

**KONSTRUKSI SISTEM INFORMASI MANAJEMEN AKADEMIK  
BERBASIS WEB DENGAN MENERAPKAN *PATTERN MODEL-  
VIEW-CONTROLLER (MVC)***

**Studi Kasus Di: Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas  
Malang**

**SKRIPSI**

Disusun oleh:  
Dimas Habib R  
NIM: 125150401111049



**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2017**

## PENGESAHAN

KONSTRUKSI SISTEM INFORMASI MANAJEMEN AKADEMIK BERBASIS *WEB*  
DENGAN MENERAPKAN *PATTERN MODEL-VIEW-CONTROLLER (MVC)*  
STUDI KASUS DI: MADRASAH TSANAWIYAH MUHAMMADIYAH 1 TLOGOMAS  
MALANG

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Dimas Habib R

NIM: 125150401111049

Skripsi ini telah diuji dan dinyatakan lulus pada  
12 Januari 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Satrio Agung W., S.Kom, M.Kom

NIP: 198605212012121001

Retno Indah Rokhmawati., S.Pd., M.Pd.

NIK: 2016099009172001

Mengetahui

Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T., M.T

NIP: 19740823 200012 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Januari 2017



DIMAS HABIB R

NIM: 125150401111049

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan, karena dengan rahmat, dan karunia-Nya penulis dapat menyelesaikan skripsi dengan judul **“Konstruksi Sistem Informasi Manajemen Akademik Berbasis Web Dengan Menerapkan Pattern Model-View-Controller (MVC) Studi Kasus Di: Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang”**. Skripsi ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Komputer di Program Studi Sistem Informasi, Fakultas Ilmu Komputer Universitas Brawijaya. Melalui pengantar ini penulis mengucapkan banyak terima kasih karena dalam penulisan dan penyusunan laporan skripsi ini penulis mendapat bantuan dan dorongan baik secara langsung maupun tidak langsung dari berbagai pihak diantaranya:

1. Bapak Satrio Agung W., S.Kom, M.Kom., selaku dosen pembimbing I dan Ibu Retno Indah Rokhmawati, S.Pd., M.Pd., selaku dosen pembimbing II yang telah sabar memberikan bimbingan, arahan, ilmu dan saran untuk laporan skripsi ini.
2. Bapak Abdul Soleh dan Ibu Elly Viana Komariyatin selaku orang tua, atas segala dukungan, kasih sayang, perhatian dan kesabarannya, serta yang senantiasa selalu memberikan doa dan semangat demi terselesaikannya skripsi ini.
3. Bapak dan Ibu dosen serta seluruh staf Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya yang telah memberikan ilmunya serta arahan selama masa perkuliahan.
4. Davi, Dio, Mades, Izzul, Cicha, Voni, Faishal dan teman-teman seperjuangan FILKOM angkatan 2012 yang tidak bisa disebutkan satu persatu yang ikut mendukung dan menyemangati selama proses pengerjaan ini.
5. Seluruh Keluarga Besar Fakultas Ilmu Komputer yang telah memberikan tempat untuk kuliah dan mencari pengalaman di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini. Semoga segala jasa dan amal baik mendapatkan balasan dari Tuhan YME.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan tidak terlepas dari kekurangan dan kesalahan karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Oleh karena itu, segala kritik dan saran yang bersifat membangun penulis harapkan untuk penyempurnaan skripsi ini. Penulis juga berharap mudah-mudahan skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 23 Januari 2017

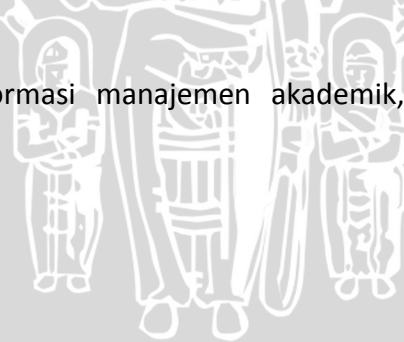
Dimas Habib R

[dimas.habib01@gmail.com](mailto:dimas.habib01@gmail.com)

