# BAB III
# PEMBAHASAN

## 3.1 Analisis Sistem

Analisis sistem (system analysis) dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalah-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat memahami sistem pemikiran yang kompleks dengan memecahnya ke dalam unsur-unsur yang lebih sederhana sehingga hubungan antar unsur-unsur itu menjadi jelas. Dalam menganalisis sistem secara O*bject Oriented* yang nantinya di bangun aplikasi berorientasi objek, sehingga hubungan antar unsur-unsur itu menjadi jelas.

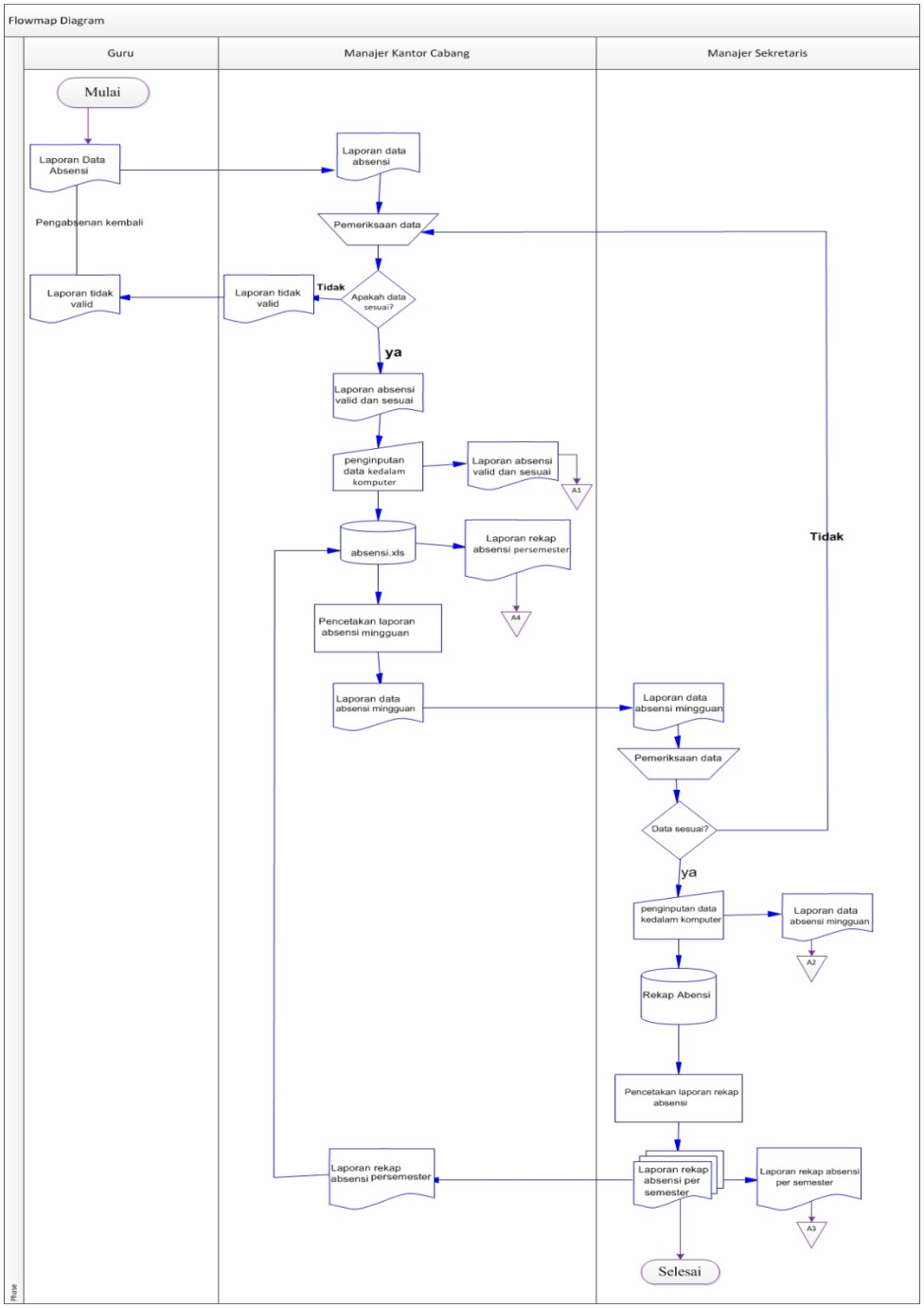
Berikut adalah hasil analisis dari sistem manual yang ada dengan menggunakan metode UML (*Unified Modeling Language*).

### 3.1.1. Analisis Sistem yang Berjalan

Dalam menangani data absensi siswa, Science Group Indonesia menggunakan proses sebagai berikut:

1. Guru melakukan absensi terhadap siswa kelas, serta memberikan laporan data kepada manajer cabang.
2. Manajer cabang menerima laporan absensi siswa, dan melakukan pemeriksaan data absensi.
3. Jika laporan sesuai dengan data yang tersimpan di absensi.xls maka Laporan tersebut akan diinputkan kedalam komputer dan disimpan dalam file absensi(.xls), dan dokumennya akan disimpan sebagai arsip (**A1**)
4. Jika laporan yang diperiksa tidak sesuai maka data tersebut akan dikembalikan kepada guru untuk diperiksa kembali.

5. Setelah data laporan yang valid telah disimpan, maka data harian tersebut akan diakumulasi sampai sebanyak 6 hari dari hari senin-sabtu (data laporan mingguan) .
6. Data tersebut akan dicetak sebanyak 2(dua) rangkap untuk dibawa ke kantor pusat untuk menjadi laporan mingguan, sedangkan laporan yang lain disimpan menjadi arsip di kantor cabang.
7. Manajer cabang akan membawa dokumen laporan absensi mingguan ke kantor pusat.
8. Manajer sekretaris (Kantor pusat) akan menerima dokumen laporan mingguan tersebut.
9. Manajer sekretaris akan melakukan pemeriksaan dokumen laporan tersebut dengan data yang terdapat dalam rekap absensi.xls.
10. Jika data yang diperiksa tidak sesuai dengan data yang tersimpan di kantor pusat, maka dokumen akan disimpan dikembalikan ke manajer cabang untuk diperiksa kembali.
11. Jika data yang diperiksa sesuai dengan data yang tersimpan, maka data yang terdapat dalam dokumen akan diinputkan melalui komputer kedalam file rekap absensi.xls.
12. Manajer sekretaris (Kantor Pusat) akan mencetak data rekap laporan absensi persemester sebanyak 2 (dua) rangkap, yaitu: 1 rangkap untuk kantor cabang dan 1 rangkap untuk kantor pusat yang akan disimpan sebagai arsip.
13. Manajer cabang akan menerima data rekap laporan absensi persemester.




**Gambar 3.1 Flowmap diagram**

Keterangan:

1. A1 : Arsip laporan absensi harian kantor cabang.
2. A2 : Arsip laporan absensi mingguan kantor pusat
3. A3 : Arsip laporan rekap absensi per semester kantor pusat.
4. A4 : Arsip laporan rekap absensi per semester kantor cabang.

### 3.1.2. Evaluasi Sistem Yang Sedang Berjalan.

Berdasarkan analisis serta pengumpulan data penelitian (Observasi dan wawancara) maka dapat dievaluasi beberapa hal seperti berikut:

**Tabel 3.1 Tabel Evaluasi Sistem yang berjalan.**

| No. | Permasalahan |
|---|---|
| 1 | Duplikasi data absensi membuat proses rekap data menjadi lebih lama, karena harus menyesuaikan data rekap yang ada di kantor pusat dengan kantor cabang. |
| 2 | Proses pemutakhiran data serta pengolahan data menjadi lebih lama dikarenakan petugas dapat memastikan data yang tersimpan di kantor pusat itu sesuai dengan kantor cabang. |
| 3 | Setiap kantor cabang tidak memiliki data absensi kelas yang sama, dikarenakan kelas bimbingan yang tersebar di kantor cabang berbeda. |
| 4 | Pelaporan serta rekap jumlah kehadiran siswa kepada orang tua siswa menjadi lebih lama, dikarenakan harus menunggu data dari kantor pusat. |

### 3.1.3. Sistem yang Akan Dibangun.

Berikut adalah fitur-fitur yang diharapkan dimiliki oleh Aplikasi Absensi Online Home-Schooling.

**Tabel 3.2 Tabel fitur sistem absensi online**

| **No.** | **Nama Fitur** | **Penjelasan** |
|---|---|---|
| 1. | View Absen. | Menampilkan daftar absensi perkelas secara terurut |
| 2. | Pengolahan data Absensi. | Dapat melakukan Tambah, Hapus, dan *Edit* data Absensi siswa per kelas per hari. |

| | | |
|---|---|---|
| 3. | Pengolahan data Kelas. | Dapat menampilkan data kelas serta melakukan penambahan, penghapusan serta pengubahan data kelas. |
| 3. | Pengolahan data Murid | Dapat menampilkan data murid serta melakukan operasi tambah, hapus, *edit* data murid tersebut. |
| 4. | Pemilihan semester. | Dapat memilih semester yang aktif. |
| 5. | Perekapan data. | Dapat menghasilkan data absensi perkelas persemester dalam format .xls |
| 6. | Login. | Sistem dapat melakukan proteksi terhadap data yang disimpan, dengan hanya mengijinkan user tertentu saja yang dapat amsuk kedalam sistem. |

### 3.1.4. Analisis Pengguna Sistem

Berikut adalah hak akses yang dimiliki oleh user, yaitu admin.

**Tabel 3.3 Tabel Analisis Pengguna Sistem.**

| Pengguna | Hak Akses | Tingkat Keterampilan |
|---|---|---|
| Admin | Dapat melakukan *login* kedalam sistem | Dapat mengoperasikan aplikasi berbasis web pada OS Windows. |
| | Menampilkan data Absensi, Murid, Kelas serta Pilihan semester. | |
| | Dapat melakukan Tambah, Hapus serta Edit data Absensi, Murid, Kelas dan Pilihan semester. | |
| | Dapat memilih data semester (ganjil atau genap) yang aktif. | |
| | Dapat melakukan rekap data absensi perkelas per semester. | |

### 3.1.5. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang di perlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

#### 3.1.5.1. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan pada suatu instansi atau perusahaan yang digunakan sebagai media untuk menerapkan sistem sangat berpengaruh pada efisiensi dan efektivitas kinerja suatu sistem tersebut, adapun pada sistem yang berjalan, hardware yang digunakan dengan spesifikasi sebagai berikut :

1. Processor Pentium Dual Core 2.66 GHz
2. Memory (RAM) 1 GB
3. Harddisk 160 GB
4. DVD Room 52x
5. Monitor 17’’
6. Mouse
7. Keyboard
8. Printer

Perangkat keras komputer yang dibutuhkan untuk menjalankan aplikasi Pendataan Absensi Online Siswa Berbasis Web dengan menggunakan framework Codeigniter yang akan dibangun minimal memiliki :

1. Processor Pentium IV IGHz.
2. Memory (RAM) 64 MB
3. Harddisk 10 GB
4. CD Room 52x
5. Monitor 17’’

6. Mouse
7. Keyboard
8. Printer

Perangkat Keras komputer yang ada saat ini sudah memenuhi kebutuhan dalam menjalankan aplikasi pendataan absensi online siswa berbasis web dengan menggunakan framework codeigniter yang dibangun.

#### 3.1.5.2. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan oleh suatu sistem informasi belum dapat digunakan secara optimal, apabila software itu tidak mampu secara khusus melayani kebutuhan sistem. Sistem operasi yang digunakan pada tempat tersebut yaitu: Microsoft Windows XP Professional sehingga sudah sangat mencukupi untuk dapat menjalankan perangkat lunak yang akan dibangun. Perangkat lunak yang dibutuhkan dalam pembangunan aplikasi ini adalah :

1. WampServer sebagai Web Server.
2. PHP dan framework CodeIgniter.
3. Adobe Dreamweaver CS (Creative Suite) 6 sebagai editor PHP.
4. MySQL sebagai DBMS.
5. Tool antarmuka design Microsoft Visio 2007.
6. Mozila Firefox sebagai browser.

## 3.2. Analisis Sistem Aplikasi Absensi Online

Pada tahap ini dilakukan analisis terhadap sistem aplikasi yang di bangun berdasarkan pada hasil analisis tersebut maka di tuangkan dalam bentuk pemodelan aplikasi berbasis objek yang di gambarkan dalam UML.

### 3.2.1. Use Case Diagram

Use Case Diagram di bawah ini menggambarkan fungsionalitas dari sistem informasi dari aplikasi yang di bangun. Use case ini menggambarkan interaksi antara aktor dengan sistem. Dalam gambar 3.1 di bawah ini menggambarkan

hubungan antara aktor yang di wakili oleh Admin dan Sistem yang terdiri dari beberapa proses-proses yang saling berhubungan.




**Gambar 3.2 Use Case Diagram**

#### 3.2.1.1 Definisi Use Case

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap Use Case. Definisinya dapat dilihat pada tabel berikut.

**Tabel 3.4 Definisi Use Case**

| No | *Use Case* | Deskripsi |
|---|---|---|
| 1. | Login | Proses verifikasi *Username* dan *Password* sebelum masuk kedalam sistem. |
| 2. | Penambahan Absensi | Proses untuk melakukan penambahan data absen dan menampilkannya dalam bentuk tabel. |
| 3. | Update Absensi | Proses yang dilakukan untuk mengubah data absensi yang terdapat dalam database. |
| 4. | Delete Absensi | Proses yang dilakukan untuk menghapus data absensi yang terdapat dalam database. |
| 5. | Perekapan Data | Proses pengumpulan data absensi per kelas dalam satu semester. |
| 6. | Download Rekap | Proses yang dilakukan untuk mengunduh *file* rekap dalam format .xls. |
| 7. | Pemilihan Semester | Proses untuk mengaktifkan semester yang diinginkan (semester ganjil [1] atau genap [2]) |
| 8. | Penambahan Siswa | Proses untuk melakukan penambahan data siswa dan menampilkannya dalam bentuk tabel. |
| 9. | Update Siswa | Proses yang dilakukan untuk mengubah data siswa yang terdapat dalam database. |

| | | |
|---|---|---|
| 10. | Delete Siswa | Proses yang dilakukan untuk menghapus data siswa yang diinginkan. |
| 11. | Penambahan Kelas | Proses untuk menambahkan data kelas, sertaa menampilkannya dalam bentuk tabel. |
| 12. | Update Kelas | Proses yang dilakukan untuk mengubah data kelas yang terdapat dalam database. |
| 13. | Delete Kelas | Proses yang dilakukan untuk menghapus data kelas yang terdapat dalam database. |

#### 3.2.1.2 Skenario Use Case

Skenario setiap bagian pada use case menunjukkan proses apa yang terjadi pada setiap bagian didalam use case tersebut, dimana user memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada user setelah user memberikan perintah pada setiap bagian – bagian use case.

