenario *Usecase* Jadwal Belajar

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu jadwal belajar. |  |
| Aksi Aktor | Reaksi Sistem |
|  | 1. Menampilkan halaman untuk mencari jadwal. |
| 1. Memilih kelas untuk dilihat jadwal belajar. |  |
|  | 1. Menampilkan data jadwal belajar sesuai dengan kelas. |
| 1. Melihat data jadwal belajar. |  |

#### Usecase Daftar Hadir Pengajar

Nama Usecase : Daftar Hadir Pengajar

Aktor : Pengelola

Tujuan : Untuk melihat daftar hadir pengajar

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat daftar hadir pengajar yang telah diinputkan oleh bagian administrasi

Pre-condition : Pengelola sudah login, pengelola sudah berada didalam sub menu kehadiran pengajar

Post-condition : Sistem menampilkan data daftar hadir pengajar

Tabel 3.21 Skenario *Usecase* Daftar Hadir Pengajar

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih sub kehadiran pengajar. |  |
|  | 1. Menampilkan halaman untuk mencari tanggal kehadiran pengajar. |
| 1. Memasukkan tanggal. |  |
|  | 1. Menampilkan data daftar hadir pengajar sesuai dengan tanggal yang dimasukkan. |
| 1. Melihat data daftar hadir pengajar. |  |

#### Usecase Daftar Hadir Siswa

Nama Usecase : Daftar Hadir Siswa

Aktor : Pengelola

Tujuan : Untuk melihat daftar hadir siswa

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk melihat daftar hadir siswa yang telah diinputkan oleh bagian administrasi

Pre-condition : Pengelola sudah login, pengelola sudah berada didalam sub menu kehadiran siswa

Post-condition : Sistem menampilkan data daftar hadir siswa

Tabel 3.22 Skenario *Usecase* Daftar Hadir Siswa

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih sub kehadiran siswa. |  |
|  | 1. Menampilkan halaman untuk mencari tanggal dan kelas kehadiran siswa. |
| 1. Memasukkan tanggal dan kelas. |  |
|  | 1. Menampilkan data daftar hadir siswa sesuai dengan tanggal dan kelas yang dimasukkan. |
| 1. Melihat data daftar hadir siswa. |  |

#### Usecase Kelola Kuota Kelas

Nama Usecase : Kelola Kuota Kelas

Aktor : Pengelola

Tujuan : Untuk mengelola kuota kelas

Deskripsi : Proses ini adalah kegiatan untuk mengelola kuota kelas yang ada pada bimbingan belajar pada setiap tahun ajaran

Pre-condition : Pengelola sudah login, pengelola sudah berada didalam menu kuota kelas

Post-condition : Sistem menampilkan data kuota kelas

Tabel 3.23 Skenario *Usecase* Kelola Kuota Kelas

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar (*basic flow)* | | |
| 1. Memilih menu kuota kelas. |  |
|  | 1. Menampilkan halaman kuota kelas. |
| 1. Menekan tombol “tambah kuota kelas”. |  |
|  | 1. Menampilkan *form* tambah kuota kelas. |
| 1. Mengisikan data kuota kelas. |  |
| 1. Menekan tombol “simpan”. |  |
|  | 1. Memeriksa validitas data yang dimasukkan. |
|  | 1. Apabila *valid* maka data akan disimpan di *database* kuota kelas. |
|  | 1. Menampilkan daftar kuota kelas. |
| Alur Alternatif I | | |
|  | 8a. Apabila data tidak *valid* maka akan muncul pesan *error.* |
|  | 9a. Tetap dihalaman *form* tambah kuota kelas. |
| Alur Alternatif II | | |
| 6a. Menekan tombol “batal”. |  |
|  | 7a. Menampilkan daftar kuota kelas. |
| Alur alternatif III | | |
| 3a. Menekan tombol “ubah”. |  |
|  | 4a. Menampilkan halaman *edit* kuota kelas. |
| 5a. memperbaharui data kuota kelas. Beberapa pilihan data yang dapat diperbaharui kelas dan jumlah. |  |
| 6b. Menekan tombol “simpan”. |  |
|  | 7b. memeriksa validitas data yang diperbaharui. |
|  | 8b. apabila *valid* maka data akan disimpan di *database* kuota kelas. |
|  | 9b. Menampilkan daftar kuota kelas. |
| Alur alternatif IV | | |
|  | 8c. Apabila tidak *valid* maka akan menampilkan pesan *error*. |
|  | 9c. Kembali ke *form edit* kuota kelas. |
| Alur alternatif V | | |
| 6c. Menekan tombol “batal” |  |
|  | 7c. Menampilkan halaman daftar kuota kelas. |
| Alur alternatif VI | | |
| 3b. menekan tombol “hapus”. |  |
|  | 4b. Menampilkan pesan “yakin menghapus data ini?”. |
| 5b. Menekan tombol “*Ok*”. |  |
|  | 6d. data kuota kelas otomatis terhapus dari *database*. |
| Alur alternatif VII | | |
| 5c. Menekan tombol “*Cancel*”. |  |
|  | 5d. Kembali ke daftar kuota kelas. |

#### Usecase Kelola SMS Gateway

Nama Usecase : Kelola SMS Gateway

Aktor : SMS Gateway

Tujuan : Untuk mengirim SMS pemberitahuan

Deskripsi : Proses ini adalah kegiatan yang dilakukan pengelola untuk mengirimkan SMS pemberitahuan kegiatan atau yang lain nya kepada orang tua siswa

Pre-condition : SMS Gateway sudah berada didalam menu SMS

Post-condition : Sistem mengirimkan SMS

Tabel 3.24 Skenario *Usecase* Kelola SMS Gateway

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Memilih menu SMS. |  |
|  | 1. Menampilkan halaman untuk mengirimkan SMS. |
| 1. Memasukkan kata-kata yang ingin di SMS. |  |
| 1. Menekan tombol “Kirim” |  |
|  | 1. Mengirimkan SMS kepada nomor tujuan. |
| Alur alternatif I | |
| 4a. Menekan tombol “batal” |  |
|  | 5a. Kembali kehalaman utama pengelola. |

#### Usecase Terima SMS Pemberitahuan

Nama Usecase : Terima SMS Pemberitahuan

Aktor : Orang Tua

Tujuan : Untuk menerima SMS pemberitahuan

Deskripsi : Proses ini adalah kegiatan pasif yang dilakukan orang tua untuk menerima SMS pembeitahuan yang telah diproses oleh aktor SMS Gateway.

Post-condition : Orang Tua mendapatkan SMS pemberitahuan

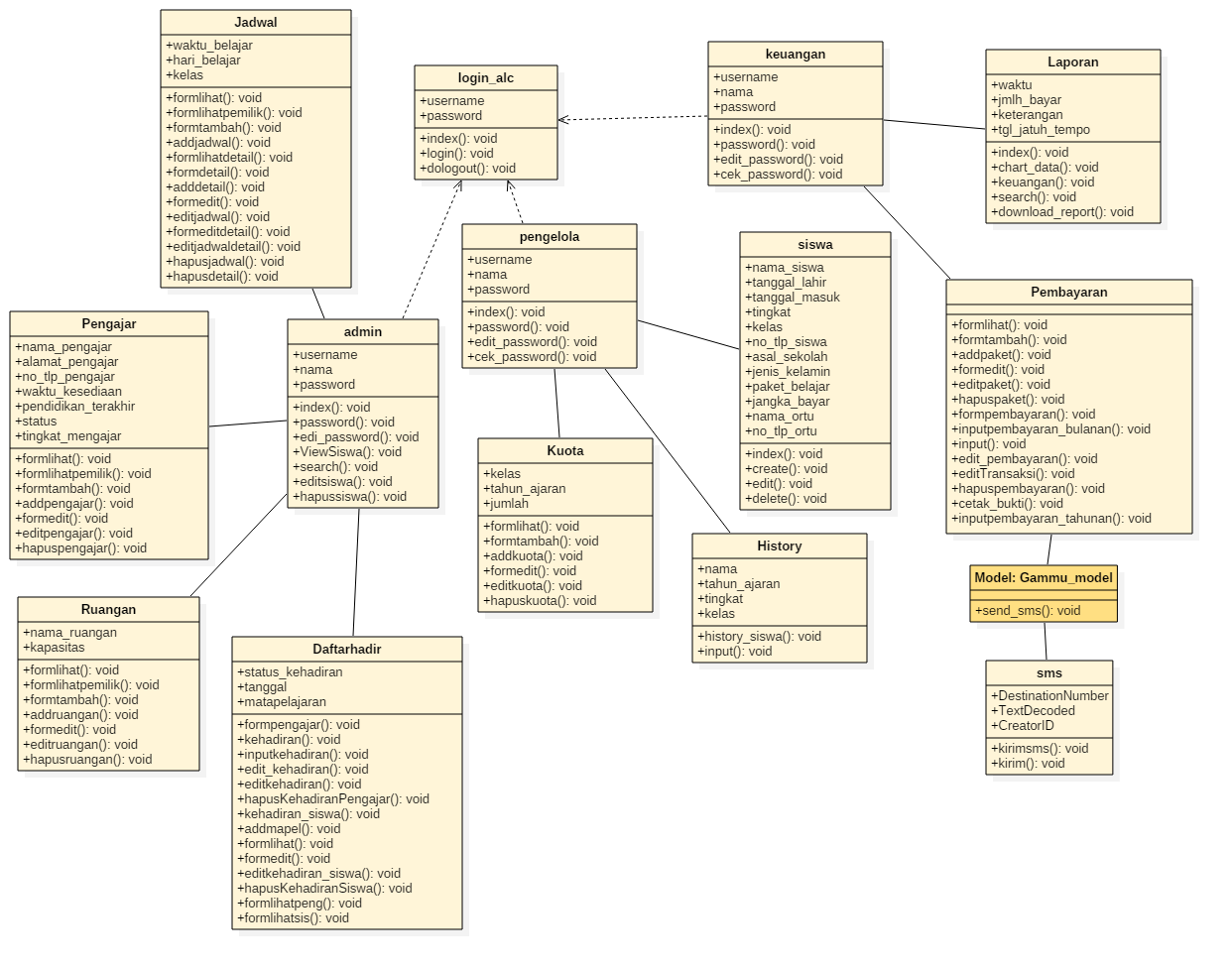
Tabel 3.25 Skenario *Usecase* Terima SMS Pemberitahuan

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Alur dasar *(basic flow)* | |
| 1. Masuk ke menu sms |  |
|  | 1. Menampilkan halaman inbox pada menu SMS |
| 1. Menerima SMS Pemberitahuan. |  |
|  | 1. Menampilkan data SMS Pemberitahuan sesuai data yang dikirmkan dari aktor SMS Gateway. |
| 1. Melihat SMS Pemberitahuan. |  |

* + 1. **Class Diagram**

**3.2.2.1 Definisi Class Diagram**

Class Diagram digunakan untuk menunjukan atau menggambarkan kelas-kelas yang ada pada sistem yang akan dibangun dan juga hubungannya antara satu dengan yang lainnya, yang juga di dalamnya terdapat atribut dan juga operasi yang akan dilakukan oleh system. Class dapat berhubungan dengan yang lain melalui berbagai cara: Associated (terhubung satu sama lain), Dependent (satu class tergantung/menggunakan class yang lain), Specialed (satu class merupakan spesialisasi dari classlainnya), atau Package (grup bersama sebagai satu unit).