## ABSTRAK

Perkembangan teknologi sudah dapat dirasakan dalam aspek apapun di kehidupan sehari-hari, tidak terkecuali di bidang pendidikan. Madrasah Tsanawiyah Muhammadiyah 1 (MTsM 1) telah memiliki *website* yang hanya berisi tentang informasi sekolah tersebut. Sedangkan, untuk kegiatan administrasi yang penting seperti memajemen jadwal mata pelajaran dan merekam presensi secara detail masih dilakukan secara konvensional (tulisan tangan) yang sangatlah rumit dan melelahkan untuk membuat jadwal mata pelajaran. Perekaman presensi yang ada juga tidak efektif karena ketika data tersebut dibutuhkan maka harus mencari dokumen satu persatu. Berdasarkan kondisi diatas dibutuhkan sistem informasi manajemen akademik yang mampu memajemen jadwal mata pelajaran dan merekam presensi secara detail yang seluruh prosesnya dilakukan secara online. Tujuan penelitian ini adalah untuk mengimplementasikan hasil analisis dan perancangan dari penelitian sebelumnya menjadi sistem informasi dan menguji sistem informasi manajemen akademik. Pengimplementasian atau kontruksi sistem dilakukan dengan menerapkan *pattern Model-View-Controller (MVC)*. Pengujian terhadap fungsionalitas sistem dilakukan untuk menunjukkan bahwa fungsi sistem telah sesuai dengan spesifikasinya dan pengujian terhadap komabilitas dilakukan untuk menunjukkan bahwa sistem dapat berjalan dengan baik pada browser *Internet Explorer, Microsoft Edge, Mozilla Firefox, Google Chrome, Safari, Opera, browser iOS, dan browser Android*.

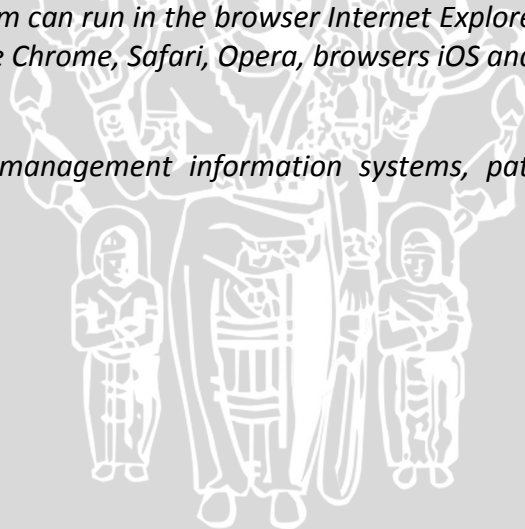
Kata kunci: Sistem informasi manajemen akademik, *pattern Model-View-Controller (MVC)*.



## ABSTRACT

*Technological developments can already be perceived in any aspect of daily life, without exception in the field of education. MTs Muhammadiyah 1 (MTsM 1) already has a website only contains information about the school. Whereas, for the important administrative activities such as managing the schedule and record the presence in detail is still done in conventional (handwriting) are very complicated and laborious to create a schedule of subjects. The existing record of presence is also ineffective because when the data is required, it must looking for documents one by one. Based on the above conditions required academic information management system that capable for managing the schedule and record the presence in detail, and the entire process is done online. The purpose of this study is to implement the results of the analysis and design of previous research into information systems and test its systems. Implementation or the construction of the system is done by applying pattern Model-View-Controller (MVC). Tests on the functionality of the system is done to show that the system functions in accordance with the specification and testing of the compatibility done to show that the system can run in the browser Internet Explorer, Microsoft Edge, Mozilla Firefox, Google Chrome, Safari, Opera, browsers iOS and browser Android.*