1. Skenario Use Case Login

**Tabel 3.5 Skenario Use Case Login**

| Identifikasi | |
|---|---|
| Nama | Login |
| Tujuan | Melakukan Login ke dalam aplikasi. |
| Deskripsi | Proses verifikasi *username* dan *password* sebelum masuk kedalam sistem. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk ke halaman login aplikasi | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memasukkan *username* dan *password* kedalam *form login* | 2. Menerima permintaan login |
| | 3. Memeriksa dalam *database* dan melakukan verifikasi *username* dan *password.* |
| | 4. Jika proses verifikasi berhasil, maka sistem akan menampilkan menu |

| | |
|---|---|
| | utama aplikasi |
| 5. Menerima tampilan utama aplikasi. | |
| Kondisi akhir : Admin telah masuk kedalam halaman utama aplikasi. | |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| | 1. Menampilkan pesan username dan password tidak benar. |
| 2. Mengisi kembali *username* dan *password* | |
| | 3. Mengautentifikasi data *username* dan *password* maka sistem akan menampilkan menu utama aplikasi |
| Kondisi akhir : Admin telah masuk kedalam halaman utama aplikasi. | |

2. Skenario Use Case Penambahan Absensi.

**Tabel 3.6 Skenario Use Case Penambahan Absensi**

| Identifikasi | |
|---|---|
| Nama | Penambahan Absensi |
| Tujuan | Menambahkan data absensi kedalam database |
| Deskripsi | Menambahkan data absensi ke dalam database dan menampilkan data absensi terbaru kedalam tabel. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu absen. | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memasukkan NIS,Tanggal dan memilih Absen [Ijin, Sakit, Alpha, hadir]. | 2. Menerima permintaan penambahan data. |
| | 3. Memeriksa data dalam database. |
| | 4. Menyimpan data dalam database. |
| | 5. Mengambil data absen terbaru dari dalam database dan menyajikan data tersebut dalam |

| | |
|---|---|
| | format tabel. |
| | 6. Memberikan pesan data telah tersimpan dan tabel data absen yang terbaru. |
| 7. Menerima pesan penyimpanan dan tabel data absensi yang terbaru. | |
| Kondisi akhir : Admin berhasil menyimpan data , menerima pesan penyimpanan dan tabel data absensi yang terbaru. | |

3. Skenario Use Case Update Absensi.

**Tabel 3.7 Skenario Use Case Update Absensi**

| Identifikasi | |
|---|---|
| Nama | Update Absensi. |
| Tujuan | Mengubah data absensi yang ada di dalam database |
| Deskripsi | Proses untuk mengubah data absensi yang ada dalam database dan menampilkan data absensi terbaru dalam bentuk tabel. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu siswa. | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih data yang akan diubah | 2. Menerima permintaan *update* data. |
| | 3. Menyiapkan form *update*. |
| 4. Mengisi form *update*. | 5. Meng-*update* data dalam database. |
| | 6. Menyusun data absensi dalam bentuk tabel |
| 7. Menampilkan data absensi yang telah disimpan dalam bentuk tabel | |
| Kondisi akhir : Admin berhasil mengubah data absensi dan menampilkan data absensi yang terbaru dalam bentuk tabel. | |

4. Skenario Use Case Delete Absensi.

**Tabel 3.8 Use Case Delete Absensi.**

<table>
<tr><td colspan="2">Identifikasi</td></tr>
<tr><td>Nama</td><td>Delete Absensi.</td></tr>
<tr><td>Tujuan</td><td>Menghapus data absensi yang ada di dalam database.</td></tr>
<tr><td>Deskripsi</td><td>Proses untuk menghapus data absensi yang ada dalam database dan menampilkan data absensi terbaru dalam bentuk tabel.</td></tr>
<tr><td>Aktor</td><td>Admin</td></tr>
<tr><td colspan="2">Skenario Utama</td></tr>
<tr><td colspan="2">Kondisi awal : Admin telah masuk kedalam menu siswa.</td></tr>
<tr><td>Aksi Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Memilih data yang akan dihapus.</td><td>2. Menerima permintaan delete data.</td></tr>
<tr><td></td><td>3. Menyiapkan form konfirmasi delete.</td></tr>
<tr><td>4. Mengonfirmasi perintah delete dalam form .</td><td>5. Menghapus data dalam database.</td></tr>
<tr><td></td><td>6. Menyusun data kelas dalam bentuk tabel</td></tr>
<tr><td>7. Menampilkan data absensi yang telah disimpan dalam bentuk tabel</td><td></td></tr>
<tr><td colspan="2">Kondisi akhir : Admin berhasil menghapus data absensi dan menampilkan data absensi yang terbaru dalam bentuk tabel.</td></tr>
</table>

5. Skenario Use Case Perekapan Data.

**Tabel 3.9 Skenario Use Case Perekapan Data**

<table>
<tr><td colspan="2">Identifikasi</td></tr>
<tr><td>Nama</td><td>Perekapan Data</td></tr>
<tr><td>Tujuan</td><td>Menampilkan rekap data absensi berdasarkan kelas dan semester dalam bentuk tabel.</td></tr>
<tr><td>Deskripsi</td><td>Proses untuk menampilkan rekap data absensi per kelas terbaru yang terdapat dalam database</td></tr>
<tr><td>Aktor</td><td>Admin</td></tr>
<tr><td colspan="2">Skenario Utama</td></tr>
<tr><td colspan="2">Kondisi awal : Admin telah masuk kedalam menu rekap dan memilih kelas yang akan direkap datanya.</td></tr>
<tr><td>Aksi Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Memilih kelas yang diinginkan dan melakukan permintaan rekap.</td><td>2. Menerima permintaan rekap data.</td></tr>
<tr><td></td><td>3. Memeriksa data dalam database berdasarkan id_kelas dan id_semester.</td></tr>
<tr><td></td><td>4. Menyusun rekap data tersebut dalam bentuk tabel</td></tr>
<tr><td>5. Menerima rekap data absensi dan menampilkan data tersebut dalam bentuk tabel.</td><td></td></tr>
<tr><td colspan="2">Kondisi akhir : Admin berhasil menerima rekap data absensi perkelas dalam bentuk tabel.</td></tr>
</table>

6. Skenario Use Case Download Rekap.

**Tabel 3.10 Skenario Use Case Download Rekap.**

| Identifikasi | |
|---|---|
| Nama | Download Rekap. |
| Tujuan | Mengunduh rekap data absen dari database. |
| Deskripsi | Proses untuk mengunduh rekap data absen per kelas dalam satu semester kedalam *file* format .xls. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu rekap. | |
| Aksi Aktor | Reaksi Sistem |
| 1. Mengirim permintaan download | 2. Menerima permintaan *download* data. |
| | 3. Menyiapkan data yang akan di *download* dari *database* |
| | 4. Menyusun data yang telah dikumpulkan kedalam *file* berformat .xls. |
| 5. Mengunduh *file* rekap. | |
| Kondisi akhir : Admin berhasil mengunduh rekap data absensi dalam *file* berformat .xls | |

7. Skenario Use Case Pemilihan Semester

**Tabel 3.11 Skenario Use Case Pemilihan Semester**

| Identifikasi | |
|---|---|
| Nama | Pemilihan Semester |
| Tujuan | Mengubah status semester yang aktif dalam database, sehingga pengolahan data hanya dapat dilakukan untuk semester tersebut. |
| Deskripsi | Proses untuk mengubah status semester yang aktif dalam database. |
| Aktor | Admin |
| Skenario Utama | |

| Kondisi awal : Admin telah masuk kedalam menu semester. | |
|---|---|
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih semester yang akan diaktifkan. | 2. Menerima permintaan pengaktifan semester. |
| | 3. Mengaktifkan semester tersebut, dan menonaktifkan semester yang lain. |
| | 4. Menyimpan status semester yang aktif kedalam database. |
| | 5. Mengubah tampilan status semester yang aktif dalam antar muka aplikasi |
| 6. Menerima konfirmasi status semester yang aktif melalui tampilan antar muka menu semester dalam aplikasi | |
| Kondisi akhir : Admin berhasil mengubah status semester yang aktif. | |

8. Skenario Use Case Penambahan Siswa.

**Tabel 3.12 Skenario Use Case Penambahan Siswa.**

| Identifikasi | |
|---|---|
| Nama | Penambahan Siswa |
| Tujuan | Menambahkan data siswa dalam database. |
| Deskripsi | Proses untuk menambahkan data siswa dan menampilkan data tersebut dari database dalam bentuk tabel. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu siswa. | |
| Aksi Aktor | Reaksi Sistem |
| 1. *Input* NIS, Nama dan memilih Kelas | 2. Menerima permintaan input data. |
| | 3. Melakukan pengujian terhadap data yang duplikat, dengan mencari data (NIS) yang akan diinput |

| | |
|---|---|
| | dalam database. |
| | 4. Mengecek ketersediaan data kelas. |
| | 5. Jika NIS belum disimpan, dan Kelas tersedia. Sistem menyimpan data NIS, Nama dan Kelas kedalam *database*. |
| | 6. Menyiapkan data siswa yang telah disimpan, serta pesan konfirmasi penyimpanan |
| 7. Menerima konfirmasi penyimpanan data dan menampilkan data siswa dalam bentuk tabel. | |
| Kondisi akhir : Admin berhasil menyimpan data siswa dan menampilkan data siswa yang disimpan dalam bentuk tabel. | |

9. Skenario Use Case Update Siswa.

**Tabel 3.13 Skenario Use Case Update Siswa.**

| Identifikasi | |
|---|---|
| Nama | Update Siswa |
| Tujuan | Mengubah data siswa yang ada di dalam database |
| Deskripsi | Proses untuk mengubah data siswa yang ada dalam database dan menampilkan data siswa terbaru dalam bentuk tabel. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu siswa. | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih data yang akan diubah | 2. Menerima permintaan *update* data. |
| | 3. Menyiapkan form *update*. |
| 4. Mengisi form *update*. | 5. Meng-*update* data dalam database. |

| | |
|---|---|
| | 6. Menyusun data siswa dalam bentuk tabel |
| 7. Menampilkan data siswa yang telah disimpan dalam bentuk tabel | |
| Kondisi akhir : Admin berhasil mengubah data siswa dan menampilkan data siswa yang terbaru dalam bentuk tabel. | |

10. Skenario Use Case Delete Siswa.

**Tabel 3.14 Skenario Use Case Delete Siswa.**

| Identifikasi | |
|---|---|
| Nama | Delete Siswa |
| Tujuan | Menghapus data siswa yang ada di dalam database |
| Deskripsi | Proses untuk menghapus data siswa yang ada dalam database dan menampilkan data siswa terbaru dalam bentuk tabel. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu siswa. | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih data yang akan dihapus. | 2. Menerima permintaan *delete* data. |
| | 3. Menyiapkan form konfirmasi *delete*. |
| 4. Mengonfirmasi perintah *delete* dalam form . | 5. Menghapus data dalam database. |
| | 6. Menyusun data siswa dalam bentuk tabel |
| 7. Menampilkan data siswa yang telah disimpan dalam bentuk tabel | |
| Kondisi akhir : Admin berhasil menghapus data siswa dan menampilkan data siswa yang terbaru dalam bentuk tabel. | |

11. Skenario Use Case Penambahan Kelas.

**Tabel 3.15 Skenario Use Case Penambahan Kelas.**

<table>
<tr><td colspan="2">Identifikasi</td></tr>
<tr><td>Nama</td><td>Penambahan Kelas</td></tr>
<tr><td>Tujuan</td><td>Menambahkan data kelas dalam database dan menampilkan data kelas terbaru.</td></tr>
<tr><td>Deskripsi</td><td>Proses untuk menambahkan data kelas dan menampilkan data tersebut dari database dalam bentuk tabel.</td></tr>
<tr><td>Aktor</td><td>Admin</td></tr>
<tr><td colspan="2">Skenario Utama</td></tr>
<tr><td colspan="2">Kondisi awal : Admin telah masuk kedalam menu kelas.</td></tr>
<tr><td>Aksi Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Input id_kelas, dan Nama kelas.</td><td>2. Menerima permintaan input data.</td></tr>
<tr><td></td><td>3. Melakukan pengujian terhadap data yang duplikat, dengan mencari data (id_kelas) yang akan diinput dalam database.</td></tr>
<tr><td></td><td>4. Jika id kelas belum disimpan, sistem menyimpan data kelas kedalam database.</td></tr>
<tr><td></td><td>5. Menyiapkan data kelas yang telah disimpan, serta pesan konfirmasi penyimpanan</td></tr>
<tr><td>6. Menampilkan data kelas yang telah disimpan dalam bentuk tabel</td><td></td></tr>
<tr><td colspan="2">Kondisi akhir : Admin berhasil menambahkan data kelas dan menampilkan data kelas yang terbaru dalam bentuk tabel.</td></tr>
</table>

12. Skenario Use Case Update Kelas.

**Tabel 3.16 Skenario Use Case Update Kelas.**

<table>
<tr><td colspan="3">Identifikasi</td></tr>
<tr><td>Nama</td><td colspan="2">Update Kelas</td></tr>
<tr><td>Tujuan</td><td colspan="2">Mengubah data kelas yang ada di dalam database</td></tr>
<tr><td>Deskripsi</td><td colspan="2">Proses untuk mengubah data kelas yang ada dalam database dan menampilkan data kelas terbaru dalam bentuk tabel.</td></tr>
<tr><td>Aktor</td><td colspan="2">Admin</td></tr>
<tr><td colspan="3">Skenario Utama</td></tr>
<tr><td colspan="3">Kondisi awal : Admin telah masuk kedalam menu kelas.</td></tr>
<tr><td>Aksi Aktor</td><td colspan="2">Reaksi Sistem</td></tr>
<tr><td>1. Memilih data yang akan diubah</td><td colspan="2">2. Menerima permintaan update data.</td></tr>
<tr><td></td><td colspan="2">3. Menyiapkan form update.</td></tr>
<tr><td>4. Mengisi form update.</td><td colspan="2">5. Meng-update data dalam database.</td></tr>
<tr><td></td><td colspan="2">6. Menyusun data kelas dalam bentuk tabel</td></tr>
<tr><td>7. Menampilkan data kelas yang telah disimpan dalam bentuk tabel</td><td colspan="2"></td></tr>
<tr><td colspan="3">Kondisi akhir : Admin berhasil mengubah data siswa dan menampilkan data siswa yang terbaru dalam bentuk tabel.</td></tr>
</table>

13. Skenario Use Case Delete Kelas.

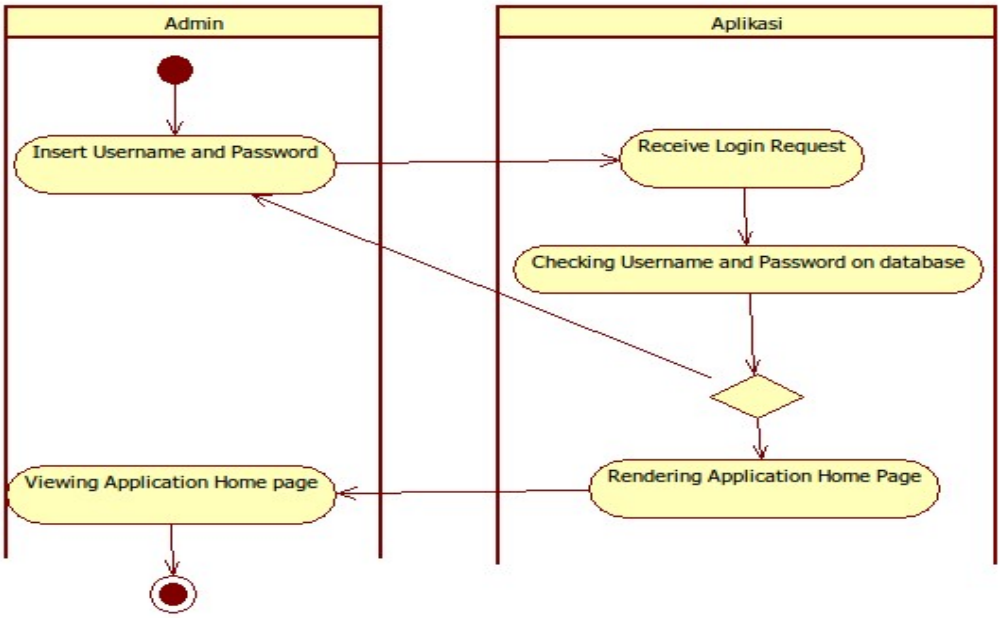
**Tabel 3.17 Skenario Use Case Delete Kelas.**

| Identifikasi | |
|---|---|
| Nama | Delete Kelas. |
| Tujuan | Menghapus data kelas yang ada di dalam *database*. |
| Deskripsi | Proses untuk menghapus data siswa yang ada dalam *database* dan menampilkan data siswa terbaru dalam bentuk tabel. |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi awal : Admin telah masuk kedalam menu siswa. | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih data yang akan dihapus. | 2. Menerima permintaan *delete* data. |
| | 3. Menyiapkan form konfirmasi *delete*. |
| 4. Mengonfirmasi perintah *delete* dalam form . | 5. Menghapus data dalam database. |
| | 6. Menyusun data kelas dalam bentuk tabel |
| 7. Menampilkan data kelas yang telah disimpan dalam bentuk tabel | |
| Kondisi akhir : Admin berhasil menghapus data kelas dan menampilkan data kelas yang terbaru dalam bentuk tabel. | |