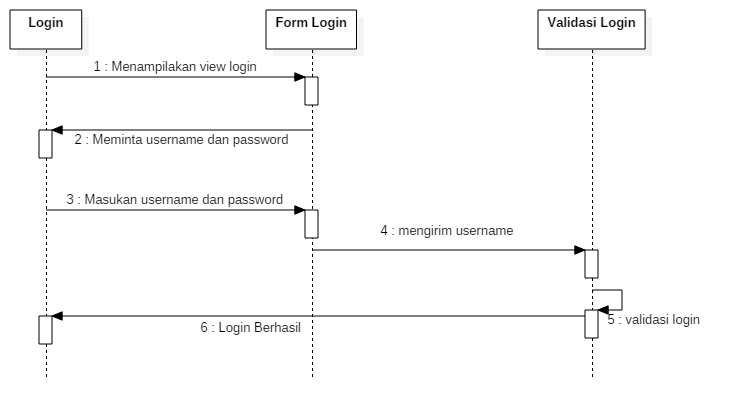
****

Gambar 3.15 Class Diagram Sistem Yang Akan Dibangun

* + 1. **Sequence Diagram**

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

* + - 1. **Sequence Diagram Login**

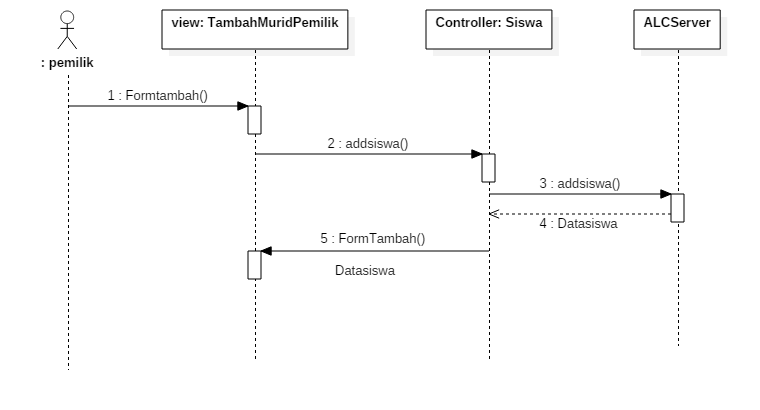
****

Gambar 3.16 Sequence Diagram Login.

Keterangan :

Pada sequence diagram login yakni pemilik, bagian administrasi, dan bagian keuangan melakukan input username dan password kemudian masuk ke form login, kemudian setelah username dan password masuk ke dalam database maka system akan melakukan validasi data login kemudian masuk ke halaman menu utama. Jika gagal system akan kembali ke form login.

* + - 1. **Sequence Diagram Tambah Data Siswa Pemilik**

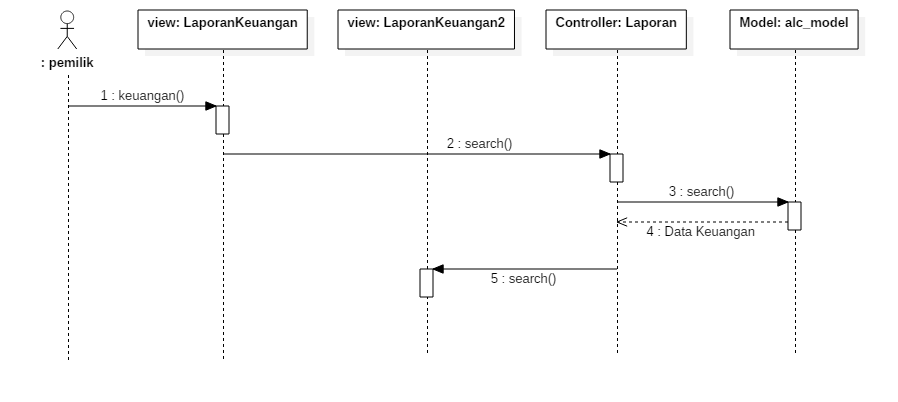
****

Gambar 3.17 Sequence Diagram Tambah Data Siswa Pemilik.

Keterangan :

Pada sequence di atas menjelaskan mengenai pemilik melakukan pendaftaran terhadap murid baru atau siswa baru yang akan mendaftar bimbingan belajar.

* + - 1. **Sequence Diagram Laporan Keuangan**

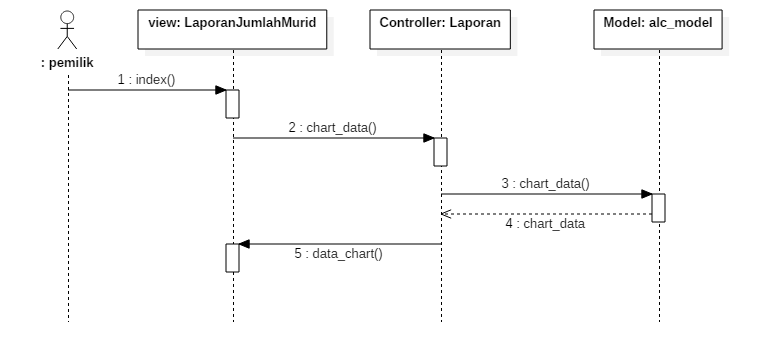
****

Gambar 3.18 Sequence Diagram Laporan Keuangan.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat melihat laporan keuangan yang dilaporkan oleh bagian keuangan.

* + - 1. **Sequence Diagram Melihat Laporan Jumlah Siswa**

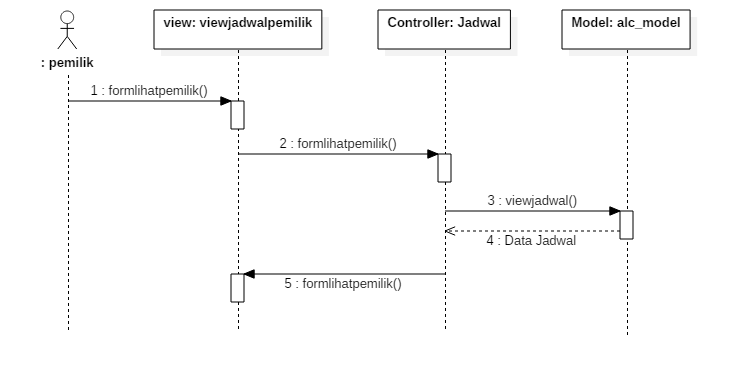
****

Gambar 3.19 Sequence Diagram Melihat Laporan Jumlah Siswa.

Keterangan :

Pada sequence di atas menjelaskan proses pemilik dapat melihat hasil laporan mengenai jumlah siswa yang ada pada bimbingan belajar ALC.

* + - 1. **Sequence Diagram Melihat Jadwal Belajar**

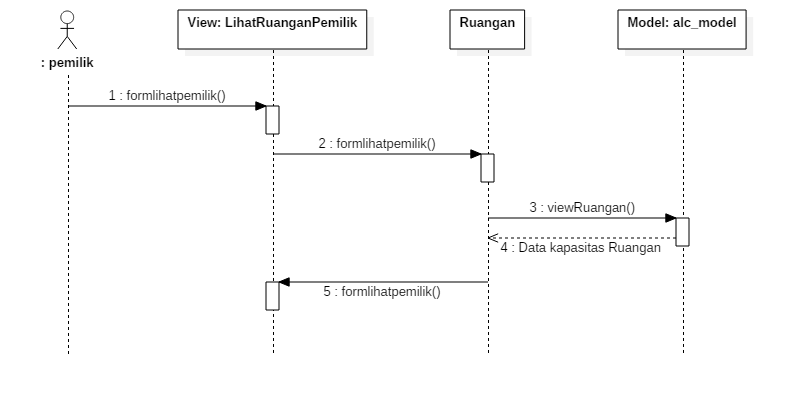
****

Gambar 3.20 Sequence Diagram Melihat Jadwal Belajar.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat melihat jadwal belajar yang sudah terjadwal di bimbingan belajar ALC.

* + - 1. **Sequence Melihat Kapasitas Ruangan**

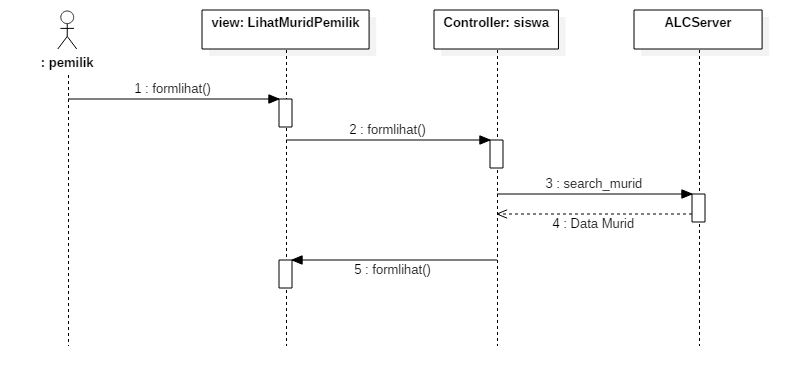
****

Gambar 3.21 Sequence Diagram Melihat Kapasitas Ruangan.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat melihat kapasitas ruangan yang digunakan untuk melakukan bimbingan belajar ALC.

* + - 1. **Sequence Diagram Melihat Data Siswa**

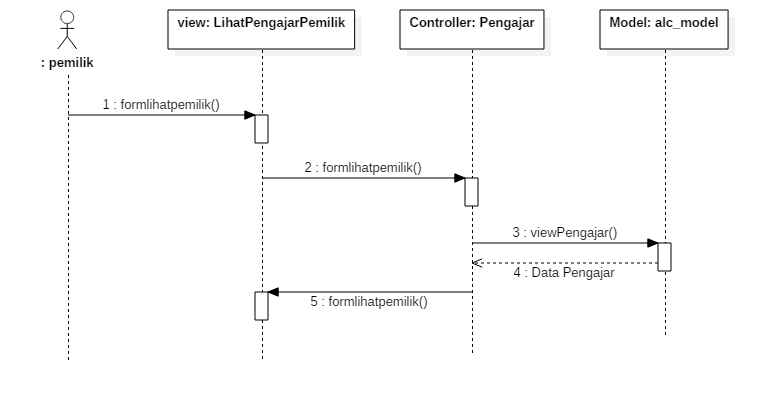
****

Gambar 3.22 Sequence Diagram Melihat Data Siswa.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat melihat data siswa pada bimbingan belajar ALC.

* + - 1. **Sequence Diagram Melihat Data Pengajar**

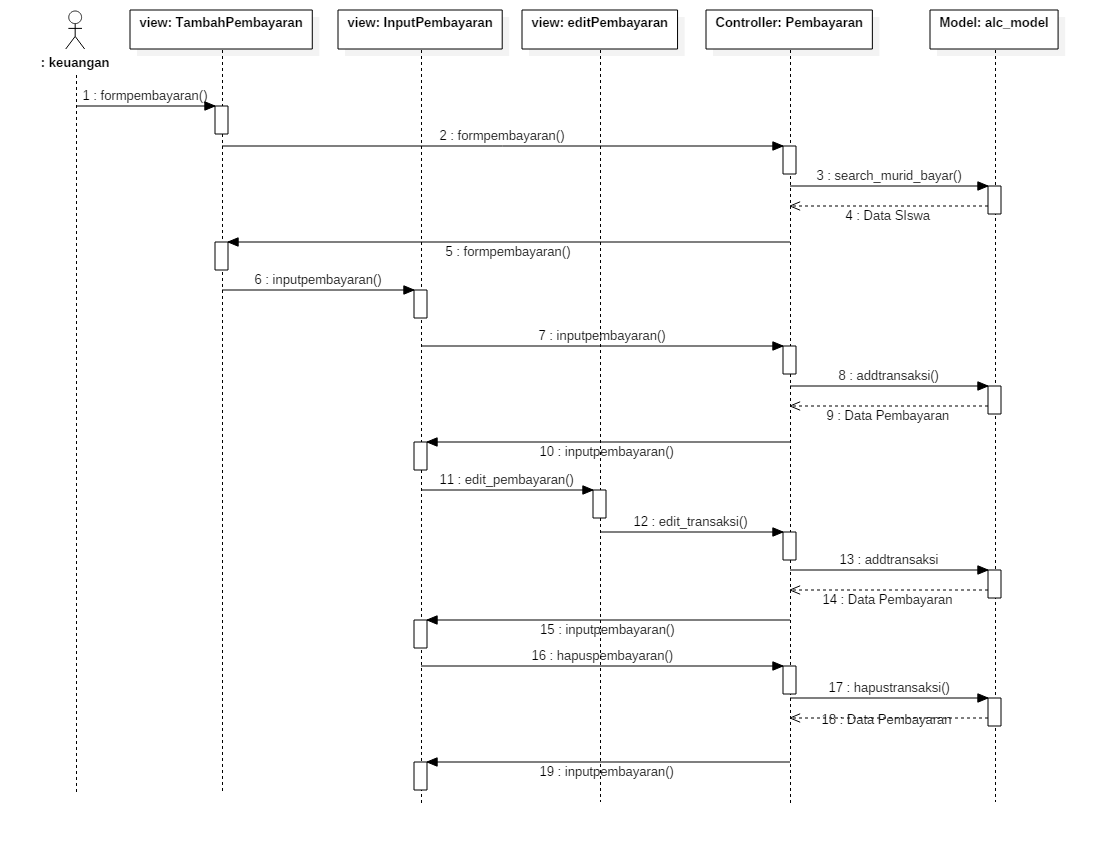
****

Gambar 3.23 Sequence Diagram Melihat Data Pengajar.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat melihat laporang tentang data dari pengajar yang ada pada bimbingan belajar ALC.

* + - 1. **Sequence Diagram Mengelola Data Pembayaran**

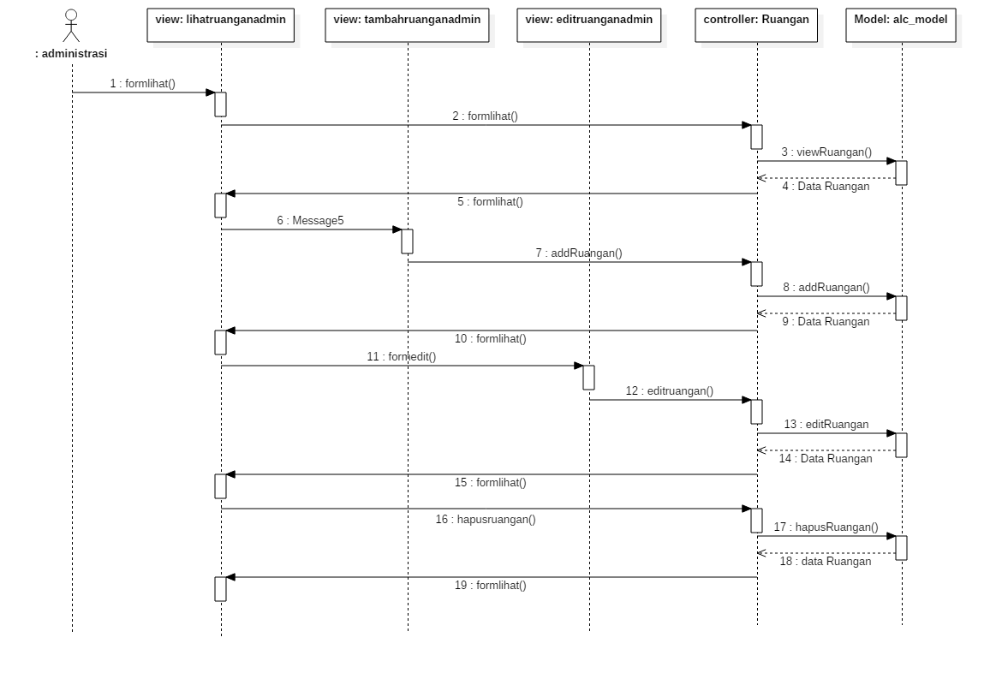
****

Gambar 3.24 Sequence Diagram Mengelola Data Pembayaran.

Keterangan :

Pada sequence di atas menjelaskan pada bagian keuangan mengelola pembayaran dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam pembayaran .

* + - 1. **Sequence Diagram Mengelola Data Ruangan**

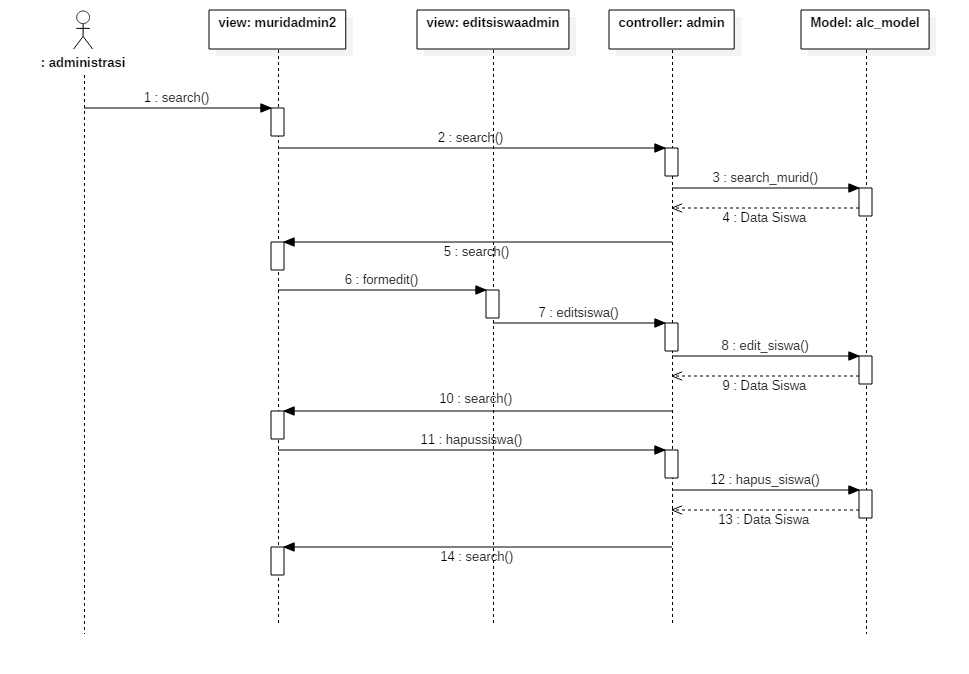
****

Gambar 3.25 Sequence Diagram Mengelola Data Ruangan.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola ruangan dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam ruangan.

* + - 1. **Sequence Diagram Mengelola Data Siswa**

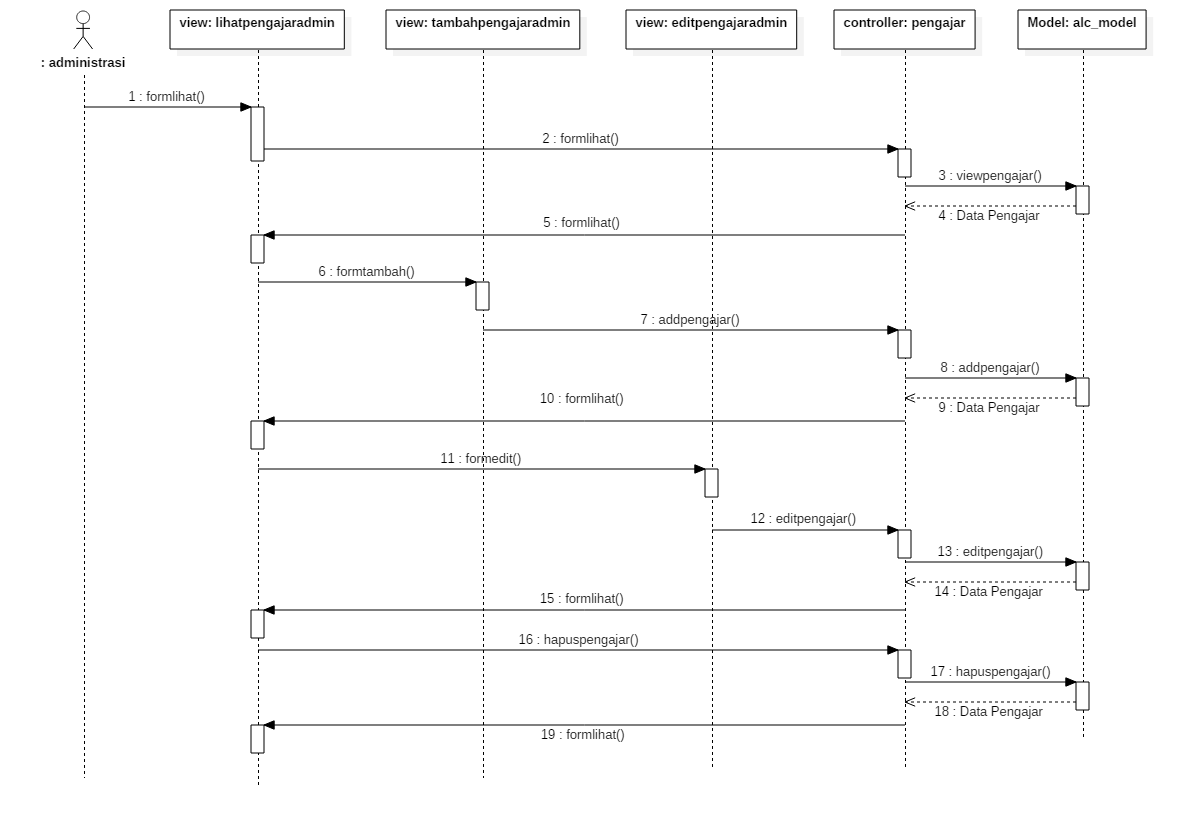
****

Gambar 3.26 Sequence Diagram Mengelola Data Siswa.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola data siswa dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam pembayaran .

* + - 1. **Sequence Diagram Mengelola Data Pengajar**

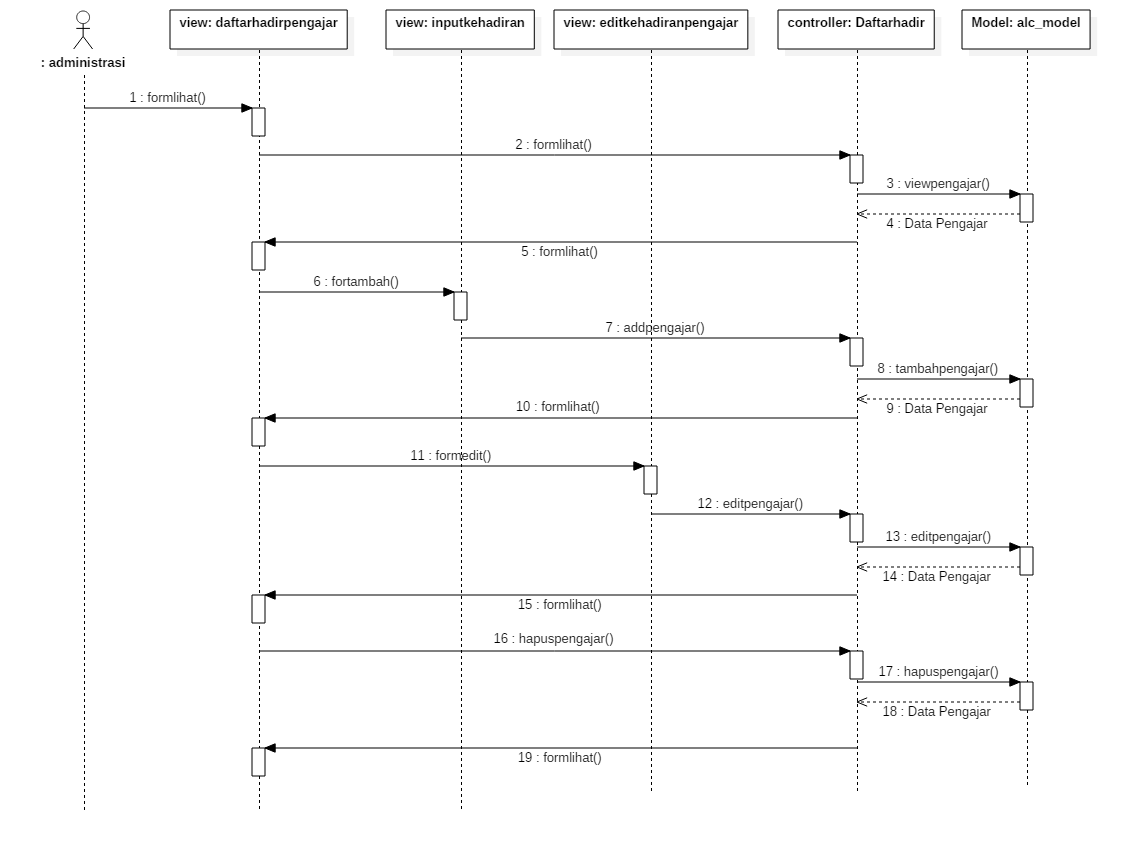
****

Gambar 3.27 Sequence Diagram Mengelola Data Pengajar.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola data pengajar dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam data pengajar.