### 3.2.1.3 Activity Diagram

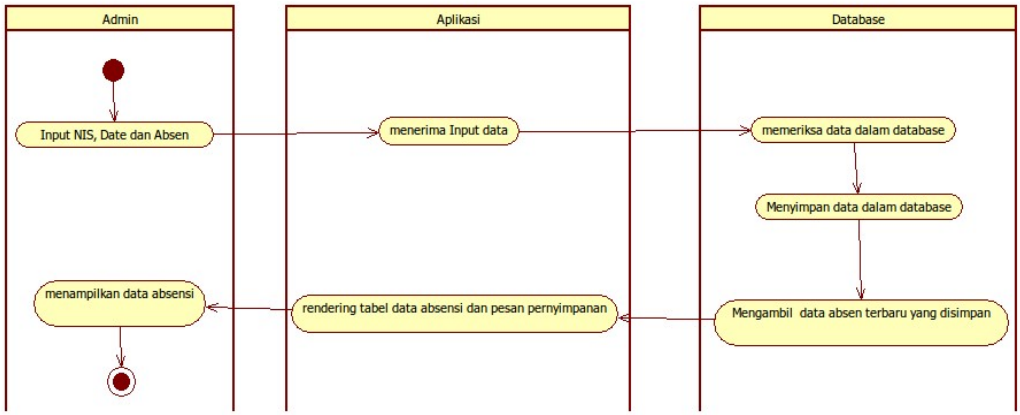
*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity 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. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