* + - 1. **Sequence Diagram Mengelola Daftar Hadir Pengajar**

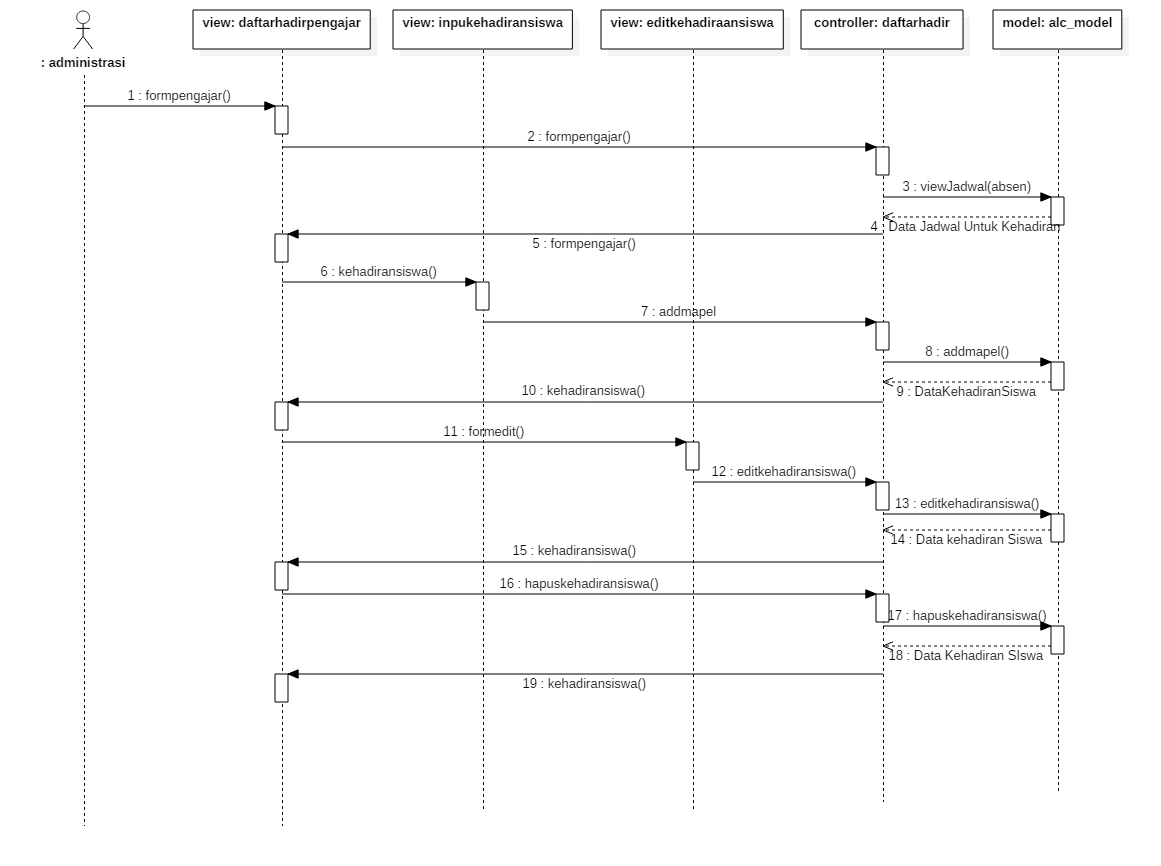
****

Gambar 3.28 Sequence Diagram Mengelola Daftar Hadir Pengajar.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola daftar hadir pengajar dalam kelas bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam daftar hadir pengajar.

* + - 1. **Sequence Diagram Mengelola Daftar Hadir Siswa**

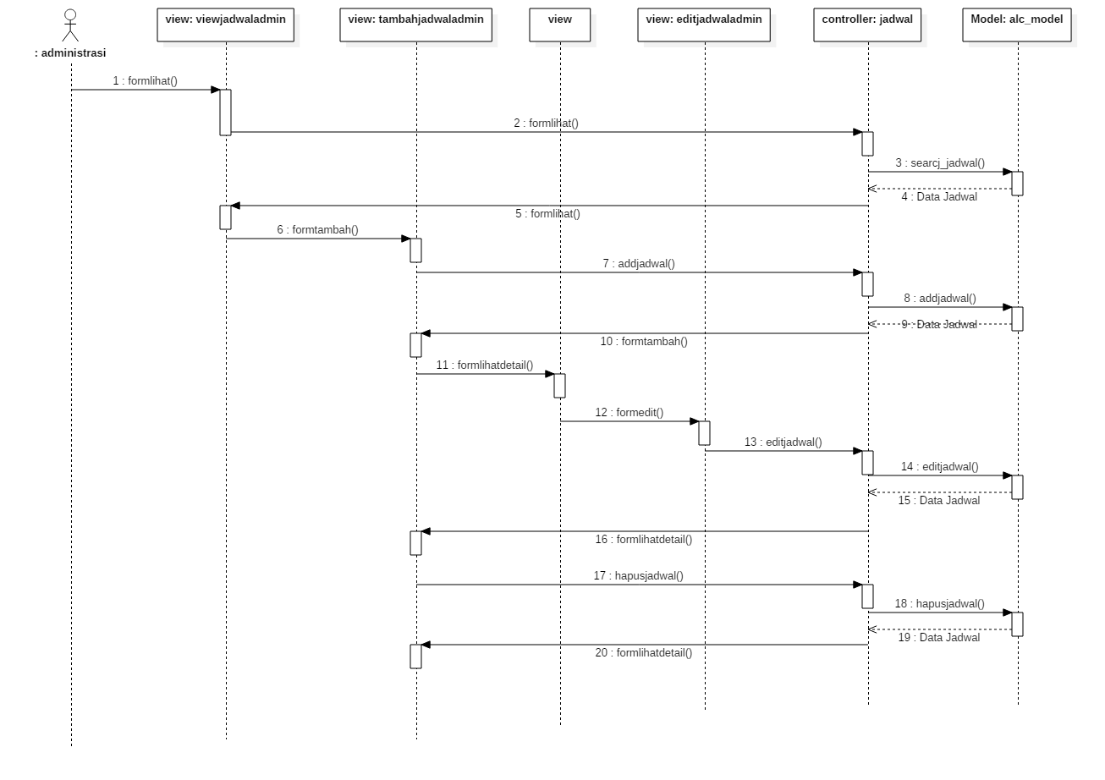
****

Gambar 3.29 Sequence Diagram Mengelola Daftar Hadir Siswa.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola daftar hadir siswa dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam daftar hadir siswa setiap siswa masuk kelas.

* + - 1. **Sequence Diagram Mengelola Jadwal Belajar**

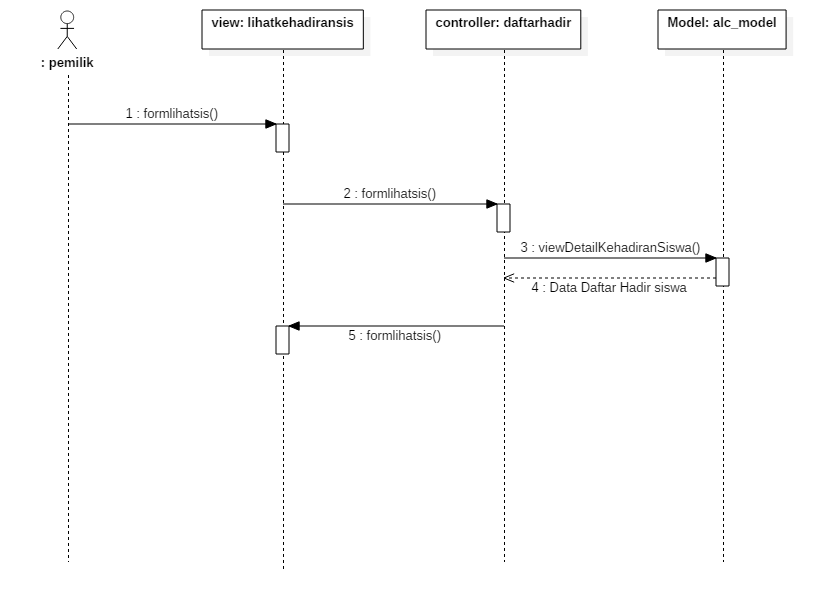
****

Gambar 3.30 Sequence Diagram Mengelola Jadwal Belajar.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola jadwal belajar dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam jadwal belajar .

* + - 1. **Sequence Diagram Melihat Daftar Hadir Siswa**

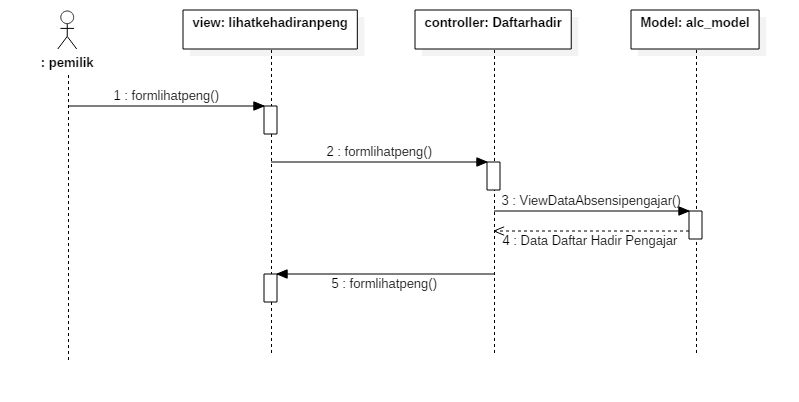
****

Gambar 3.31 Sequence Diagram Melihat Daftar Hadir Siswa.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat memantau dan melihat daftar hadir siswa pada bimbingan belajar ALC.

* + - 1. **Sequence Diagram Melihat Daftar Hadir Pengajar**

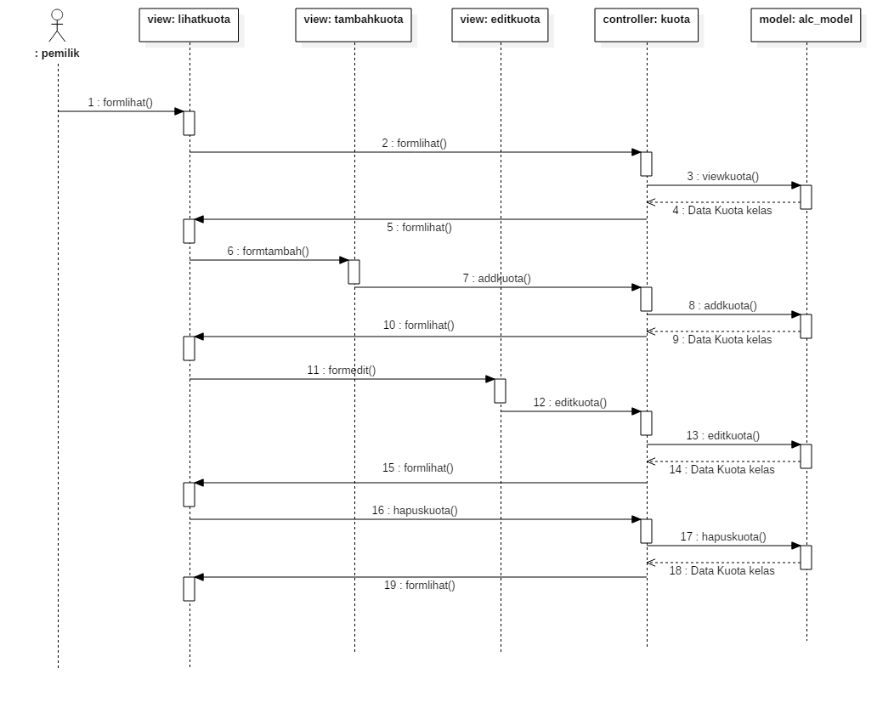
****

Gambar 3.32 Sequence Diagram Melihat Daftar Hadir Pengajar.

Keterangan :

Pada sequence di atas menjelaskan bahwa pemilik dapat memantau dan melihat daftar hadir pengajar pada bimbingan belajar ALC.

* + - 1. **Sequence Diagram Mengelola Kuota Kelas**

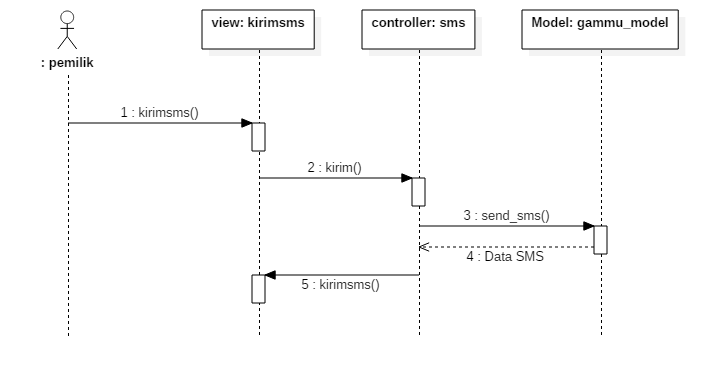
****

Gambar 3.33 Sequence Diagram Mengelola Kuota Kelas.

Keterangan :

Pada sequence di atas menjelaskan pada bagian administrasi mengelola kuota kelas dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada.