1. Activity Diagram Login



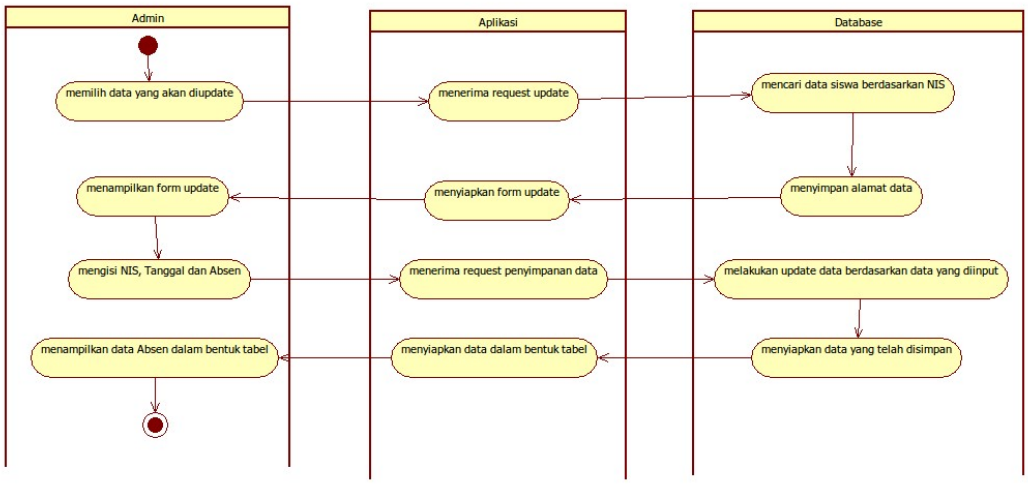

**Gambar 3.3 Activity Diagram Login**

2. Activity Diagram Penambahan Absensi



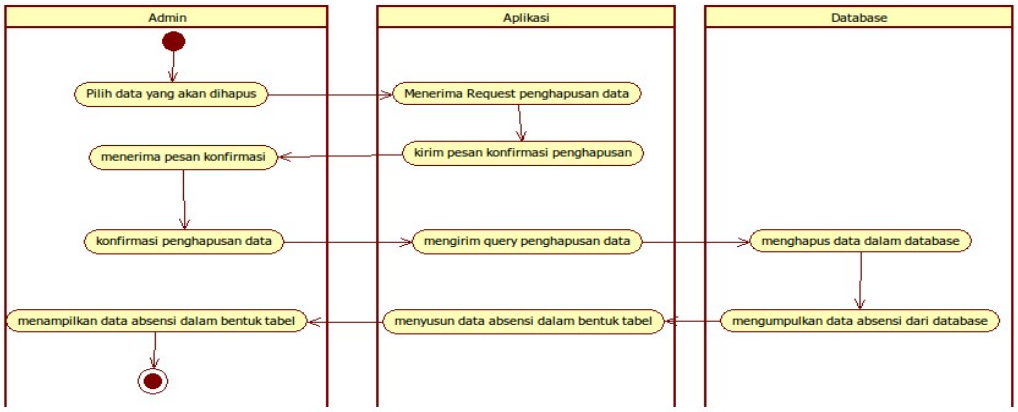

**Gambar 3.4 Activity Diagram Penambahan Absensi.**

3. Activity Diagram Update Absensi.



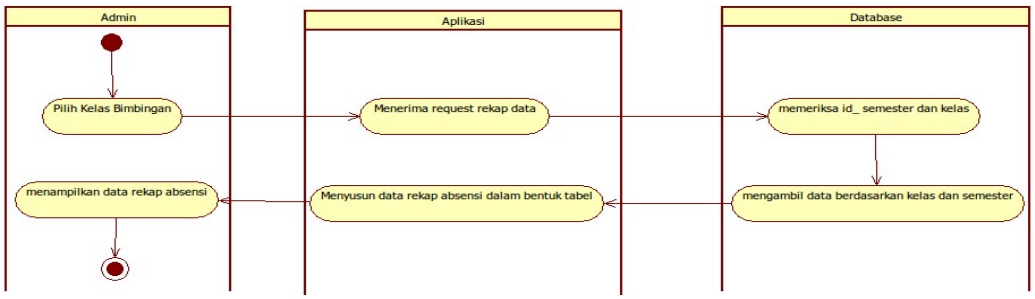

**Gambar 3.5 Activity Diagram Update Absensi**

4. Activity Diagram Delete Absensi.



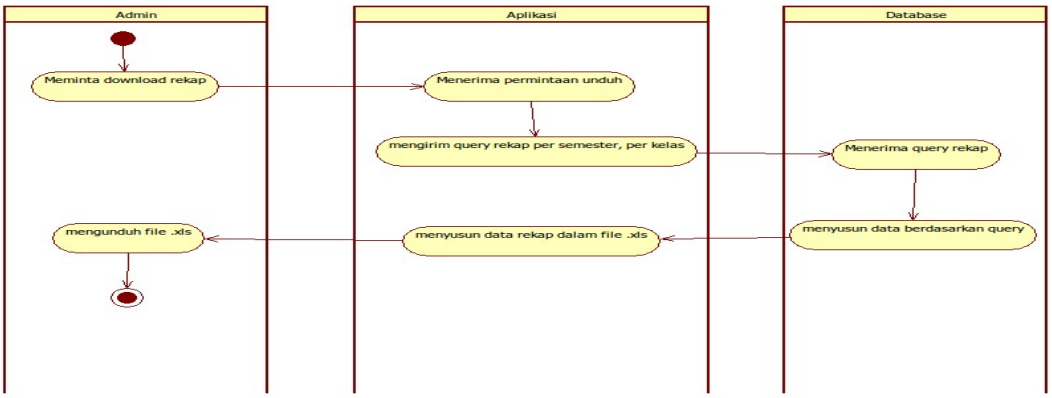

**Gambar 3.6 Activity Diagram Delete Absensi**

**5.** Activity Diagram Perekapan Data.



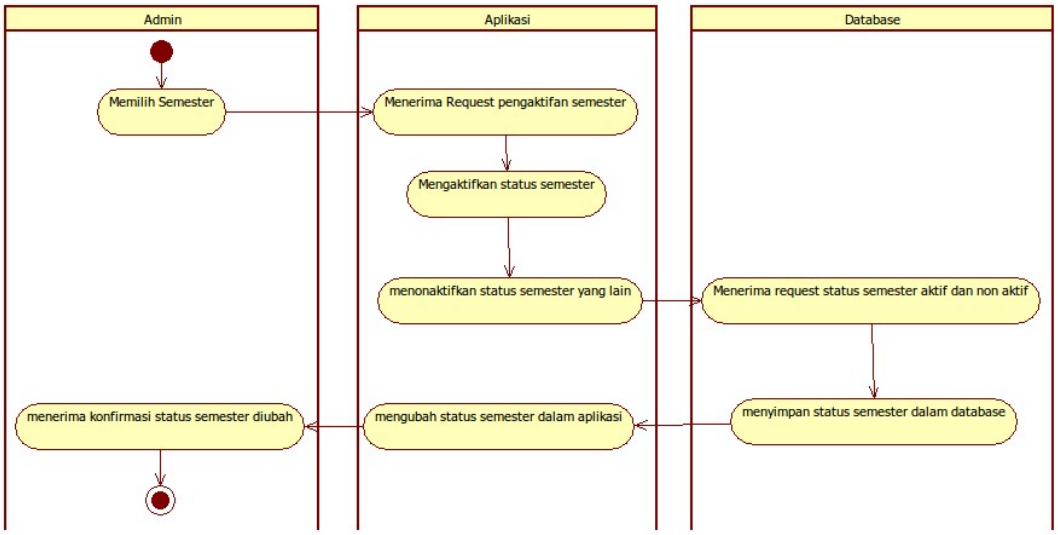

**Gambar 3.7 Activity Diagram Perekapan Data.**

6. Activity Diagram Download Rekap.



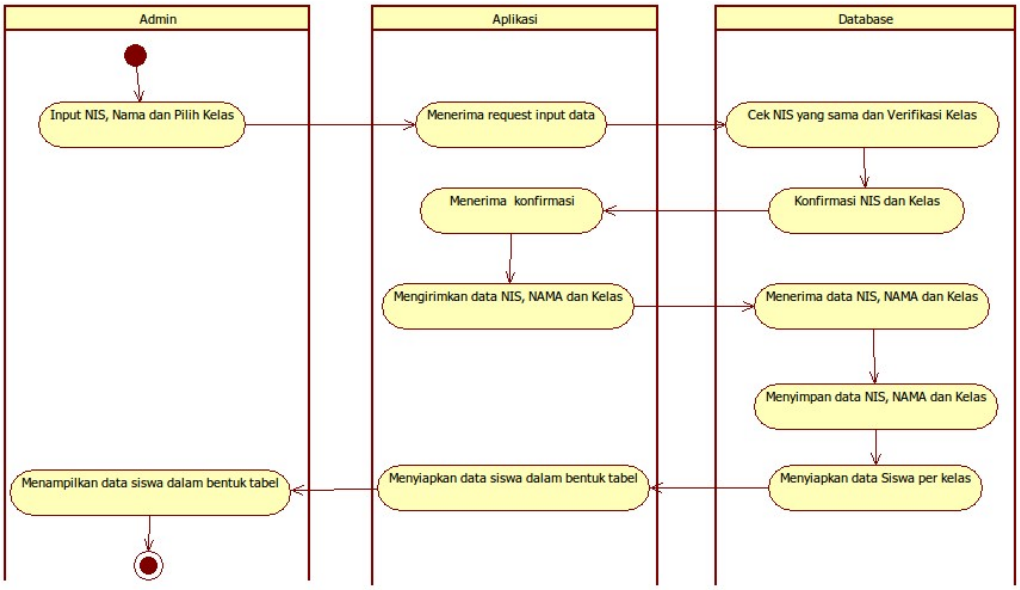

**Gambar 3.8 Activity Diagram Download Rekap.**

7. Activity Diagram Pemilihan Semester.



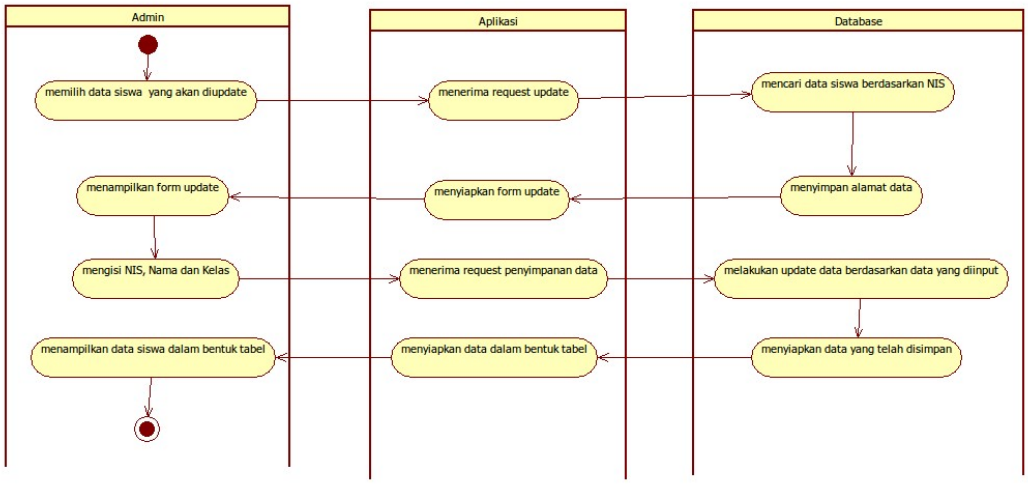

**Gambar 3.9 Activity Diagram Pemilihan Semester.**

8. Activity Diagram Penambahan Siswa.



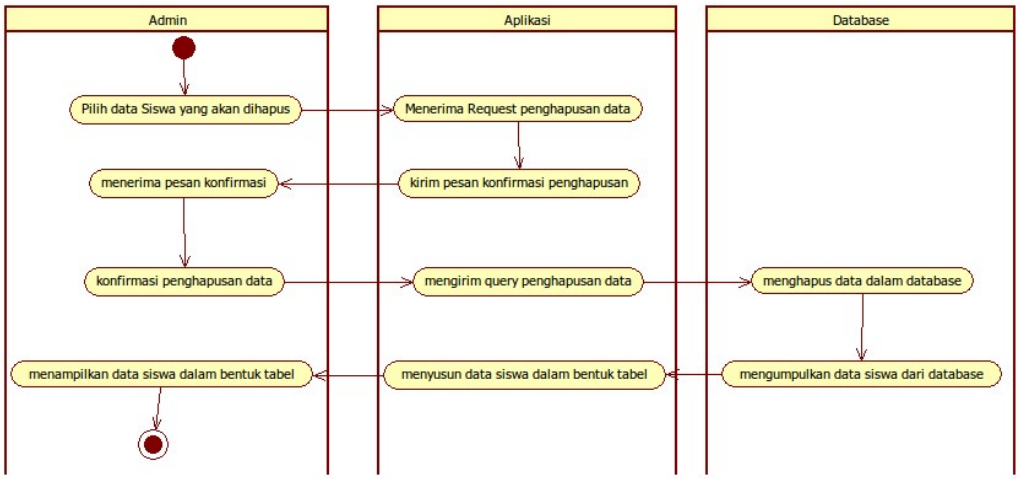

**Gambar 3.10 Activity Diagram Penambahan Siswa.**

9. Activity Diagram Update Siswa.



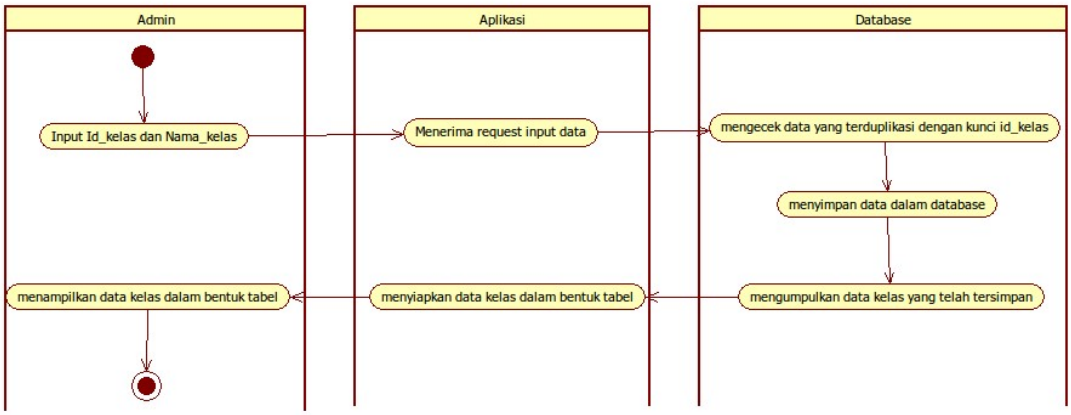

**Gambar 3.11 Activity Diagram Update Siswa.**

10. Activity Diagram Delete Siswa.



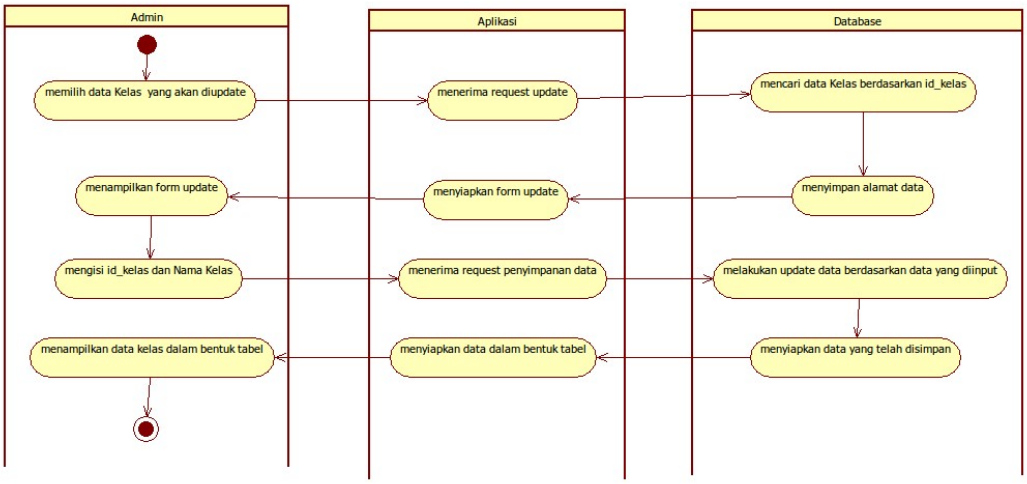

**Gambar 3.12 Activity Diagram Delete Siswa.**

11. Activity Diagram Penambahan Kelas.



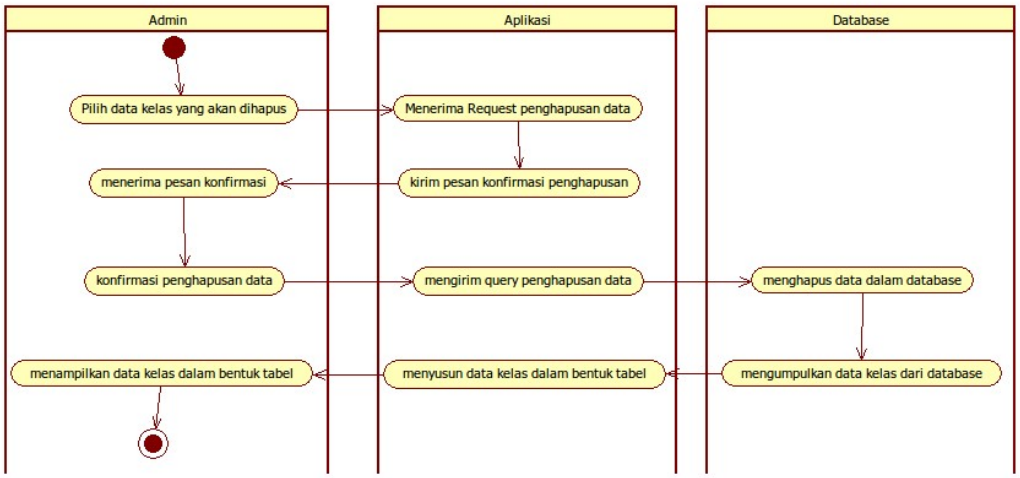

**Gambar 3.13 Activity Diagram Penambahan Kelas.**

12. Activity Diagram Update Kelas.




**Gambar 3.14 Activity Diagram Update Kelas.**

13. Activity Diagram Delete Kelas.

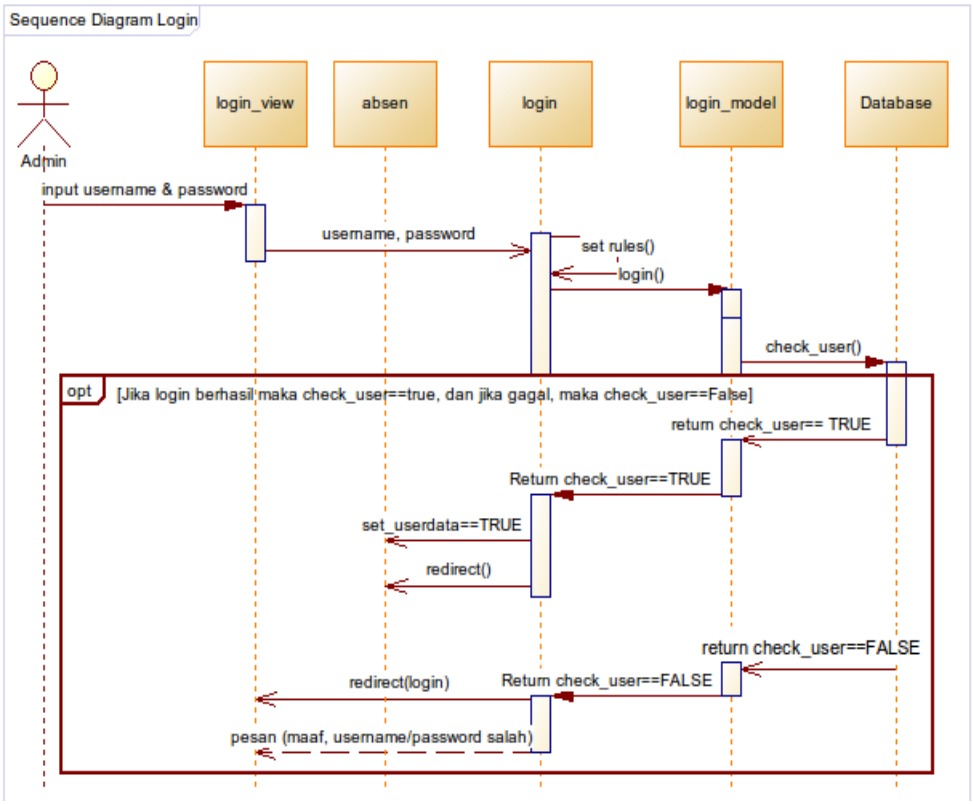



**Gambar 3.15 Activity Diagram Delete Kelas.**

### 3.2.1.4 Sequence Diagram

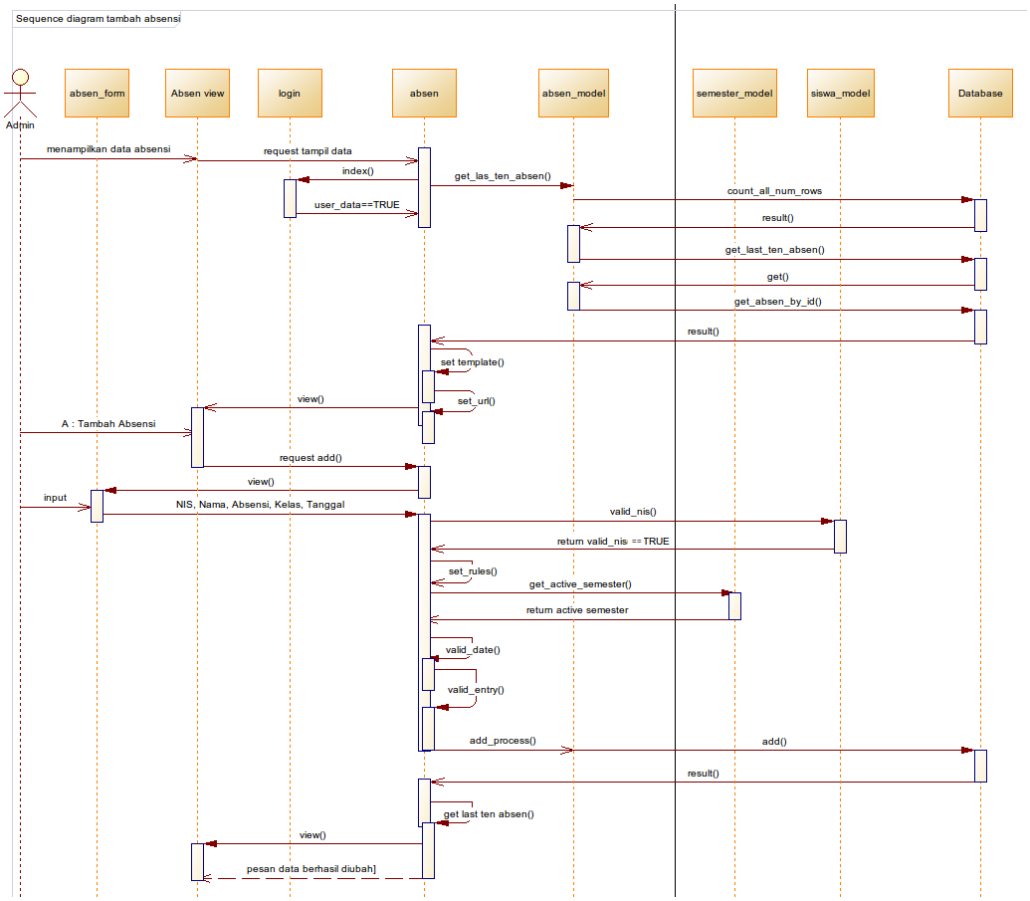
Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya). Sequence diagram terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait). Pada Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah event untuk menghasilkan output tertentu.

1. Sequence Diagram Login



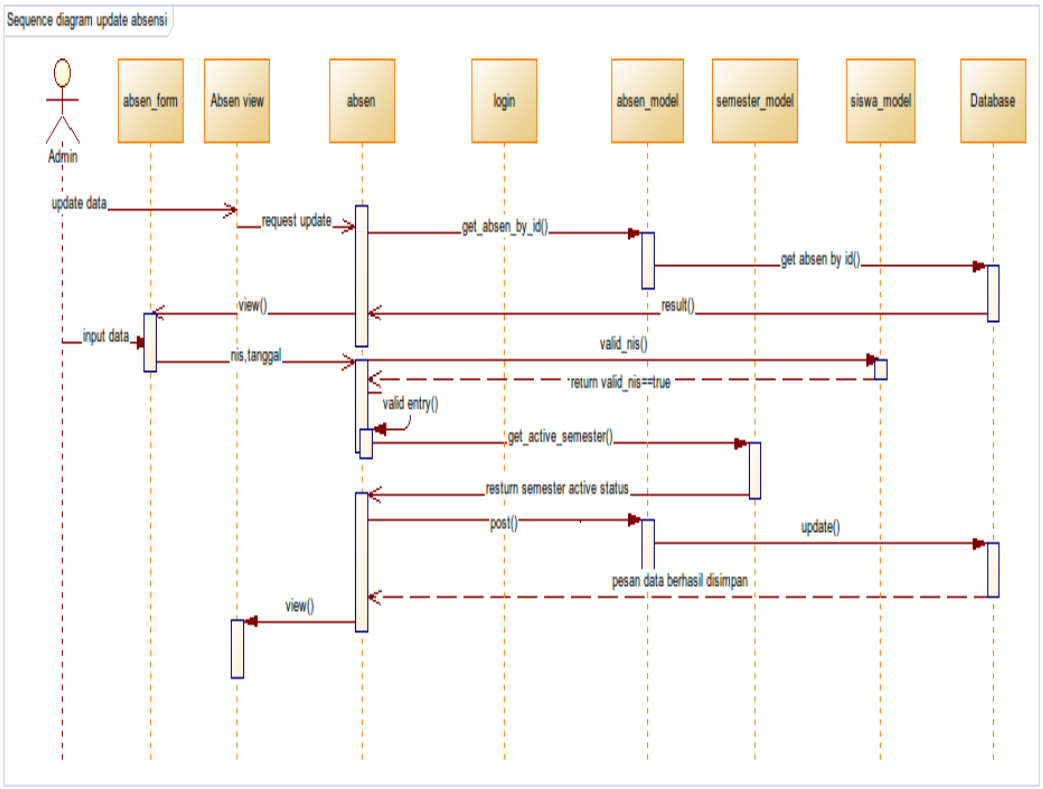

**Gambar 3.16 Sequence Diagram Login**

2. Sequence Diagram Penambahan Absensi



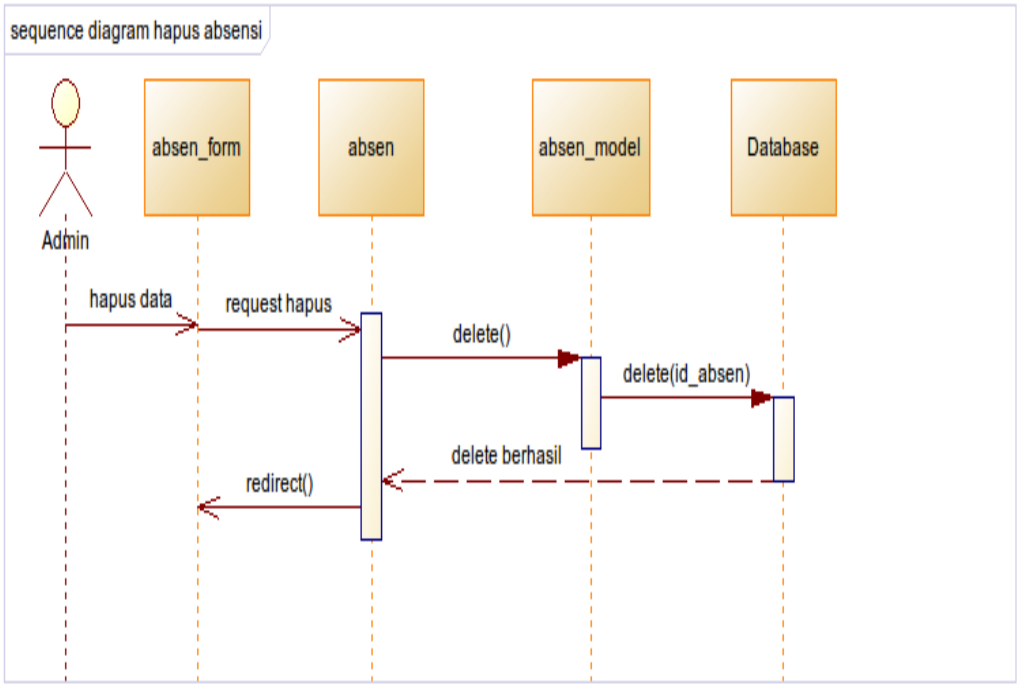

**Gambar 3.17 Sequence Diagram Penambahan Absensi**

3. Sequence Diagram Update Absensi



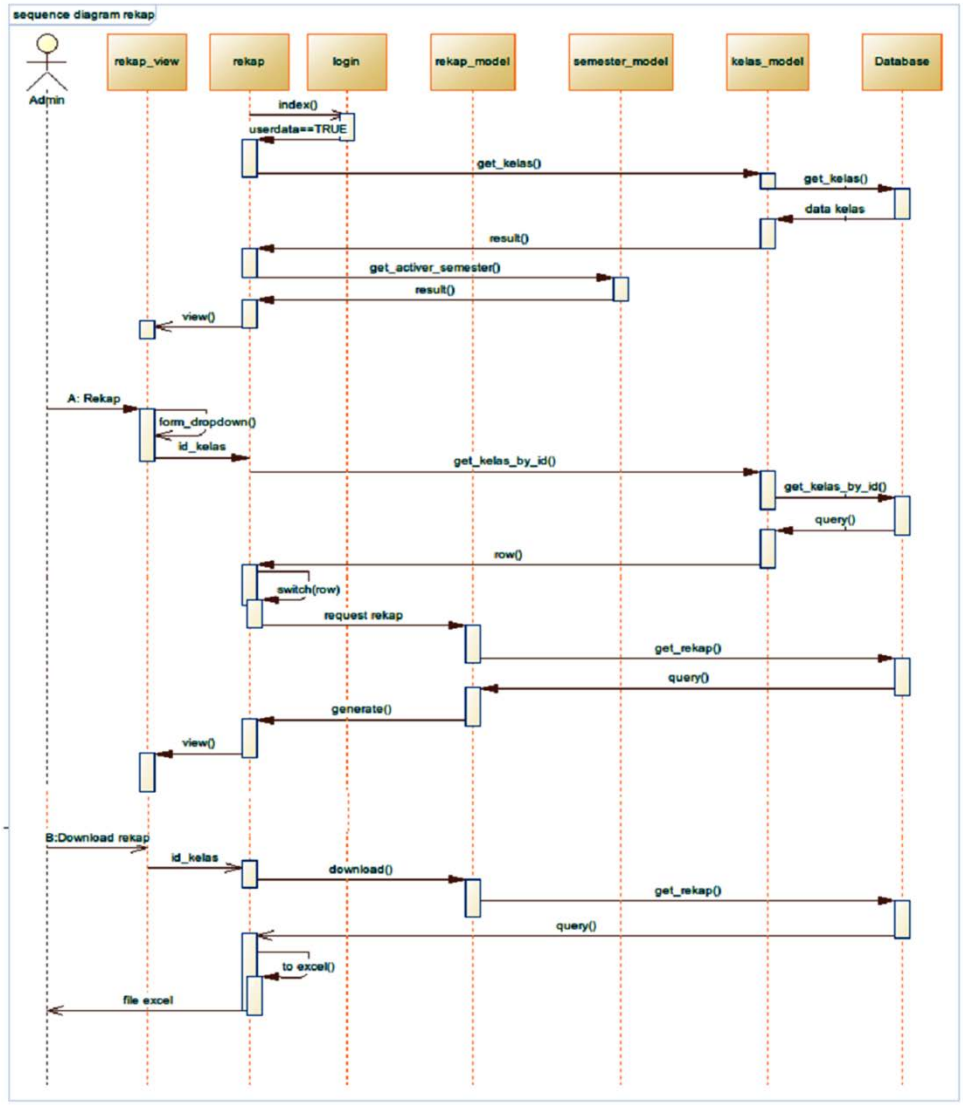

**Gambar 3.18 Sequence Diagram Update Absensi.**

4. Sequence Diagram Delete Absensi.



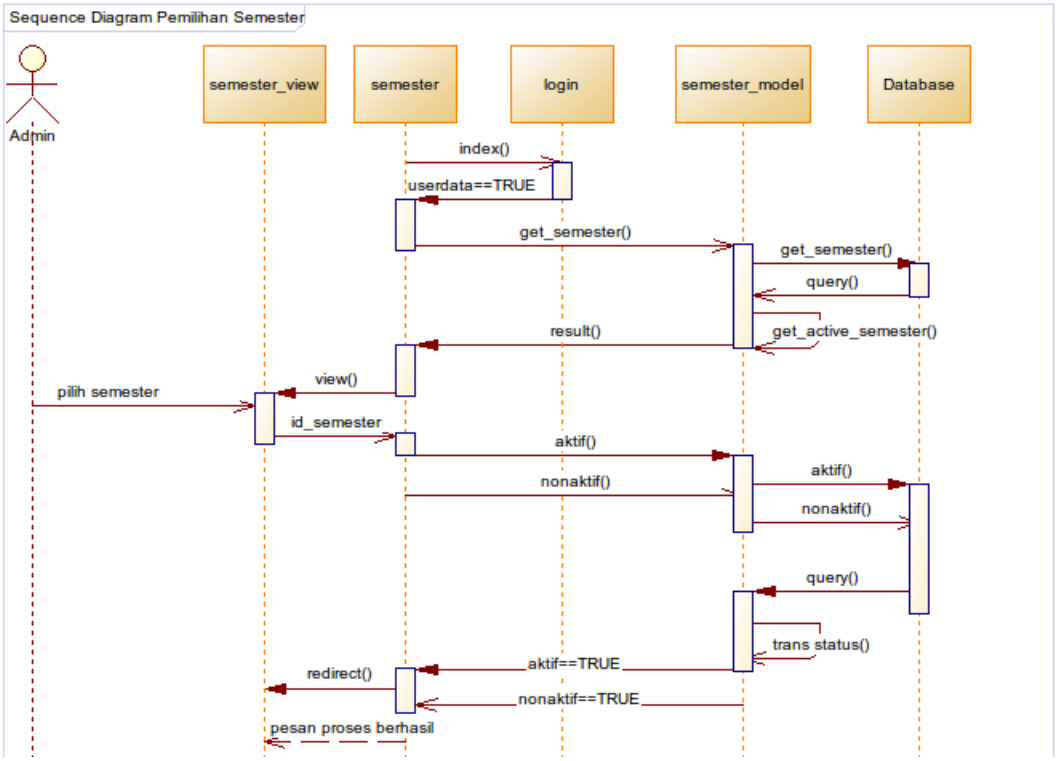
**Gambar 3.19 Sequence Diagram Delete Absensi.**

5. Sequence Diagram Pengolahan Rekap.



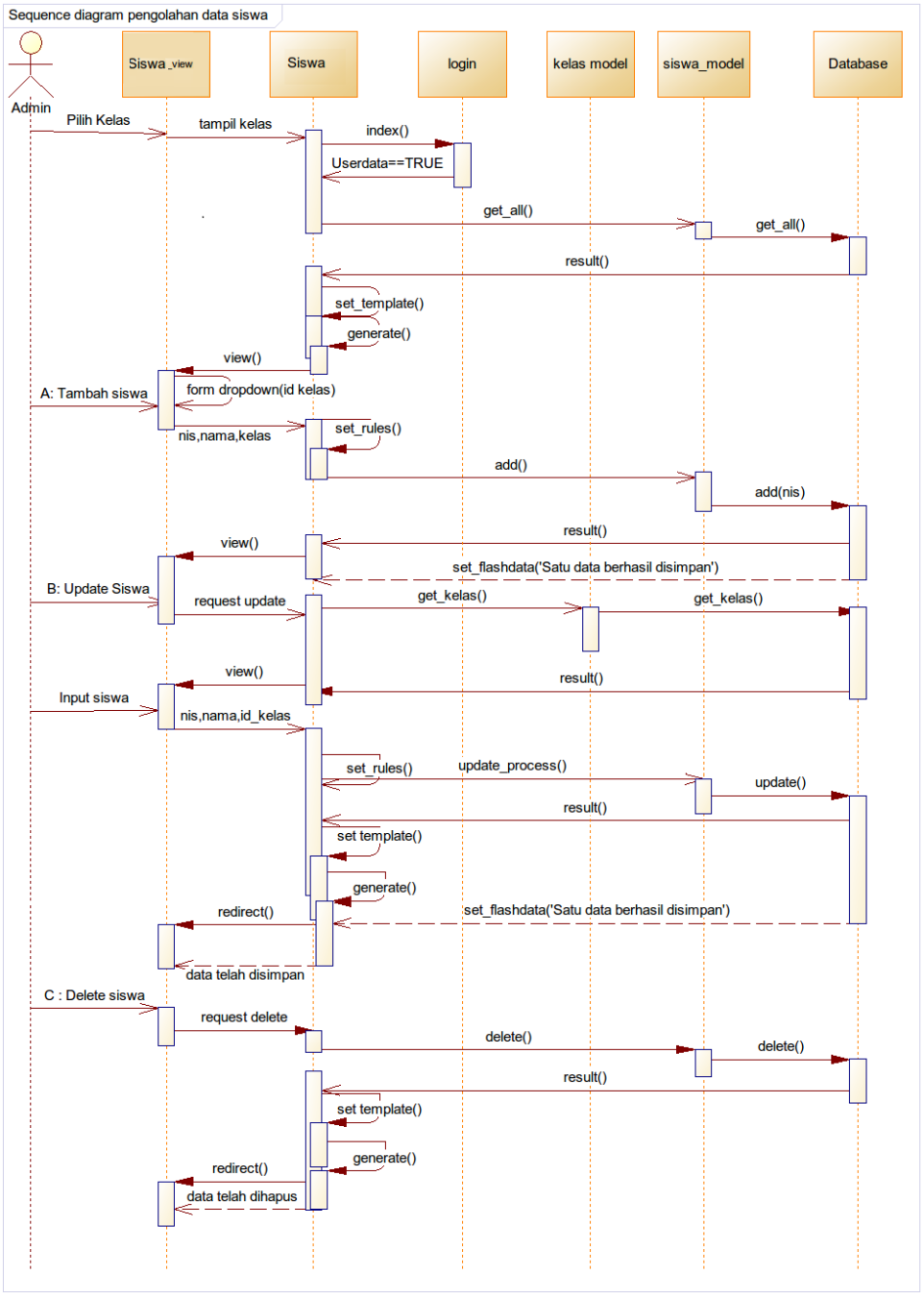

**Gambar 3.20 Sequence Diagram Pengolahan Rekap**

8. Sequence Diagram Pemilihan Semester



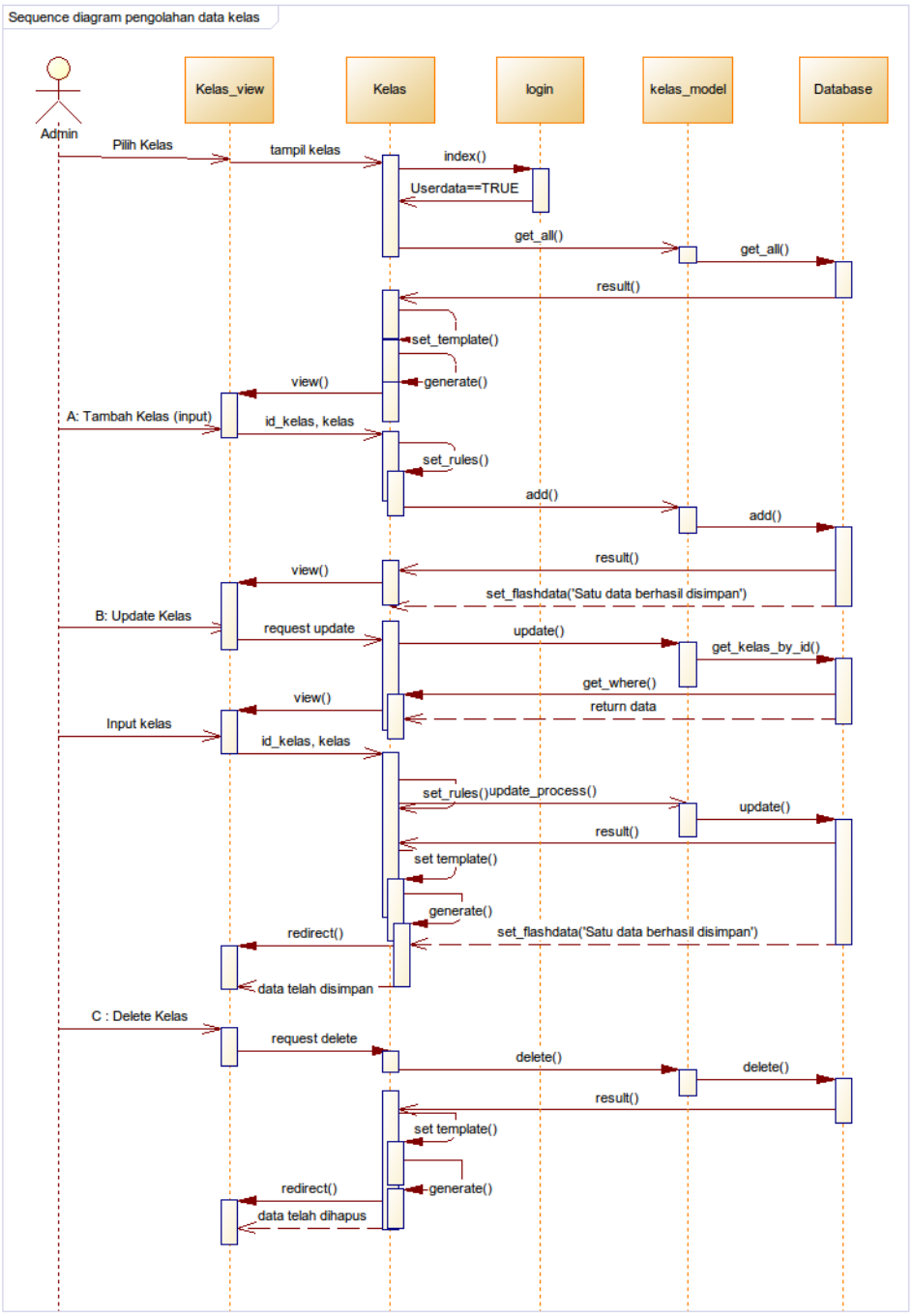

**Gambar 3.21 Sequence Diagram Pemilihan Semester**

9. Sequence Diagram Pengolahan Siswa.



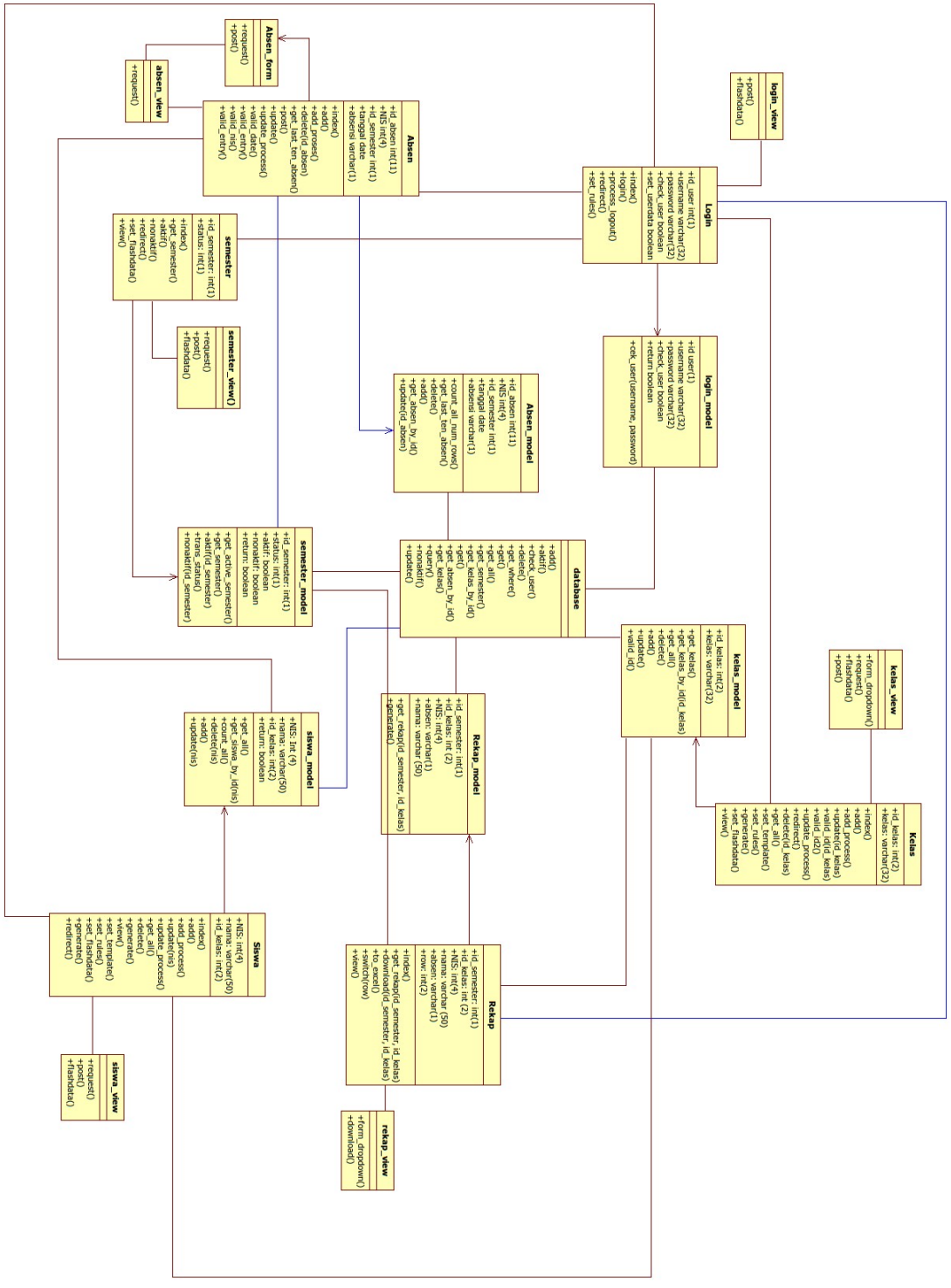

**Gambar 3.22 Sequence Diagram Pengolahan Siswa.**

10. Sequence Diagram Pengolahan Kelas.




**Gambar 3.23 Sequence Diagram Pengolahan Kelas.**

### 3.2.1.5 Class Diagram




**Gambar 3.24 Class Diagram Aplikasi Absensi Online Home-Schooling**

Spesifikasi kelas berisi pemaparan setiap kelas yang terdapat pada class diagram aplikasi absensi online. Class diagram aplikasi absensi online yang dibangun terdiri dari tiga jenis kelas, yaitu *class control*, dan *class entity* dan *class boundary*. Adapun daftar setiap kelas yang terdapat pada class diagram aplikasi ini dapat dilihat dalam tabel berikut.