* + - 1. **Sequence Diagram Mengirim SMS Pemberitahuan**

****

Gambar 3.34 Sequence Diagram Mengirim SMS Pemberitahuan.

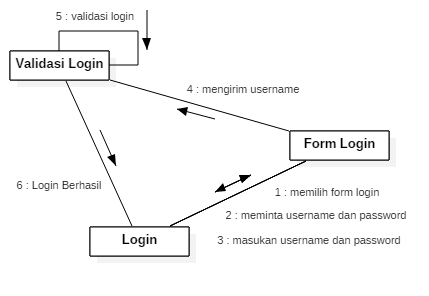
Keterangan :

Pada sequence di atas menjelaskan pemilik dapat mengirimkan pesan pemberitahuan kepada orang tua dari siswa bimbingan belajar ALC.

* + 1. **Collaboration Diagram**

Collaboration diagram merupakan diagram yang mengelompokan pesan pada kumpulan diagram sekuen menjadi sebuah diagram. Pada collaboration diagram ini, objek harus melakukan sinkronisasi pesan dengan serangkaian pesan – pesan lainnya.

* + - 1. **Collaboration Diagram Login**

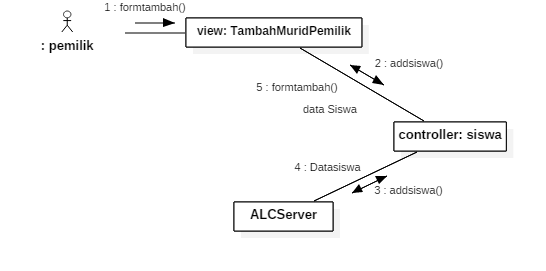
****

Gambar 3.35 Collaboration Diagram Login.

Keterangan:

Pada gambar diatas terdapat Collaboration diagram login yakni pemilik, bagian administrasi, dan bagian keuangan melakukan input username dan password kemudian masuk ke form login, kemudian setelah username dan password masuk ke dalam database maka system akan melakukan validasi data login kemudian masuk ke halaman menu utama. Jika gagal system akan kembali ke form login.

* + - 1. **Collaboration Diagram Tambah Data Siswa Pemilik**

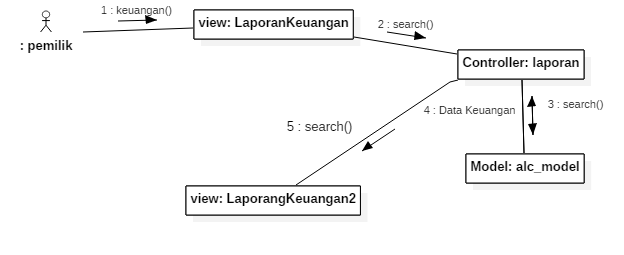
****

Gambar 3.36 Collaboration Diagram Tambah Data Siswa Pemilik.

Keterangan:

Pada gambar Collaboration diagram menjelaskan mengenai pemilik melakukan pendaftaran terhadap murid baru atau siswa baru yang akan mendaftar bimbingan belajar.

* + - 1. **Collaboration Diagram Laporan Keuangan**

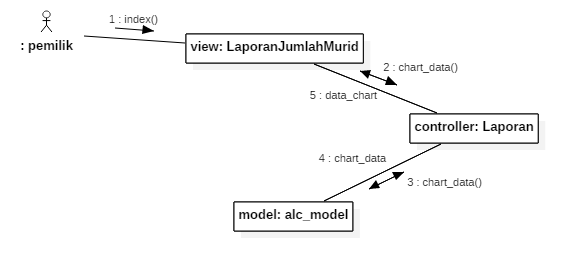
****

Gambar 3.37 Collaboration Diagram Laporan Keuangan.

Keterangan:

Pada gambar collaboration diagram di atas menjelaskan bahwa pemilik dapat melihat laporan keuangan yang dilaporkan oleh bagian keuangan.

* + - 1. **Collaboration Diagram Melihat Laporan Jumlah Siswa**

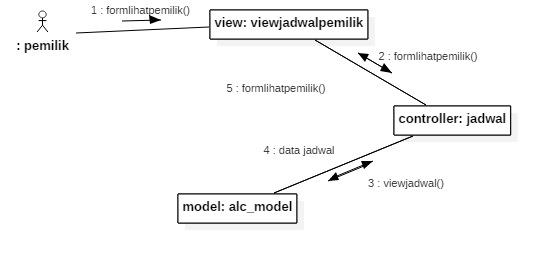
****

Gambar 3.38 Collaboration Diagram Laporan Jumlah Siswa.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan proses pemilik dapat melihat hasil laporan mengenai jumlah siswa yang ada pada bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Melihat Jadwal Belajar**

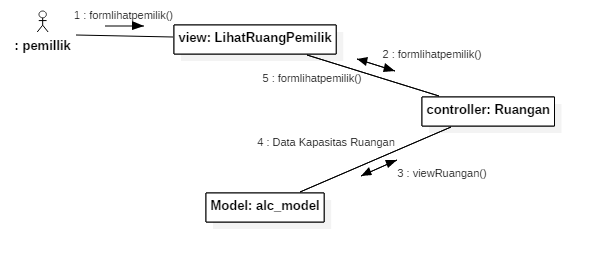
****

Gambar 3.39 Collaboration Diagram Melihat Jadwal Belajar.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan bahwa pemilik dapat melihat jadwal belajar yang sudah terjadwal di bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Melihat Kapasitas Ruangan**

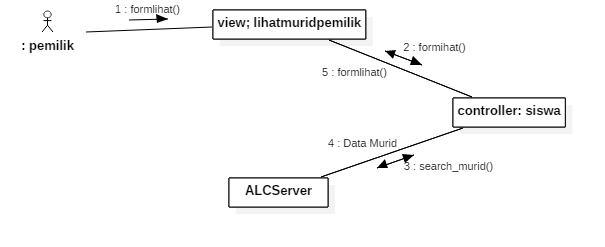
****

Gambar 3.40 Collaboration Diagram Melihat Kapasitas Ruangan.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan bahwa pemilik dapat melihat kapasitas ruangan yang digunakan untuk melakukan bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Melihat Data Siswa**

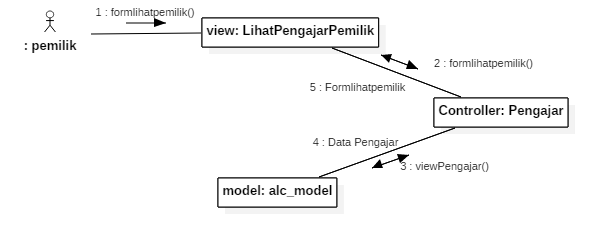
****

Gambar 3.41 Collaboration Diagram Melihat Data Siswa.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan bahwa pemilik dapat melihat data siswa pada bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Melihat Data Pengajar**

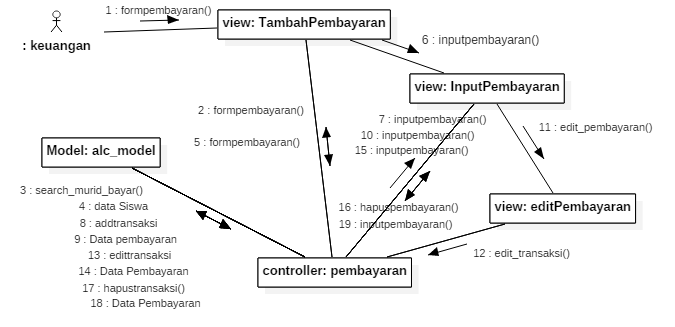
****

Gambar 3.42 Collaboration Diagram Melihat Data Pengajar.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan bahwa pemilik dapat melihat laporang tentang data dari pengajar yang ada pada bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Mengelola Data Pembayaran**

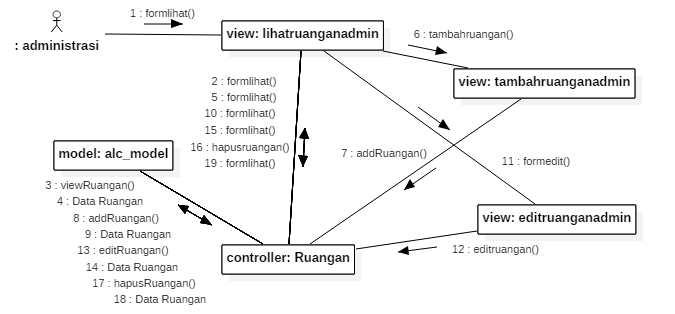
****

Gambar 3.43 Collaboration Diagram Mengelola Data Pembayaran.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan pada bagian keuangan mengelola pembayaran dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam pembayaran .

* + - 1. **Collaboration Diagram Mengelola Data Ruangan**

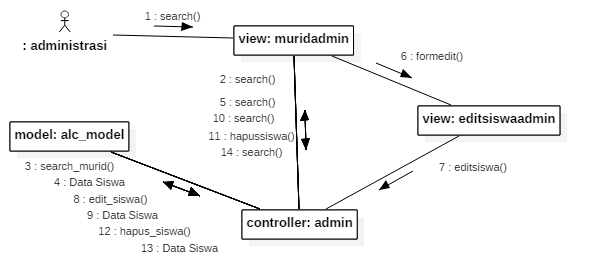
****

Gambar 3.44 Collaboration Diagram Mengelola Data Ruangan.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan pada bagian administrasi mengelola ruangan dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam ruangan.

* + - 1. **Collaboration Diagram Mengelola Data Siswa**

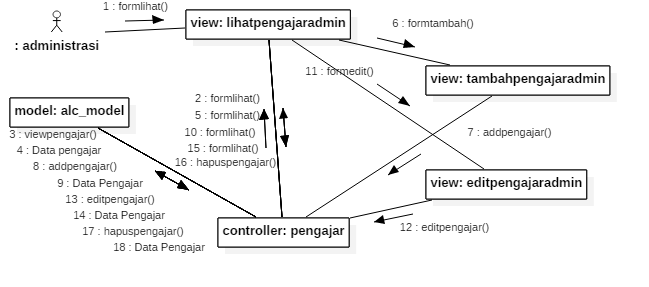
****

Gambar 3.45 Collaboration diagram Mengelola Data Siswa.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan pada bagian administrasi mengelola data siswa dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam pembayaran.

* + - 1. **Collaboration Diagram Mengelola Data Pengajar**

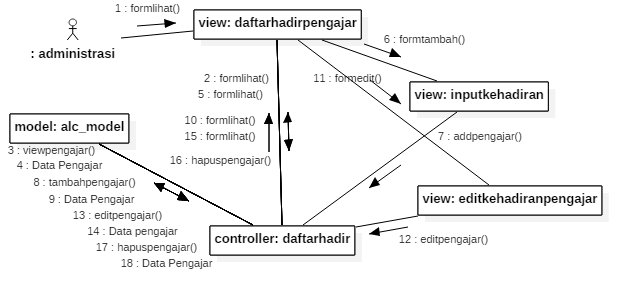
****

Gambar 3.46 Collaboration Diagram Mengelola Data Pengajar.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan pada bagian administrasi mengelola data pengajar dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam data pengajar.

* + - 1. **Collaboration Diagram Mengelola Daftar Hadir Pengajar**

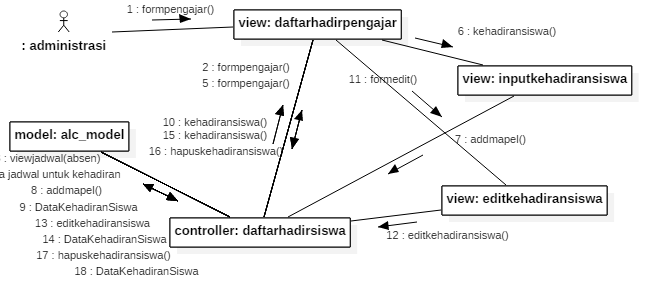
****

Gambar 3.47 Collaboration Diagram Mengelola Daftar Hadir Pengajar.

Keterangan:

Pada gambar Collaboration diagram mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam daftar hadir pengajar.