**Tabel 3.17 Daftar Spesifikasi Tabel**

| No | Nama Kelas | Jenis Kelas | Deskripsi |
|---|---|---|---|
| 1 | Login_model | *class entity* | Kelas yang mengatur operasi *login* kedalam sistem serta mengatur session admin. |
| 2 | Absen_model | *class entity* | Kelas yang mengatur pengolahan data absen dalam database. |
| 3 | Siswa_model | *class entity* | Kelas yang mengatur pengolahan data siswa dalam database. |
| 4 | Semester_model | *class entity* | Kelas yang mengatur pengaktifan semester dalam system. |
| 5 | Kelas_model | *class entity* | Kelas yang mengatur pengolahan data kelas dalam database. |
| 6 | Rekap_model | *class entity* | Kelas yang mengatur rekap data dalam database, serta pengunduhan rekap data dalam sistem. |
| 7 | Login | *class control* | Kelas yang mengatur hubungan antara database dengan kelas model. |

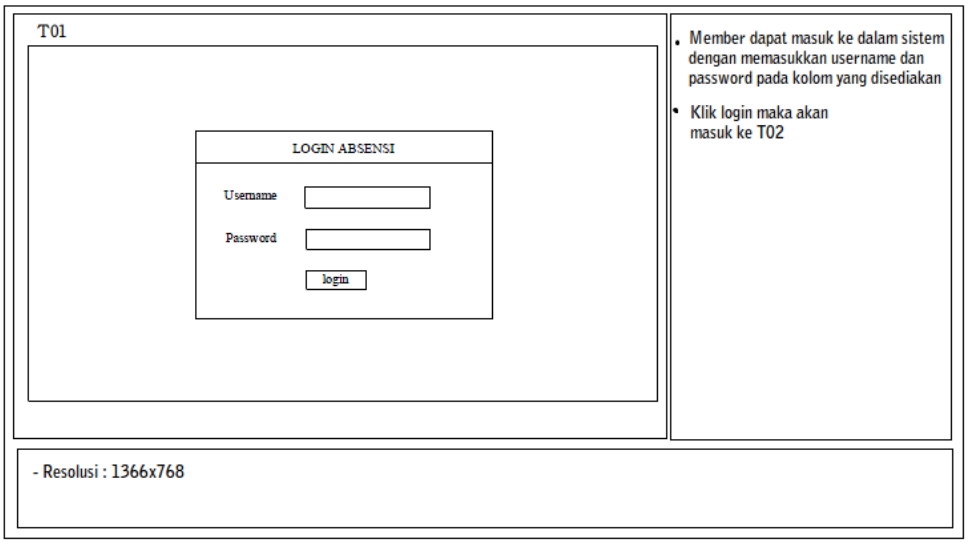
| | | | |
|---|---|---|---|
| 8 | Absen | *class control* | Kelas yang mengatur penyajian hasil pengolahan absen. |
| 9 | Siswa | *class control* | Kelas yang mengatur penyajian pengolahan data siswa |
| 10 | Semester | *class control* | Kelas yang mengatur penyajian pengaktifan dan penonaktifan semester. |
| 11 | Kelas | *class control* | Kelas yang mengatur penyajian data kelas . |
| 12 | Rekap | *class control* | Kelas yang mengatur penyajian rekap data dari database. |
| 13 | Login_view | *class boundary* | Kelas yang mengirimkan perintah login. |
| 14 | Absen_view | *class boundary* | Kelas yang mengirimkan permintaan operasi tambah, hapus serta update absen serta menampilkan data absensi. |
| 15 | Absen_form | *class boundary* | Kelas yang menampilkan tampilan tambah absen dan mengirimkan permintaan penambahan data absen. |
| 16 | Siswa_view | *class boundary* | Kelas yang mengirimkan permintaan operasi tambah, hapus serta update siswa serta menampilkan data siswa. |
| 17 | Semester_view | *class boundary* | Kelas yang mengirimkan permintaan operasi pengaktifan serta penonaktifan saat pemilihan semester, dan |

| | | | |
|---|---|---|---|
| | | | menampilkan data semester yang terdapat dalam database. |
| 18 | Kelas_view | *class boundary* | Kelas yang mengirimkan permintaan operasi tambah, hapus serta update kelas serta menampilkan data kelas. |
| 20 | Rekap_view | *class boundary* | Kelas yang menampilkan data rekap per kelas serta mengirimkan permintaan *download* data. |

### 3.3. Desain Antar Muka

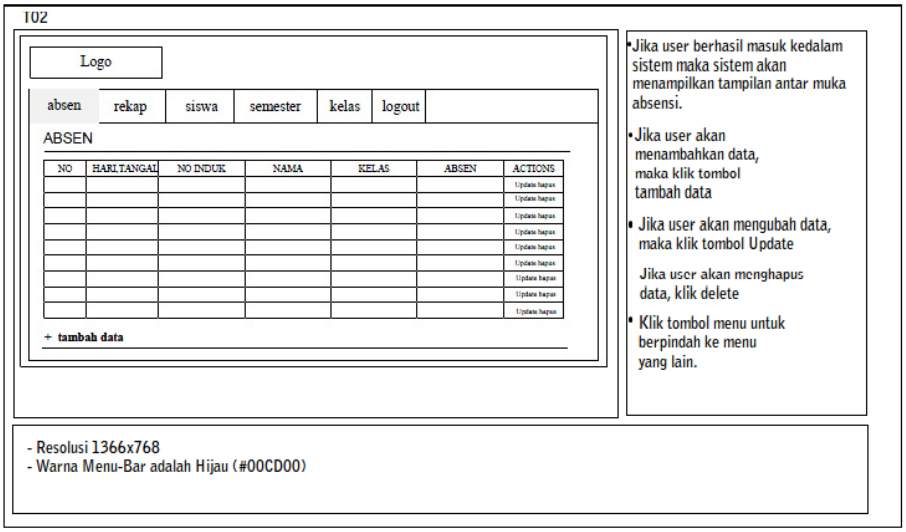
Interface atau antar muka merupakan tampilan dari suau program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan user. Sistem yang akan dibangun diharapkan menyediakan interface yang mudah dipahami dan digunakan oleh user.

1. Antar muka login (T01).



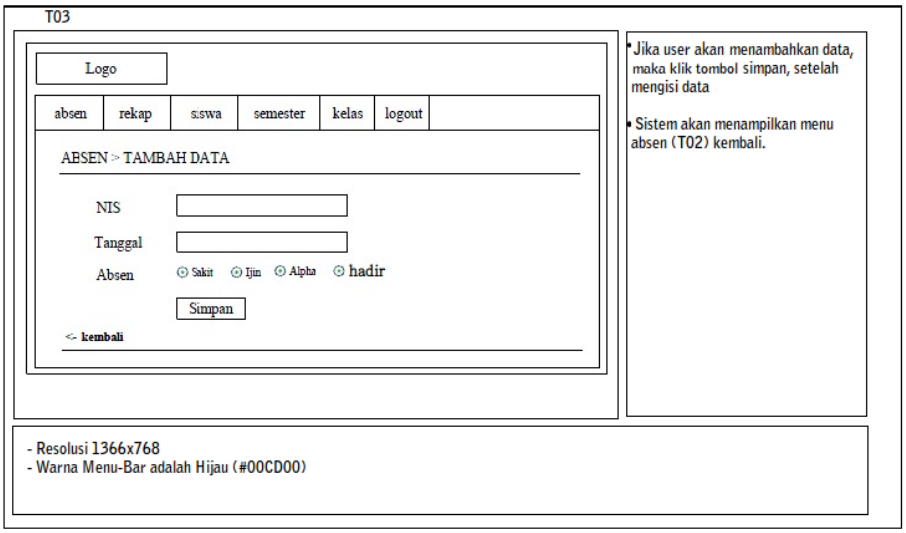

Gambar 3.25 Desain antar muka login

2. Antar Muka Absen (T02).



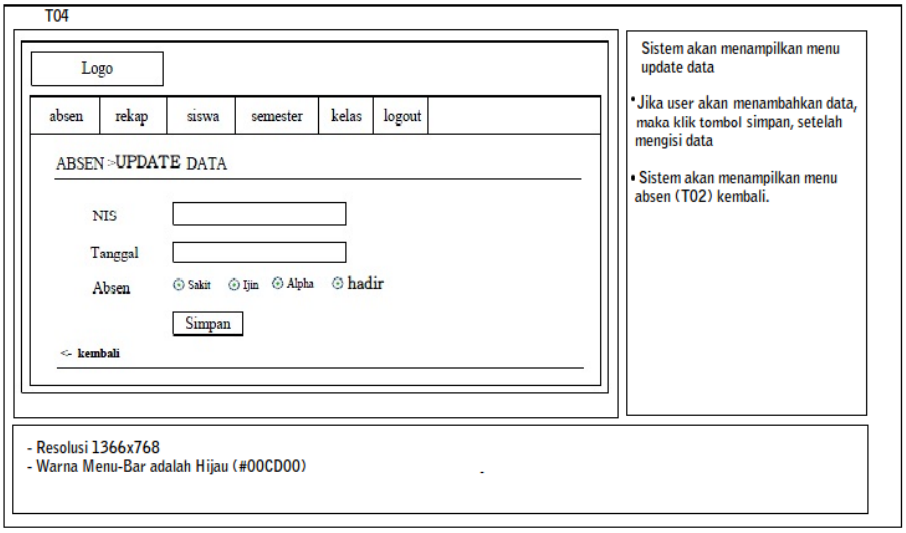

Gambar 3.26 Desain antar muka absen

3. Antar muka tambah absen (T03).



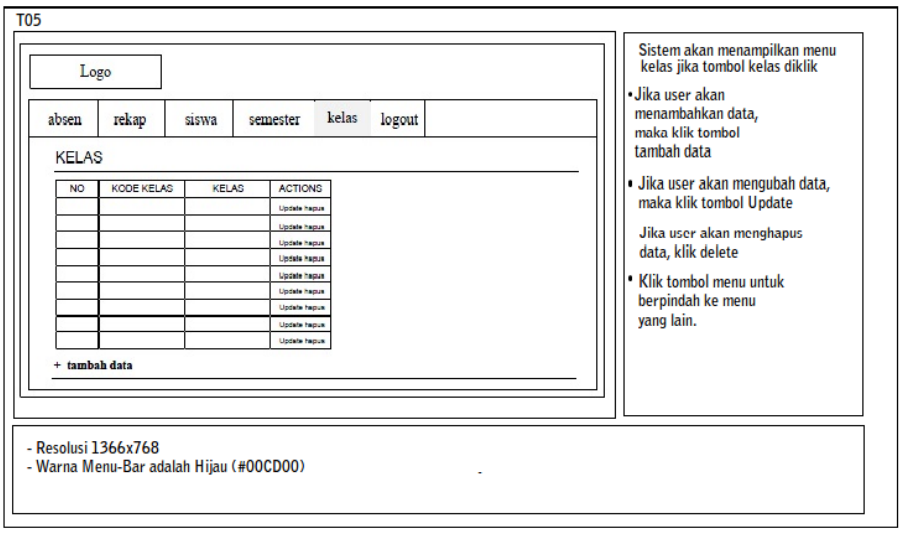

Gambar 3.27 Desain antar muka tambah absen.

4. Antar muka update absen (T04).



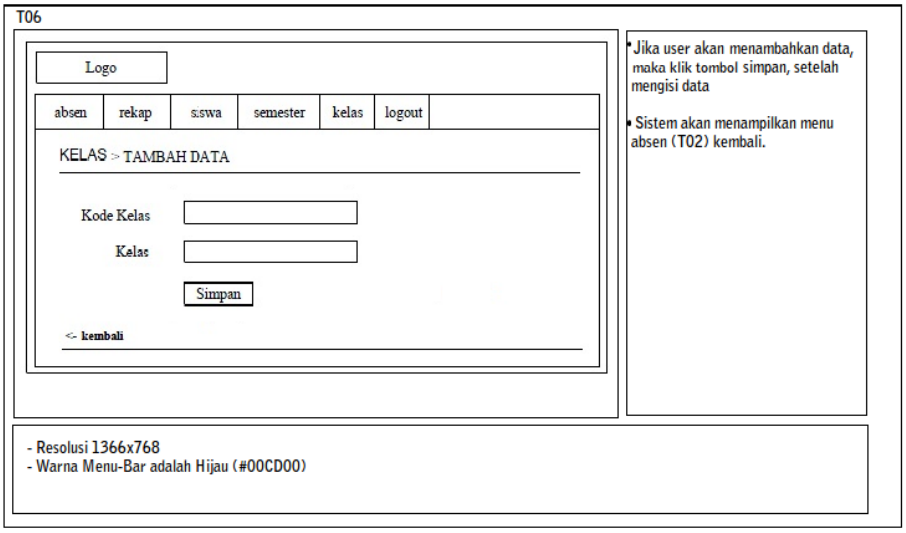

Gambar 3.28 Desain antar muka update absen.

5. Antar muka kelas (T05).



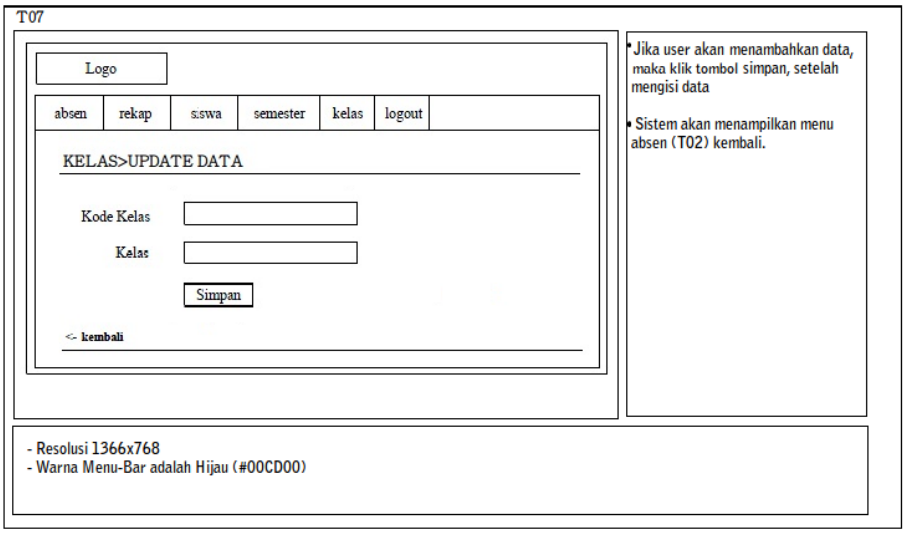

Gambar 3.29 Desain antar muka kelas.

6. Antar muka tambah kelas (T06).



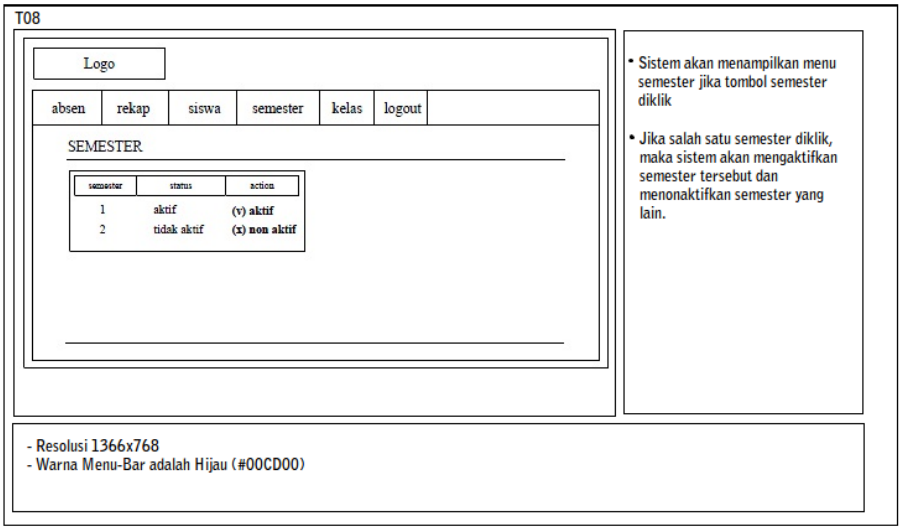

Gambar 3.30 Desain antar muka tambah kelas.

7. Antar muka update kelas (T07).



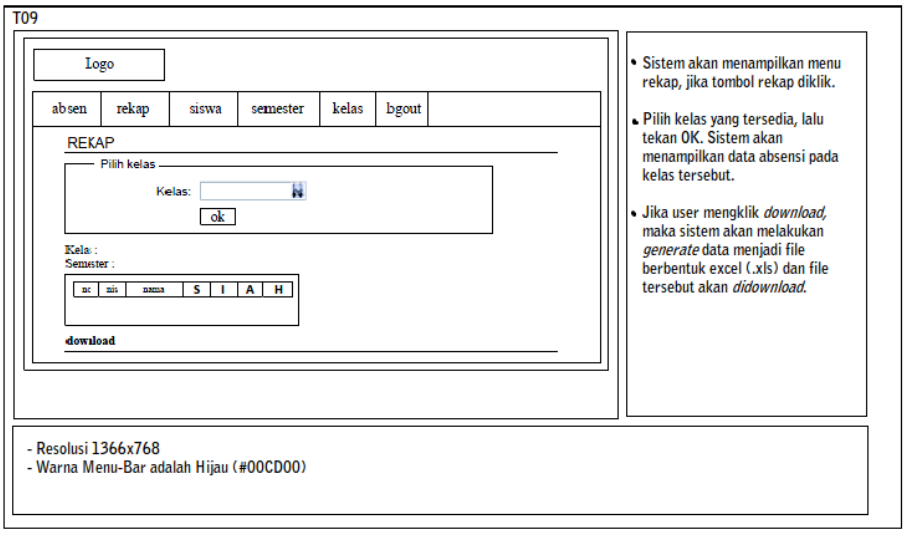

Gambar 3.31 Desain antar muka update kelas.

8. Antar muka pilih semester (T08)




Gambar 3.32 Desain antar muka pilih semester.

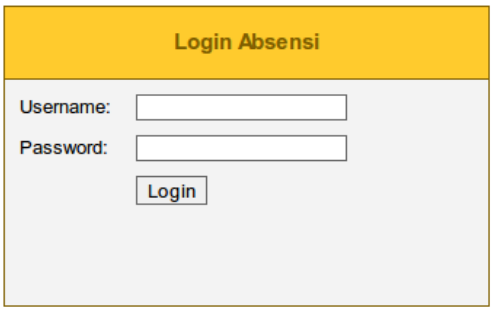
9. Antar muka Rekap data.




Gambar 3.33 Desain antar muka rekap data.

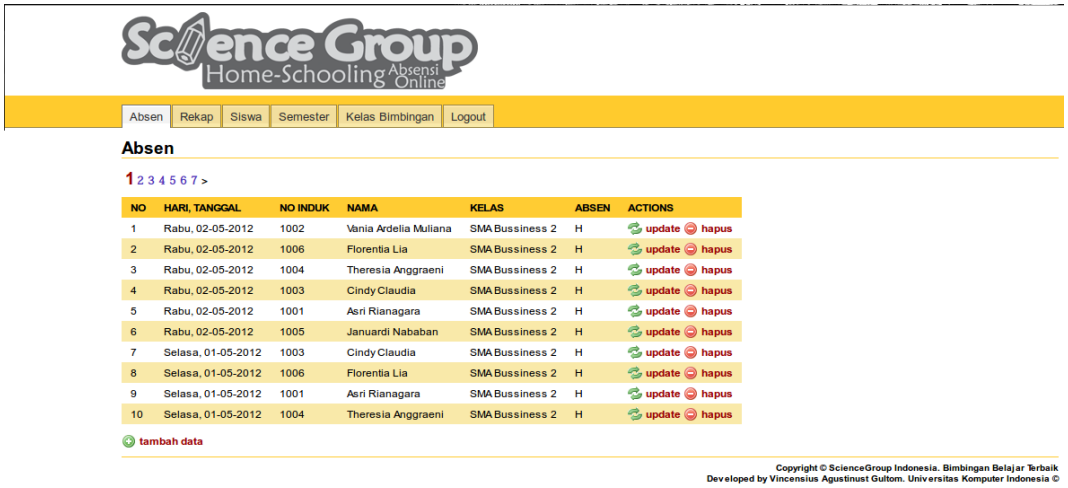
## 3.4. Implementasi

1. Tampilan Login.



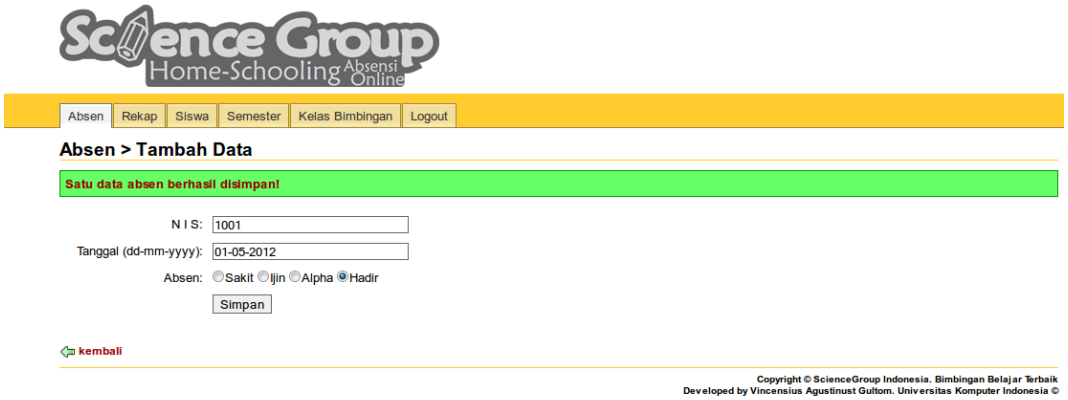

Gambar 3.34 Tampilan Login.

2. Tampilan Absen




Gambar 3.35 Tampilan Absen.

3. Tampilan Tambah Absen.



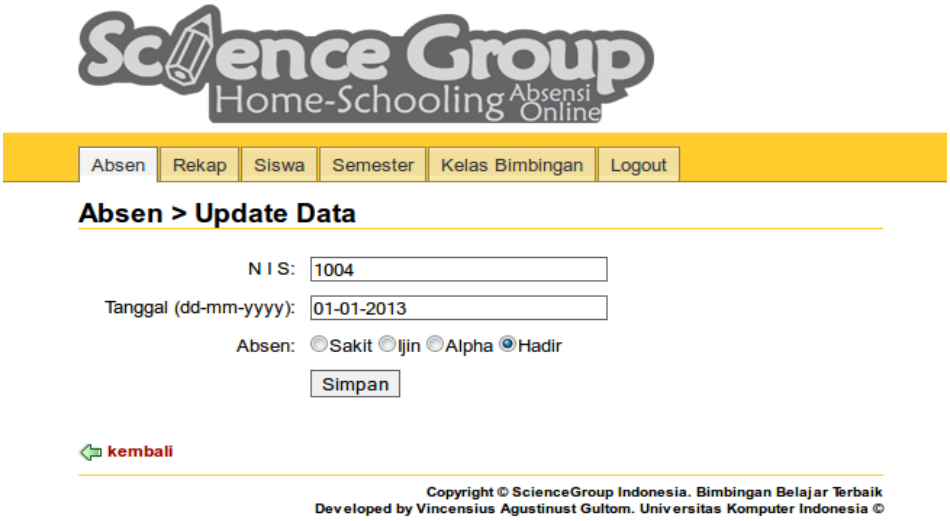

Gambar 3.36 Tampilan Tambah Absen.

4. Tampilan Error saat Tambah Absen.



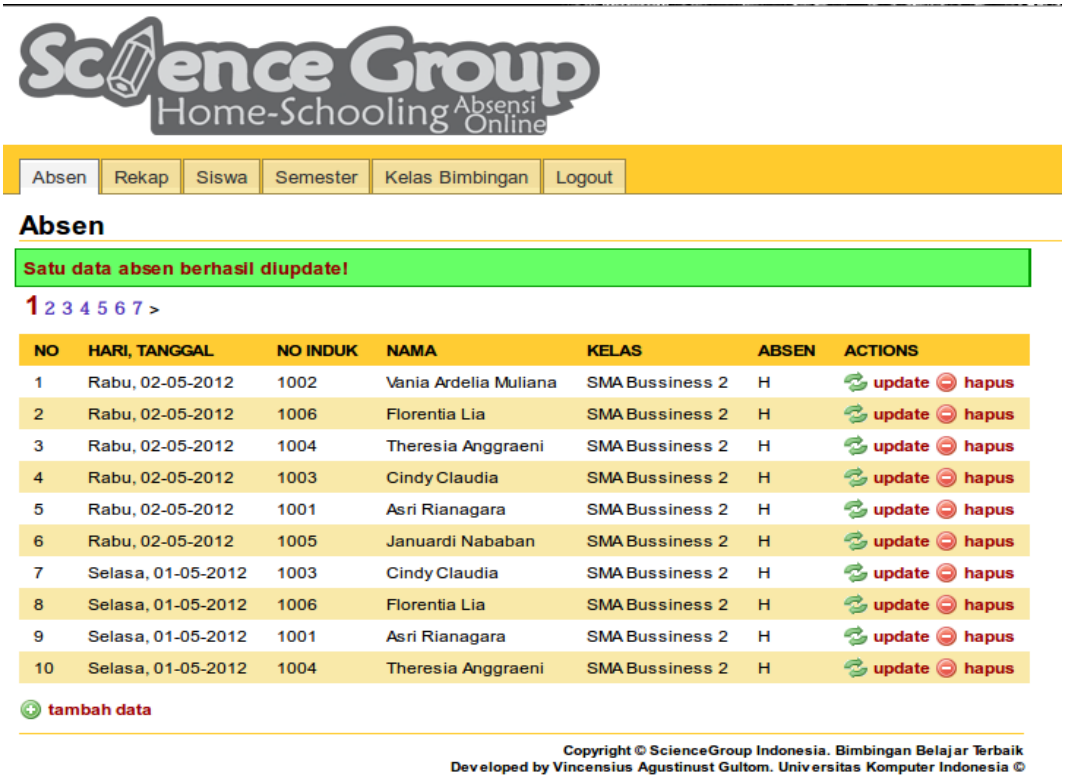

Gambar 3.37 Error saat Tambah Absen.

5. Tampilan Update Absen.



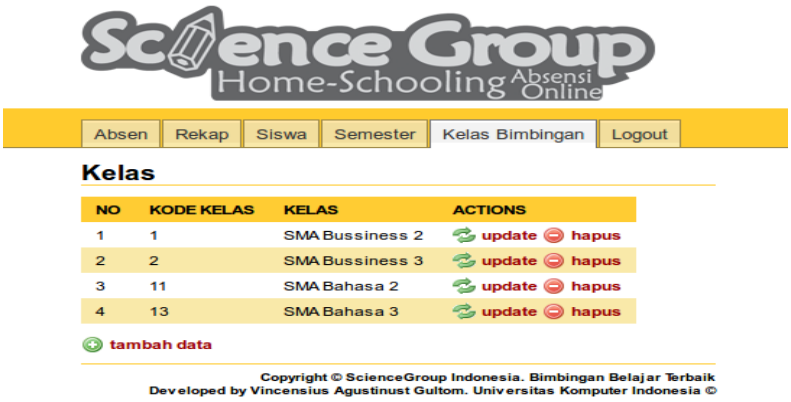

Gambar 3.38 Update Absen.

6. Tampilan Sukses Update Absen.



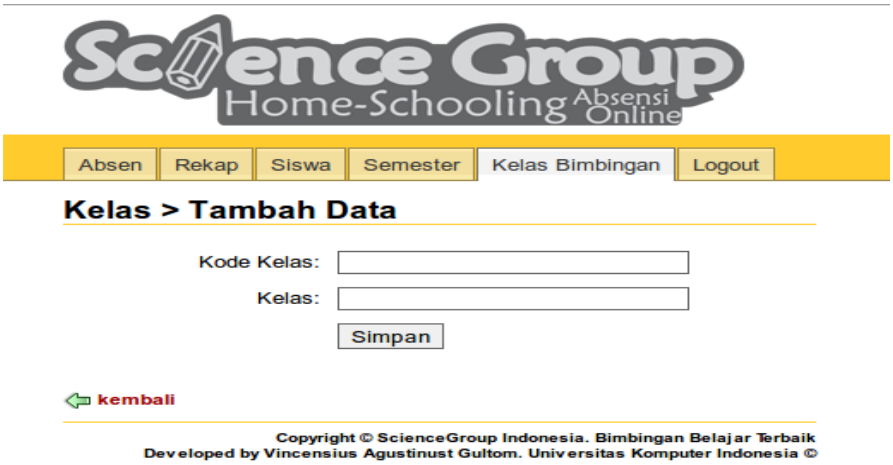

Gambar 3.39 Tampilan Sukses Update Absen.

7. Tampilan menu kelas.



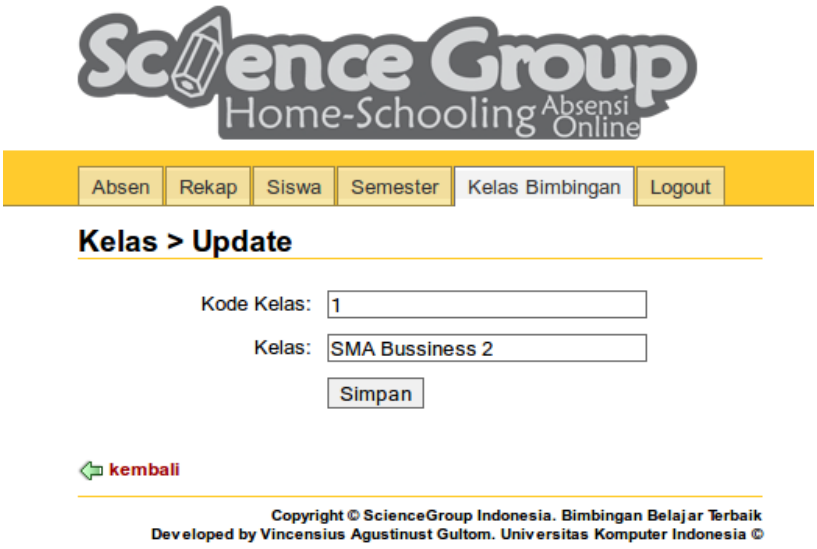

Gambar 3.40 Tampilan menu kelas.

8. Tampilan menu tambah kelas



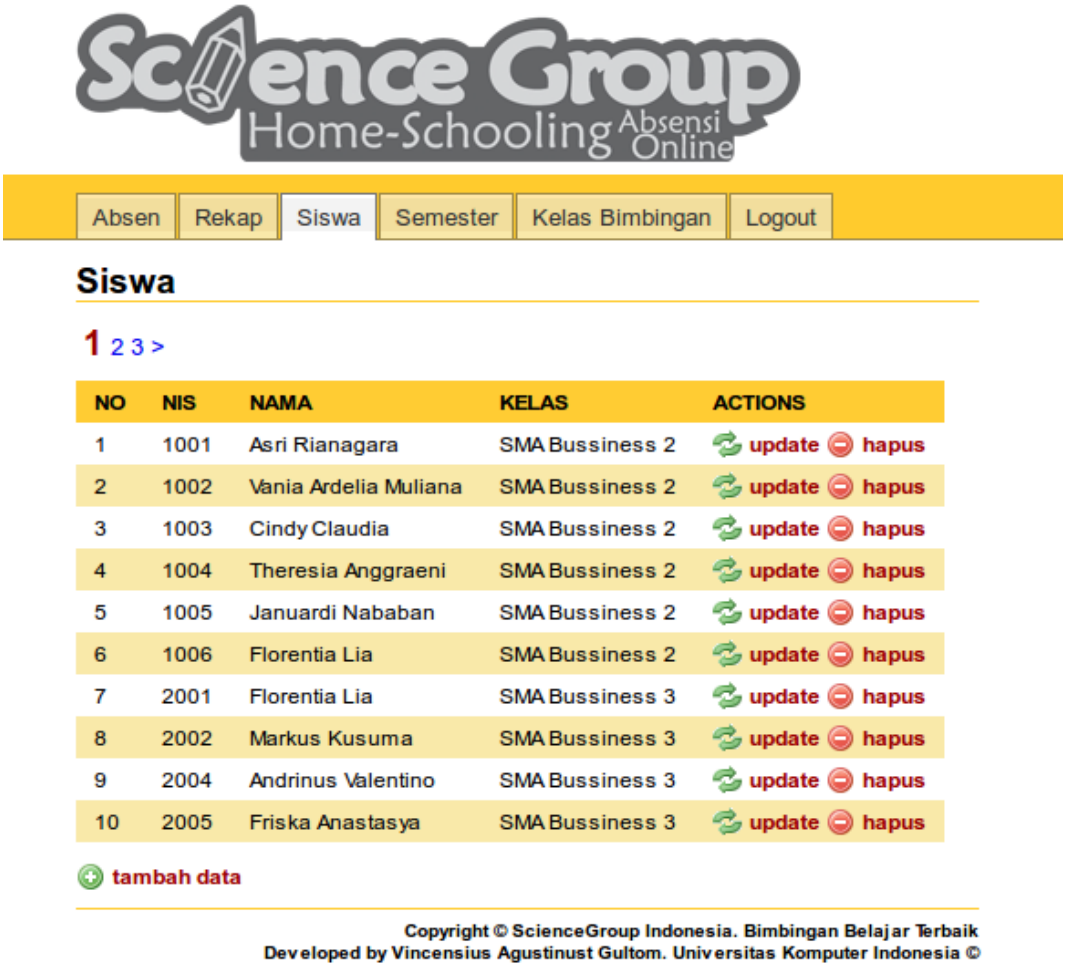

Gambar 3.41 Tampilan menu tambah kelas.