* + - 1. **Collaboration Diagram Melihat Daftar Hadir Siswa**

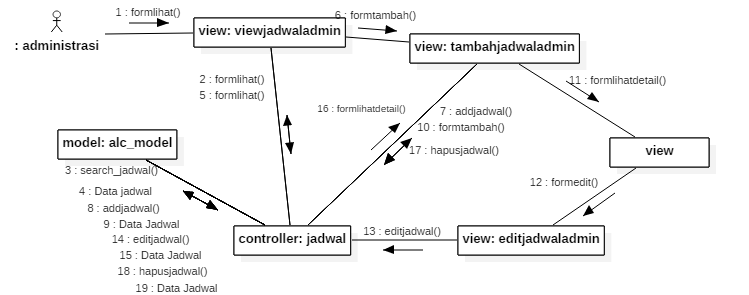
****

Gambar 3.48 Collaboration Diagram Mengelola Daftar Hadir Siswa.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan pada bagian administrasi mengelola daftar hadir siswa dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam daftar hadir siswa setiap siswa masuk kelas.

* + - 1. **Collaboration Diagram Mengelola Jadwal Belajar**

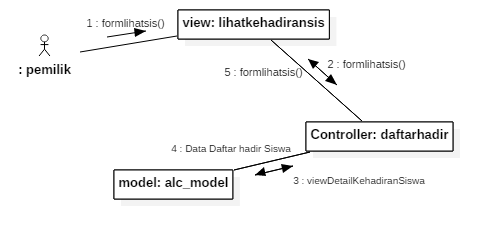
****

Gambar 3.49 Collaboration Diagram Mengelola Jadwal Belajar.

Keterangan:

Pada gambar Collaboration diagram di atas mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam jadwal belajar.

* + - 1. **Collaboration Diagram Melihat Daftar Hadir Siswa**

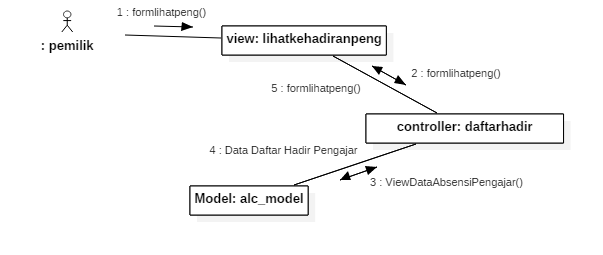
****

Gambar 3.50 Collaboration Diagram Melihat Daftar Hadir Siswa.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan bahwa pemilik dapat memantau dan melihat daftar hadir siswa pada bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Melihat Daftar Hadir Pengajar**

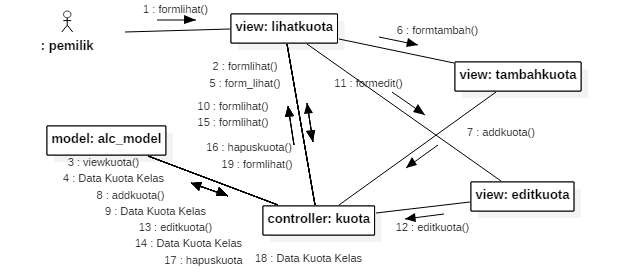
****

Gambar 3.51 Collaboration Diagram Melihat Daftar Hadir Pengajar.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan bahwa pemilik dapat memantau dan melihat daftar hadir pengajar pada bimbingan belajar ALC.

* + - 1. **Collaboration Diagram Mengelola Kuota Kelas**

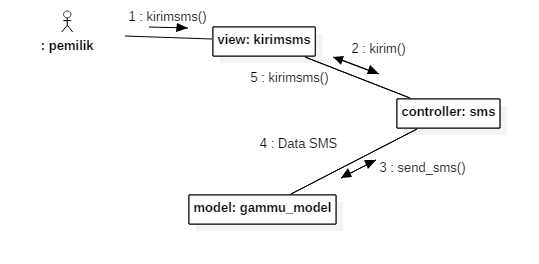
****

Gambar 3.52 Collaboration Diagram Mengelola Kuota Kelas.

Keterangan:

Pada gambar Collaboration diagram di atas menjelaskan pada bagian administrasi mengelola kuota kelas dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada.

* + - 1. **Collaboration Diagram Mengirim SMS Pemberitahuan**

****

Gambar 3.53 Collaboration Diagram Mengelola Kuota Kelas.

Keterangan:

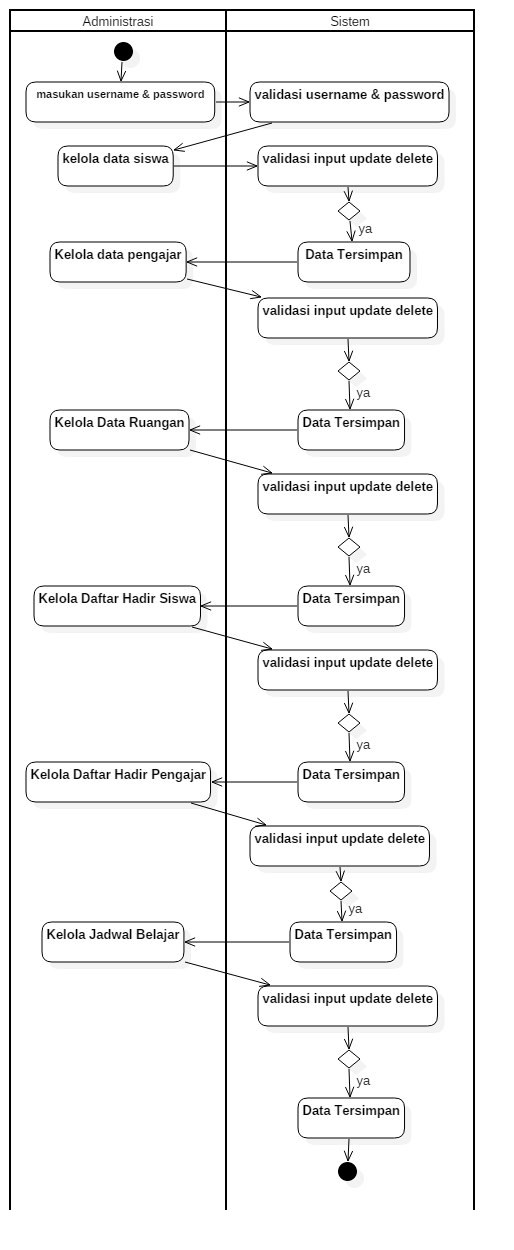
Pada gambar Collaboration diagram di atas menjelaskan pemilik dapat mengirimkan sms pemberitahuan kepada orang tua siswa bimbingan belajar ALC.

* + 1. **Activity Diagram**

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Activity diagram dibangun dari sejumlah bentuk, dihubungkan dengan panah. Jenis Bentuk yang paling penting:

* persegi panjang bulat merupakan tindakan;
* berlian merupakan keputusan;
* bar mewakili awal (split) atau akhir (bergabung) kegiatan bersamaan;
* lingkaran hitam merupakan awal (initial state) dari alur kerja;
* ingkaran hitam dikelilingi mewakili akhir (keadaan akhir).
* Panah dijalankan dari awal menuju akhir dan merupakan urutan kegiatan terjadi.
  + - 1. **Activity Diagram Bagian Administrasi**

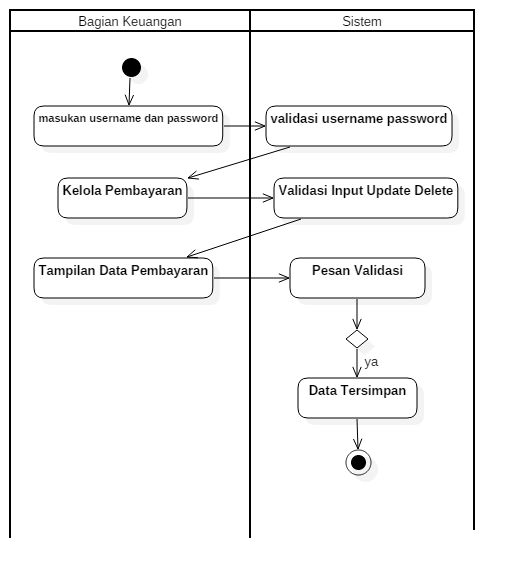
Bagian Administrasi melakukan login terlebih dahulu. Setelah itu melakukan pengelolaan berbagai macam data yang ada di Bimbingan belajar ALC yaitu mengelola data siswa, mengelola data pengajar, mengelola data ruangan, mengelola daftar hadir siswa dan pengajar serta mengelola data jadwal belajar. Dalam mengelola data siswa, bagian administrasi hanya dapat mengubah dan menghapus.

****

Gambar 3.54 Activity Diagram Bagian Administrasi.

* + - 1. **Activity Bagian Keuangan**

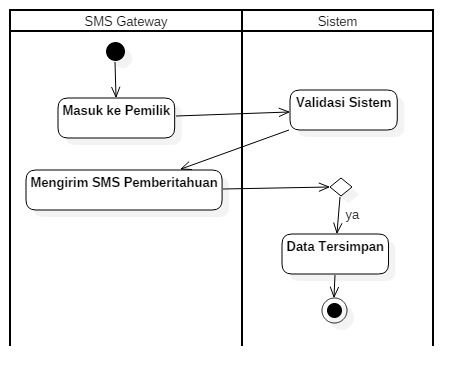
Bagian keuangan melakukan login terlebih dahulu, setelah masuk kedalam halaman utama dan memilih menu yang diinginkan yaitu paket belajar dan pembayaran. Dalam menu pembayaran bagian keuangan dapat memasukkan, mengubah, melihat dan menghapus data pembayaran.

****

Gambar 3.55 Activity Diagram Bagian Keuangan

* + - 1. **Activity Diagram SMS Gateway**

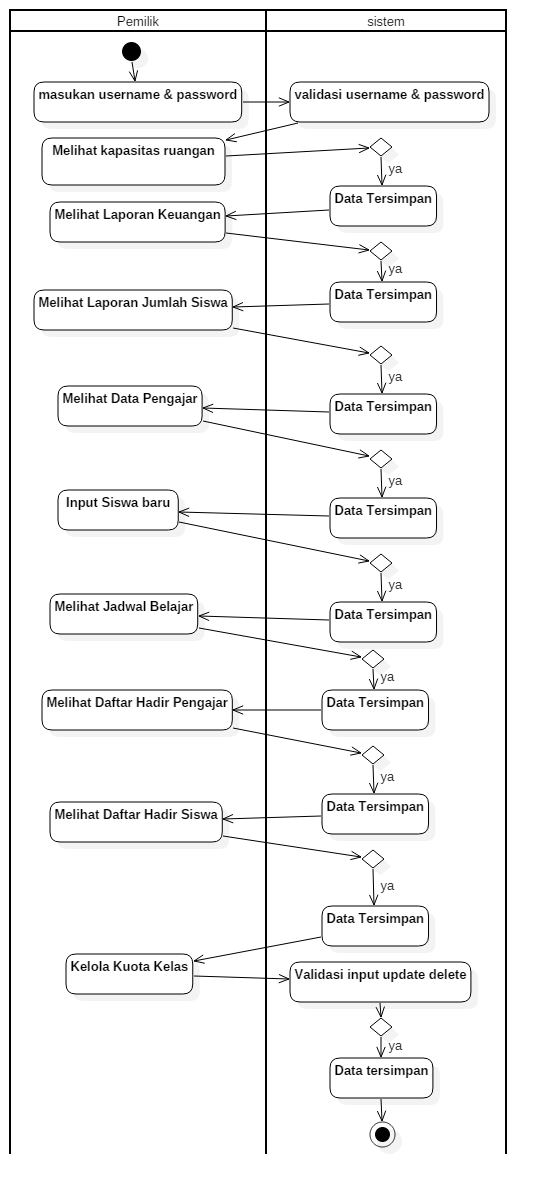
SMS Gateway harus melalui pemilik melakukan login, setelah masuk kedalam sistem informasi dan berada di halaman utama, pemilik memilih menu yang diinginkan. Pemilik dapat mengirim SMS pemberitahuan informasi tentang bimbingan belajar ALC.



Gambar 3.56 Activity Diagram SMS Gateway

* + - 1. **Activity Diagram Pemilik**

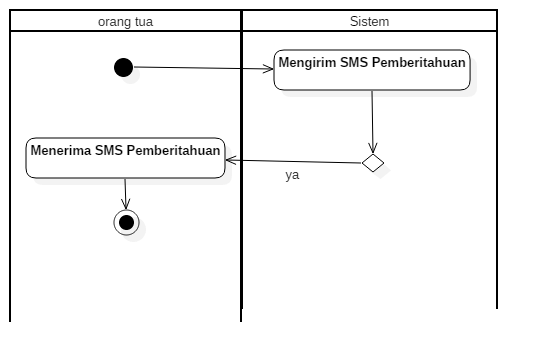
Pemilik melakukan login, setelah masuk kedalam aplikasi dan berada di halaman utama, pemilik memilih menu yang diinginkan. Pemilik hanya dapat memasukkan data siswa baru dan melihat data siswa. Pemilik tidak dapat mengubah atau menghapus data siswa tersebut. Selain itu pemilik dapat melihat data ruangan, data pengajar, laporan keuangan, laporan jumlah siswa dan melihat jadwal belajar. Untuk kelola kuota kelas, pemilik dapat memasukkan data kelas beserta kuota nya. selain itu juga pemilik dapat mengubah dan menghapus data kuota tersebut. Pemilik dapat mengirim SMS pemberitahuan informasi tentang bimbingan belajar ALC.

****

Gambar 3.57 Activity Diagram Pemilik

* + - 1. **Activity Diagram Orang Tua**

Orang Tua merupakan aktor pasif yang menerima SMS Pemberitahuan dari pemilik. Pemilik sebagai aktor untuk menjalankan SMS Gateway dan hanya menerima SMS Pemberitahuan mengenai status pembayaran dan kegiatan yang ada pada bimbingan belajar.

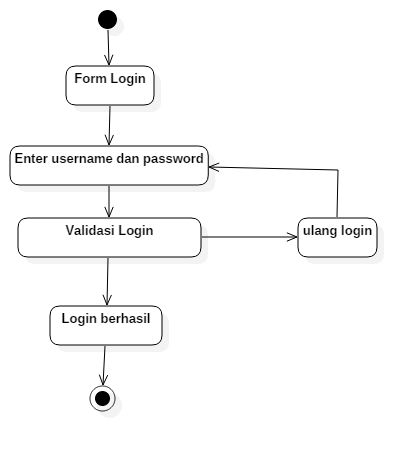


Gambar 3.58 Activity Diagram Orang Tua

* + 1. **Statechart Diagram**

State chart diagram adalah suatu diagram yang menggambarkan daur hidup (behavior pattern) dari sebuah objek, dari awal objek tsb diinisialisasi sampai di-destroy. Menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu obyek pada sistem sebagai akibat dari stimulans yang diterima.

* + - 1. **Statechart Diagram Login**

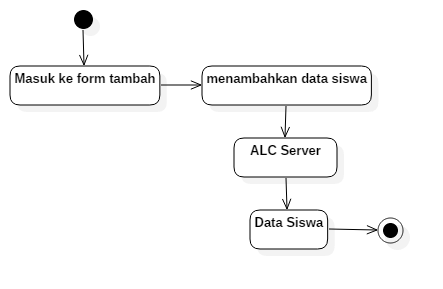
****

Gambar 3.59 Statechart Diagram Login.

Keterangan:

Pada gambar diatas terdapat Statechart diagram login yakni pemilik, bagian administrasi, dan bagian keuangan melakukan input username dan password kemudian masuk ke form login, kemudian setelah username dan password masuk ke dalam database maka system akan melakukan validasi data login kemudian masuk ke halaman menu utama. Jika gagal system akan kembali ke form login.

* + - 1. **Statechart Diagram Tambah Data Siswa Pemilik**

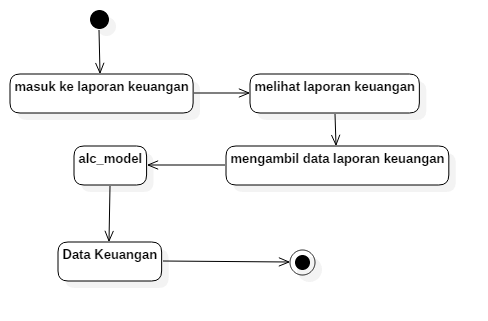
****

Gambar 3.60 Statechart Diagram Tambah Data Siswa Pemilik.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan mengenai pemilik melakukan pendaftaran terhadap murid baru atau siswa baru yang akan mendaftar bimbingan belajar.

* + - 1. **Statechart Diagram Laporan Keuangan**

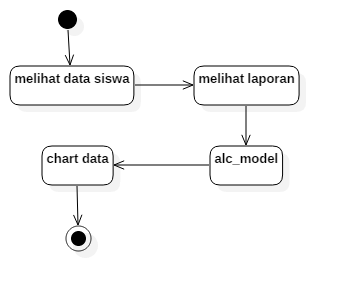
****

Gambar 3.61 Statechart Diagram Laporan Keuangan

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat melihat laporan keuangan yang dilaporkan oleh bagian keuangan.

* + - 1. **Statechart Diagram Melihat Laporan Jumlah Siswa**

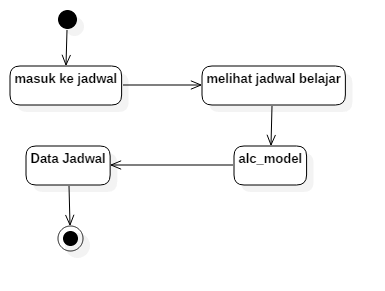
****

Gambar 3.62 Statechart Diagram Melihat Laporan Jumlah Siswa

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan proses pemilik dapat melihat hasil laporan mengenai jumlah siswa yang ada pada bimbingan belajar ALC.

* + - 1. **Statechart Diagram Melihat Jadwal Belajar**

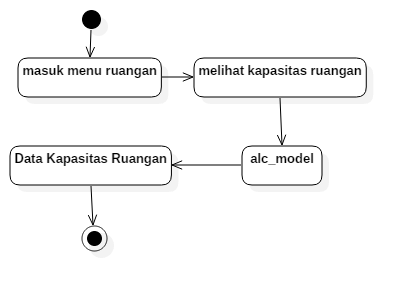
****

Gambar 3.63 Statechart Diagram Melihat Jadwal Belajar

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat melihat jadwal belajar yang sudah terjadwal di bimbingan belajar ALC.

* + - 1. **Statechart Diagram Melihat Kapasitas Ruangan**

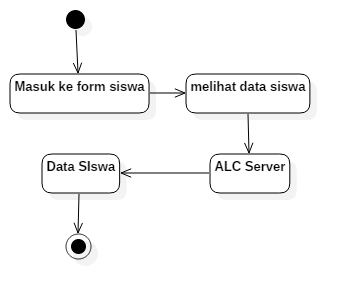
****

Gambar 3.64 Statechart Diagram Melihat Kapasitas Ruangan

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat melihat kapasitas ruangan yang digunakan untuk melakukan bimbingan belajar ALC.

* + - 1. **Statechart Diagram Melihat Data Siswa**

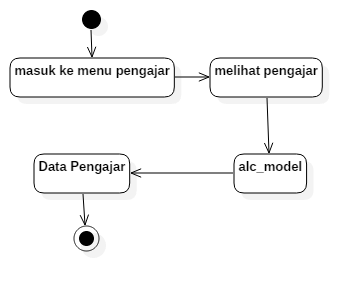
****

Gambar 3.65 Statechart Diagram Melihat Data Siswa.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat melihat data siswa pada bimbingan belajar ALC.

* + - 1. **Statechart Diagram Melihat Data Pengajar**

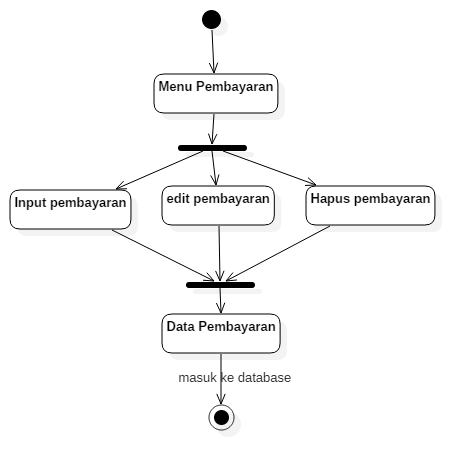
****

Gambar 3.66 Statechart Diagram Melihat Data Pengajar.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat melihat laporang tentang data dari pengajar yang ada pada bimbingan belajar ALC.

* + - 1. **Statechart Diagram Mengelola Data Pembayaran**

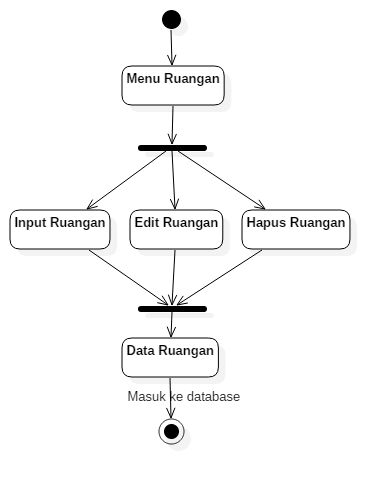
****

Gambar 3.67 Statechart Diagram Mengelola Data Pembayaran.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan pada bagian keuangan mengelola pembayaran dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam pembayaran.

* + - 1. **Statechart Diagram Mengelola Data Ruangan**

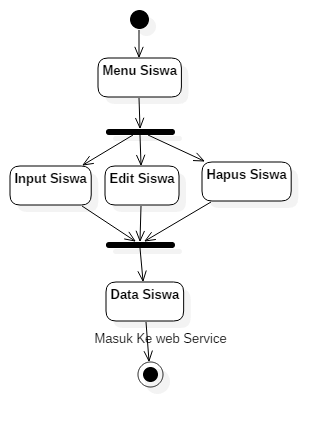
****

Gambar 3.68 Statechart Diagram Mengelola Data Ruangan

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan pada bagian administrasi mengelola ruangan dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam ruangan.

* + - 1. **Statechart Diagram Mengelola Data Siswa**

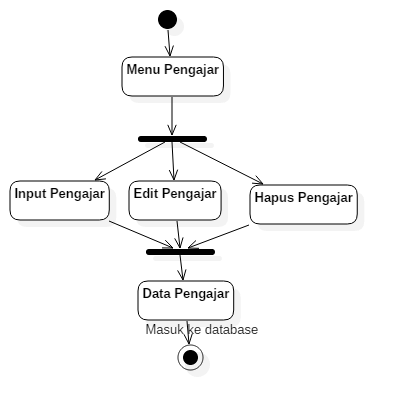
****

Gambar 3.69 Statechart Diagram Mengelola Data Siswa

Keterangan:

Pada gambar Statechart mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam pembayaran.

* + - 1. **Statechart Diagram Mengelola Data Pengajar**

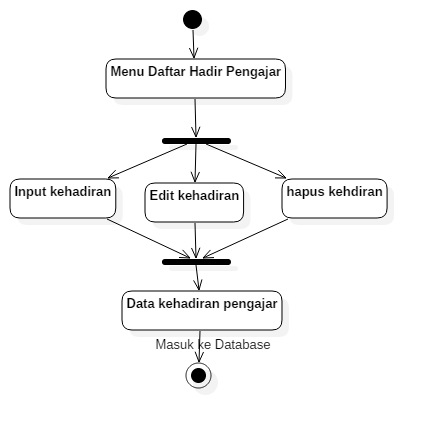
****

Gambar 3.70 Statechart Diagram Mengelola Data Pengajar

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan pada bagian administrasi mengelola data pengajar dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam data pengajar.

* + - 1. **Statechart Diagram Mengelola Daftar Hadir Pengajar**

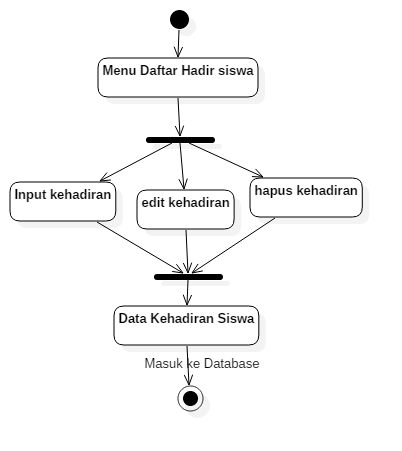
****

Gambar 3.71 Statechart Diagram Mengelola Daftar Hadir Pengajar.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan pada bagian administrasi mengelola daftar hadir pengajar dalam kelas bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam daftar hadir pengajar.