9. Tampilan menu update kelas.



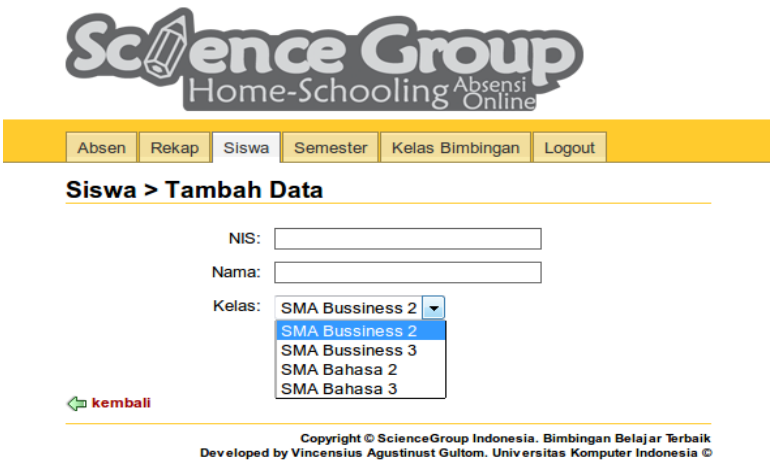

Gambar 3.42 Tampilan menu update kelas.

10. Tampilan menu siswa.



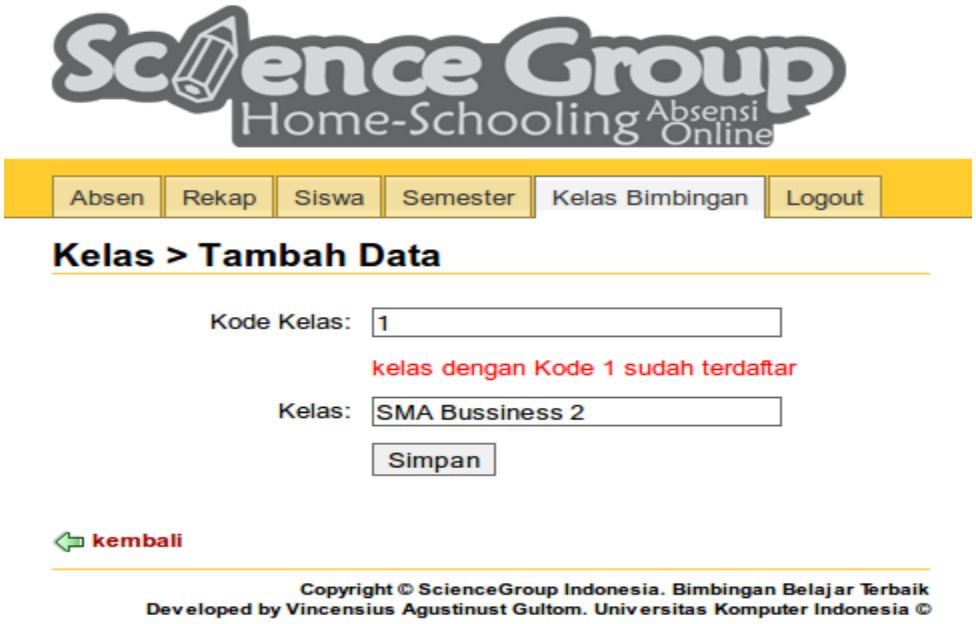

Gambar 3.43 Tampilan menu siswa.

11. Tampilan menu tambah data siswa



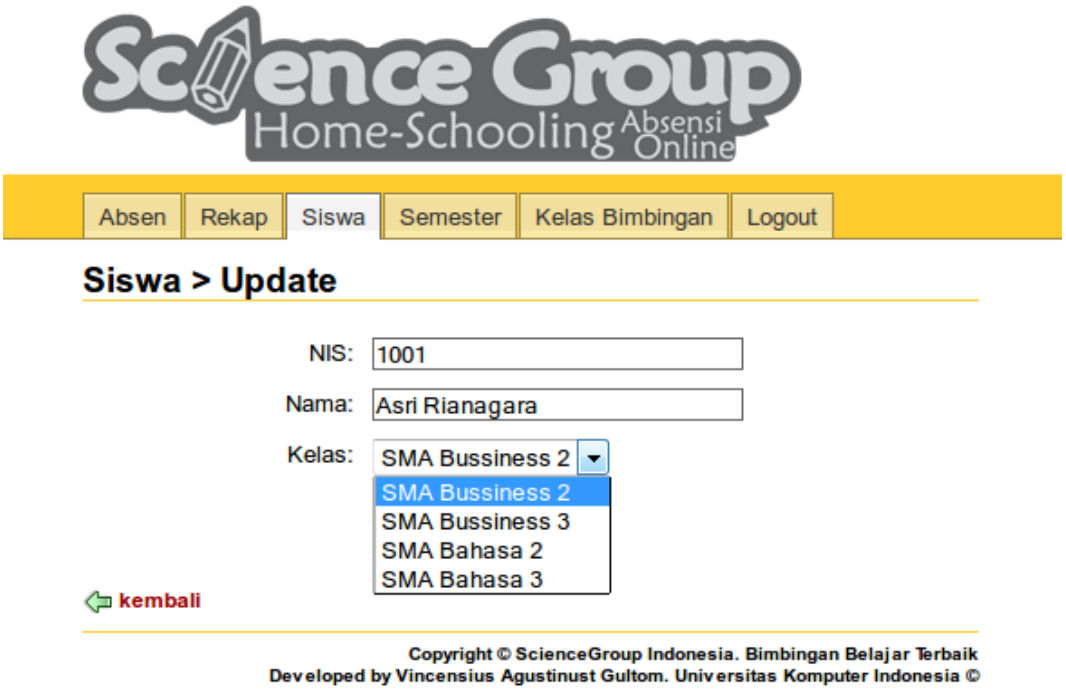

Gambar 3.44 Tampilan menu tambah data siswa.

12. Tampilan menu error saat menambahkan data siswa



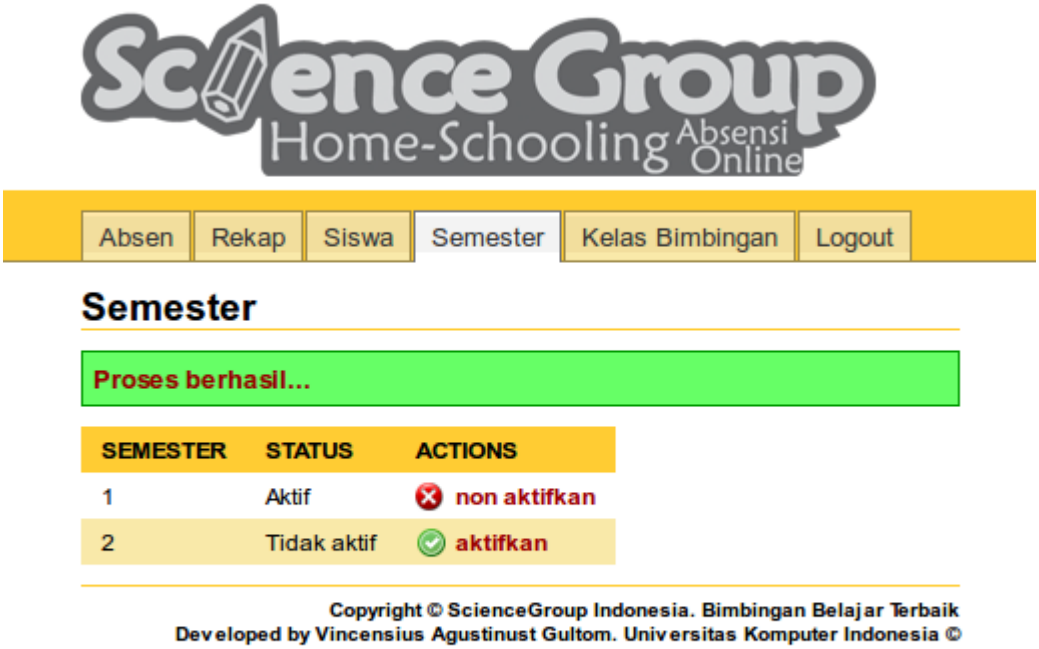

Gambar 3.45 Tampilan menu error saat menambahkan data siswa.

13. Tampilan menu update siswa.



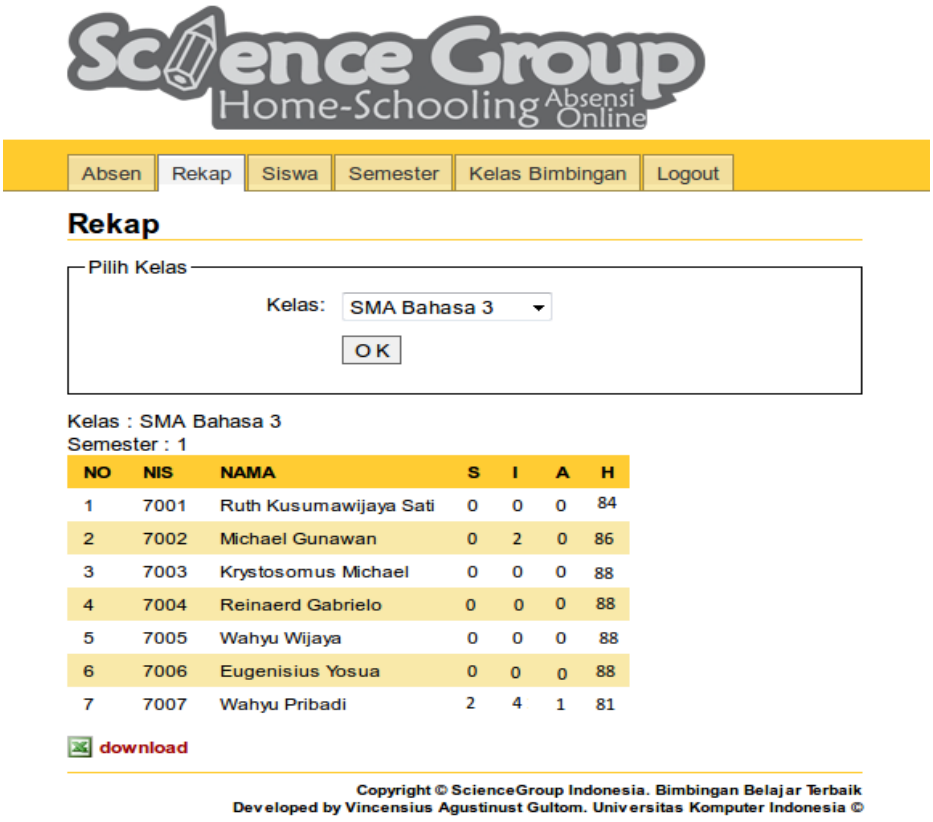

Gambar 3.46 Tampilan menu update siswa.

14. Tampilan menu pemilihan semester.



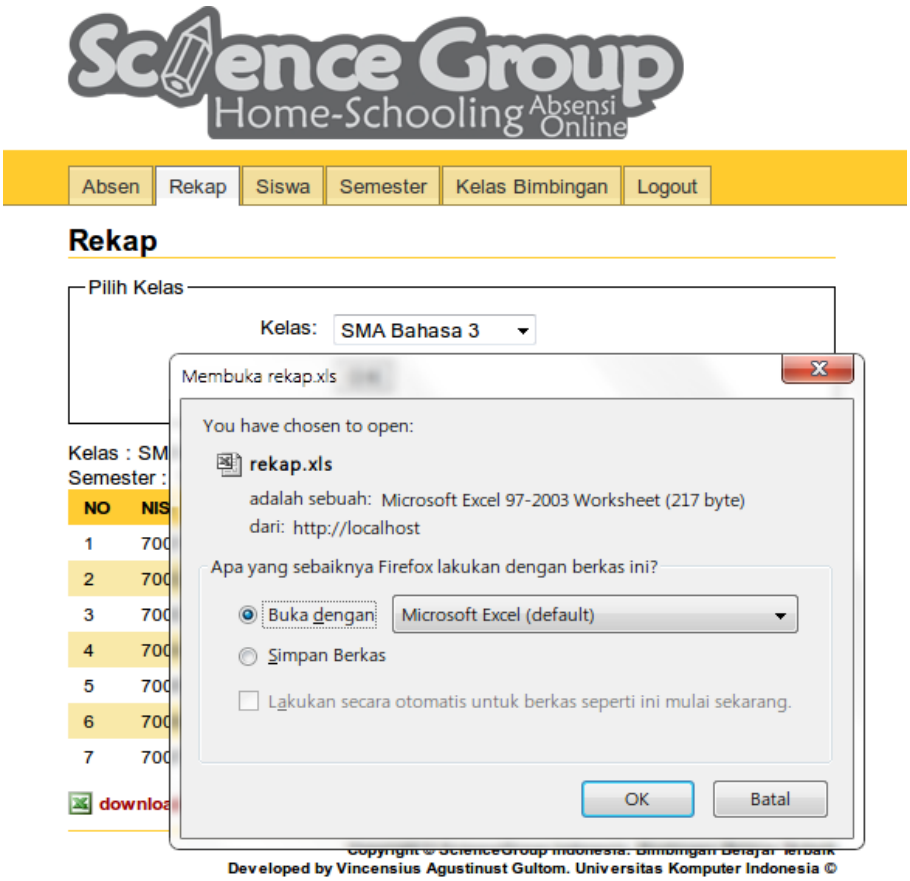

Gambar 3.47 Tampilan menu pemilihan semester.

15. Tampilan menu rekap.



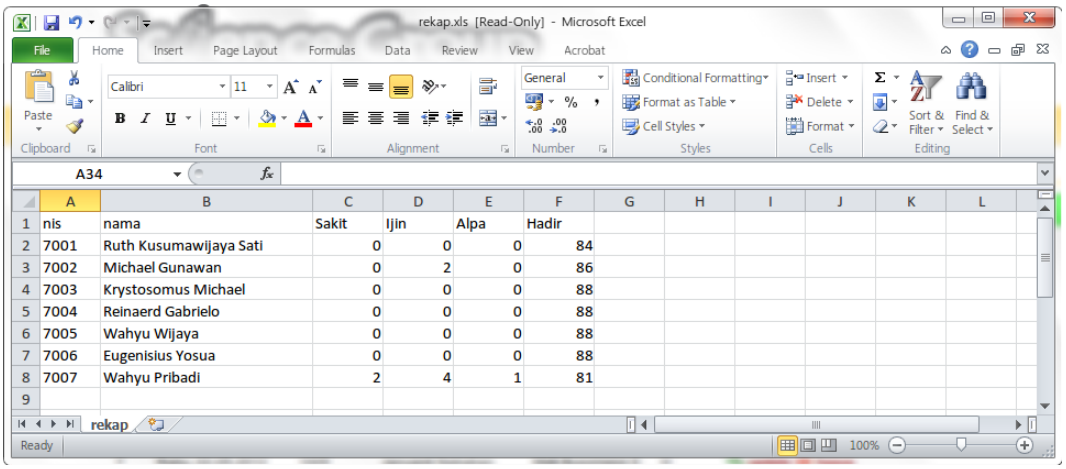

Gambar 3.48 Tampilan menu rekap.

16. Tampilan download rekap.




Gambar 3.49 Tampilan download rekap

17. Tampilan file rekap absensi.




| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | nis | nama | Sakit | Ijin | Alpa | Hadir |
| 2 | 7001 | Ruth Kusumawijaya Sati | 0 | 0 | 0 | 84 |
| 3 | 7002 | Michael Gunawan | 0 | 2 | 0 | 86 |
| 4 | 7003 | Krystosomus Michael | 0 | 0 | 0 | 88 |
| 5 | 7004 | Reinaerd Gabrielo | 0 | 0 | 0 | 88 |
| 6 | 7005 | Wahyu Wijaya | 0 | 0 | 0 | 88 |
| 7 | 7006 | Eugenisius Yosua | 0 | 0 | 0 | 88 |
| 8 | 7007 | Wahyu Pribadi | 2 | 4 | 1 | 81 |
| 9 | | | | | | |

Gambar 3.50 Tampilan file rekap absensi.