* + - 1. **Statechart Diagram Mengelola Daftar Hadir Siswa**

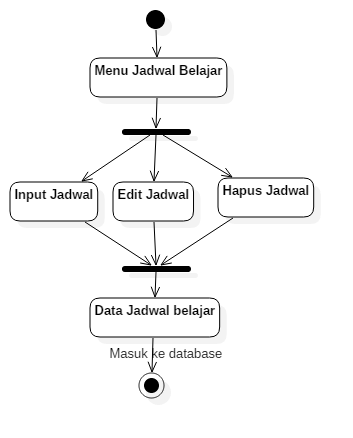
****

Gambar 3.72 Statechart Diagram Mengelola Daftar Hadir Siswa

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan pada bagian administrasi mengelola daftar hadir siswa dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam daftar hadir siswa setiap siswa masuk kelas.

* + - 1. **Statechart Diagram Mengelola Jadwal Belajar**

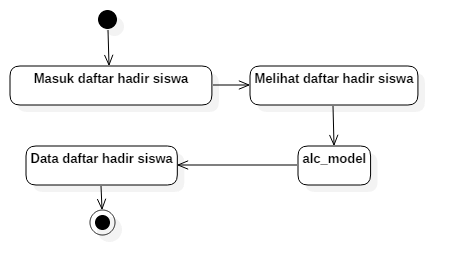
****

Gambar 3.73 Statechart Diagram Mengelola Jadwal Belajar.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan pada bagian administrasi mengelola jadwal belajar dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada dalam jadwal belajar.

* + - 1. **Statechart Diagram Melihat Daftar Hadir Siswa**

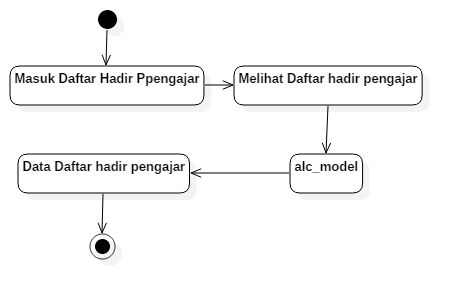
****

Gambar 3.74 Statechart Diagram Melihat Daftar Hadir Siswa.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat memantau dan melihat daftar hadir siswa pada bimbingan belajar ALC.

* + - 1. **Statechart Diagram Melihat Daftar Hadir Pengajar**

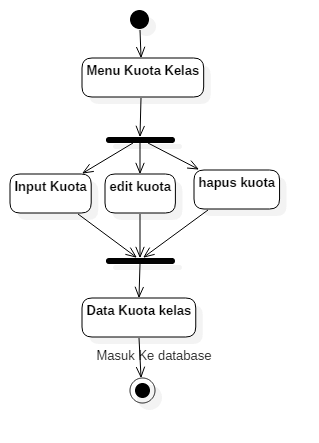
****

Gambar 3.75 Statechart Diagram Melihat Daftar Hadir Pengajar.

Keterangan:

Pada gambar Statechart diagram di atas menjelaskan bahwa pemilik dapat memantau dan melihat daftar hadir pengajar pada bimbingan belajar ALC.

* + - 1. **Statechart Diagram Mengelola Kuota Kelas**

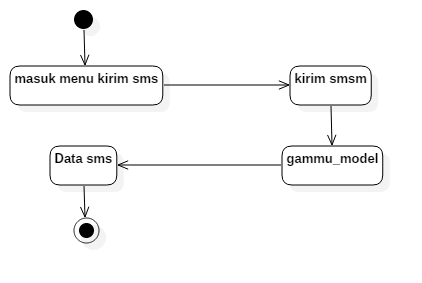
****

Gambar 3.76 Statechart Diagram Mengelola Kuota Kelas.

Keterangan:

Pada gambar Statechart diagram mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada.

* + - 1. **Statechart Diagram Mengirim SMS Pemberitahuan**

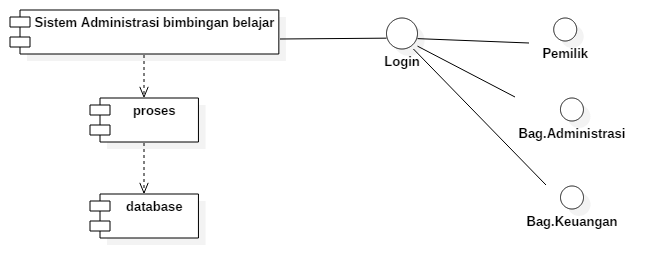
****

Gambar 3.77 Statechart Diagram Mengelola Kuota Kelas.

Keterangan:

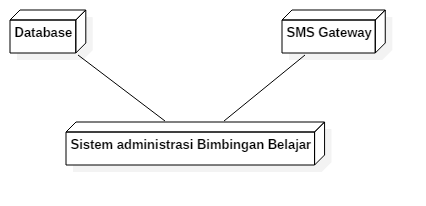
Pada gambar Statechart diagram di atas menjelaskan pada bagian administrasi mengelola kuota kelas dalam bimbingan belajar ALC. Pada diagram ini mengelola dalam memasukkan data, mengedit data, dan menghapus data yang ada.

* + 1. **Component Diagram**

****

Gambar 3.78 Component Diagram system yang akan dibangun

* + 1. **Deployment Diagram**

****

Gambar 3.79 Deployment Diagram system yang akan dibangun

* + 1. **Perancangan Basis Data**

Dalam merancang aplikasi ini, dibutuhkan sebuah database yang memiliki karakteristik scalable atau dapat diperbesar sesuai dengan kebutuhan di masa yang akan datang.

* + - 1. ***Conceptual Data Model (CDM)***

CDM digunakan untuk menggambarkan secara detail struktur basis data dalam bentuk logik. Struktur ini independen terhadap semua software maupun struktur data storage tertentu yang digunakan dalam aplikasi ini. CDM terdiri dari objek yang tidak diimplementasikan secara langsung kedalam basis data yang sesungguhnya.

Model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (entity) serta hubungan (relationship) antara entitas-entitas itu. Biasanya direpresentasikan dalam bentuk Entity Relationship Diagram.

Manfaat Penggunaan CDM dalam perancangan database :

– Memberikan gambaran yang lengkap dari struktur basis data yaitu arti, hubungan, dan batasan-batasan

– Alat komunikasi antar pemakai basis data, designer, dan analis.

* + - 1. ***Physical Data Model (PDM)***

PDM merupakan gambaran secara detail basis data dalam bentuk fisik. Penggambaran rancangan PDM memperlihatkan struktur penyimpanan data yang benar pada basis data yang digunakan sesungguhnya.

Menurut ANSI/SPARC, arsitektur basis data terbagi atas tiga level yaitu :

~ Internal/Physical Level : (yang dapat direpresentasikan dengan PDM) berhubungan dengan bagaimana data disimpan secara fisik (physical storage)

~ External /View Level : berhubungan dengan bagaimana data di representasikan dari sisi setiap user.

~ Conceptual/Logical Level : (yang dapat direpsesentasikan dengan CDM) yang menghubungkan antara internal & external level



Gambar 3.80 *Conceptual Data Model*



Gambar 3.81 *Physical Data Model*

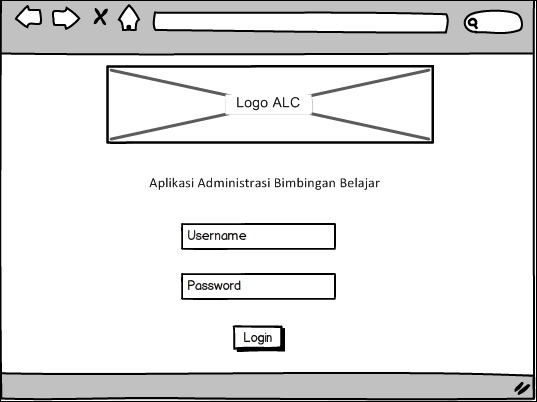
* + 1. **Struktur Menu**

Pada perancangan struktur menu akan diperlihatkan struktur menu yang akan dibuat pada system penyimpanan data, agar user yang menggunakan system penyimpanan ini lebih mudah memahami isi dari aplikasi ini.

**3.2.10.1 Form Login**

Nama Layar : Login

Deskripsi : Tampilan form untuk proses validasi masuk kemenu home.

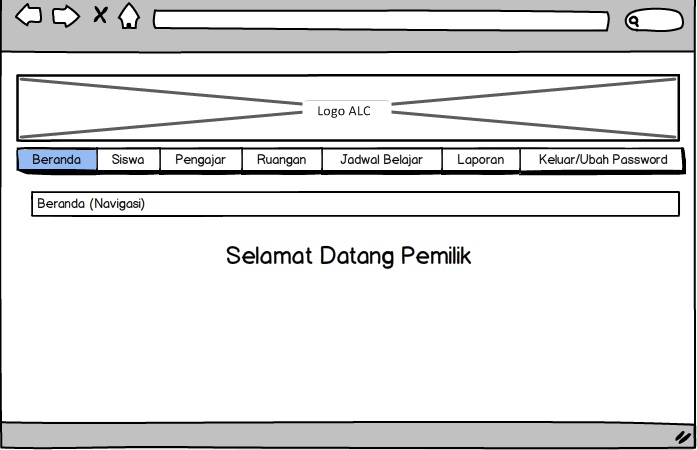


Gambar 3.82 Tampilan Antar Muka Login

Keterangan:

1. Cek apakah username dan password benar.
2. Jika salah kembali ke form login dan masukkan username dan password lagi.
3. Pada bagian header berisikan menu, logo ALC, dan motto ALC
4. Pada bagian username dan password merupakan *textbox* untuk diisi
5. Untuk tombol login digunakan bentuk pilihan *button*.
   * + 1. **Tampilan Bagian Pemilik**

Perancangan interface yaitu pada proses bagian pemilik.

****

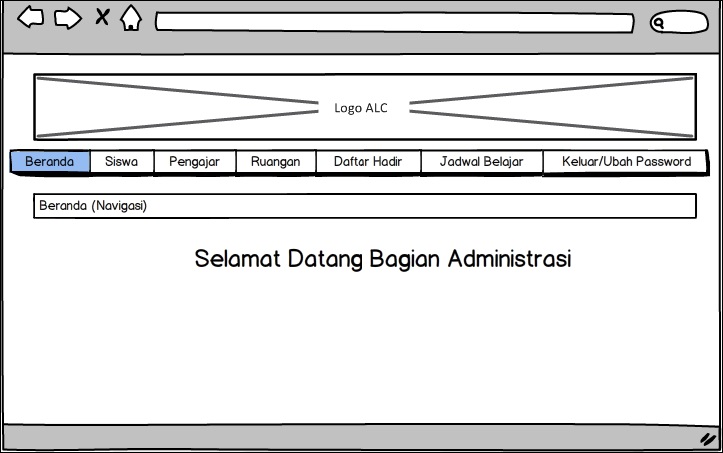
Gambar 3.83 Tampilan Antar Muka ProsesBagian Pemilik

Keterangan:

Gambar 3.83 merupakan menu untuk bagian pemilik yang memilik sub menu yaitu

1. Beranda.
2. Siswa yang didalamnya terdapat tambah siswa,edit siswa, dan hapus siswa.
3. Pengajar untuk melihat pengajar.
4. Ruangan untuk melihat ruangan.
5. Jadwal belajar untuk melihat jadwal belajar.
6. Laporan terdiri dari laporan jumlah murid dan laporan keuangan.
7. Kehadiran terdiri dari kehadiran siswa, pengajar, dan permapel
8. Kuota kelas.
9. Sms.
   * + 1. **Tampilan Menu Bagian Administrasi**

Perancangan interface yaitu pada menu bagian administrasi.



Gambar 3.84 Tampilan Antar Muka Menu Bagian Administrasi

Keterangan:

Gambar 3.84 merupakan rancangan antar muka menu bagian administrasi. Dimana pada antar muka ini bagian administrasi dapat melakukan hal-hal seperti:

1. Beranda.
2. Siswa untuk melihat data siswa.
3. Pengajar terdapat submenu untuk tambah, edit, dan hapus.
4. Ruangan terdapat submenu untuk tambah, edit, dan hapus.
5. Daftar hadir terdapat submenu tambah daftar hadir dan lihat daftar hadir.
6. Jadwal belajar terdapat menu tambah, edit, dan hapus.
   * + 1. **Tampilan Menu Bagian Keuangan**

Perancangan interface yaitu pada menu bagian keuangan.

****

Gambar 3.85 Tampilan Antar Muka Menu Bagian Keuangan

Keterangan:

Gambar 3.85 merupakan rancangan antar muka menu pada bagian keuangan yang ada dalam bimbingan belajar ALC. Menu yang ada di dalamnya berupa transaksi yang dilakukan oleh siswa bimbingan belajar yang terdiri dari paket belajar dan pembayaran.