*Keywords: academic management information systems, pattern Model-View-Controller (MVC).*



## DAFTAR ISI

|   |           |
|---|-----------|
| PENGESAHAN .....  | ii        |
| PERNYATAAN ORISINALITAS .....                           | iii       |
| KATA PENGANTAR.....                                     | iv        |
| ABSTRAK.....  | v         |
| ABSTRACT.....   | vi        |
| DAFTAR ISI.....   | vii       |
| DAFTAR TABEL.....                                       | x         |
| DAFTAR GAMBAR.....                                      | xii       |
| <b>BAB 1 PENDAHULUAN.....</b>                           | <b>1</b>  |
| 1.1 Latar Belakang.....                                 | 1         |
| 1.2 Rumusan Masalah.....                                | 2         |
| 1.3 Tujuan .....  | 3         |
| 1.4 Manfaat.....  | 3         |
| 1.5 Batasan Masalah.....                                | 4         |
| 1.6 Sistematika Pembahasan.....                         | 4         |
| <b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>                 | <b>5</b>  |
| 2.1 Penelitian Sebelumnya .....                         | 5         |
| 2.2 Sistem informasi .....                              | 6         |
| 2.3 Sistem Informasi Manajemen.....                     | 6         |
| 2.4 Sistem Informasi Manajemen Akademik.....            | 6         |
| 2.5 <i>Software Development Life Cycle (SDLC)</i> ..... | 7         |
| 2.5.1 Evaluasi.....                                     | 8         |
| 2.5.2 <i>Code</i> .....                                 | 10        |
| 2.5.3 <i>Test</i> .....                                 | 16        |
| <b>BAB 3 METODOLOGI .....</b>                           | <b>18</b> |
| 3.1 Alur Penelitian.....                                | 18        |
| 3.1.1 Studi Literatur .....                             | 19        |
| 3.1.2 Studi Rancangan yang Sudah Ada.....               | 19        |
| 3.1.3 Evaluasi Rancangan.....                           | 19        |

|   |           |
|---|-----------|
| 3.1.4 Perubahan Rancangan .....                             | 19        |
| 3.1.5 Implementasi .....                                    | 19        |
| 3.1.6 Pengujian.....  | 20        |
| 3.1.7 Kesimpulan dan Saran.....                             | 20        |
| <b>BAB 4 EVALUASI RANCANGAN.....</b>                        | <b>21</b> |
| 4.1 Evaluasi Rancangan.....                                 | 21        |
| 4.1.1 Manajemen Jadwal Mata Pelajaran.....                  | 21        |
| 4.1.2 Manajemen Presensi Siswa.....                         | 27        |
| 4.1.3 Rekap Presensi Siswa .....                            | 31        |
| 4.2 Evaluasi Antarmuka .....                                | 33        |
| <b>BAB 5 PERANCANGAN.....</b>                               | <b>35</b> |
| 5.1 Perubahan Perancangan ke Implementasi.....              | 35        |
| 5.2 <i>Class Diagram</i> .....                              | 35        |
| 5.2.1 <i>Class Diagram</i> Perancangan.....                 | 35        |
| 5.2.2 <i>Class Diagram</i> Implementasi.....                | 41        |
| 5.2.3 Perubahan <i>Class Diagram</i> yang Terjadi.....      | 43        |
| <b>BAB 6 IMPLEMENTASI .....</b>                             | <b>44</b> |
| 6.1 Lingkungan Implementasi.....                            | 44        |
| 6.1.1 Lingkungan Perangkat Keras ( <i>Hardware</i> ).....   | 44        |
| 6.1.2 Lingkungan Perangkat Lunak ( <i>Software</i> ): ..... | 44        |
| 6.2 Hasil Implementasi .....                                | 44        |
| 6.2.1 Kode Program.....                                     | 44        |
| 6.2.2 <i>Screenshot</i> .....                               | 55        |
| <b>BAB 7 PENGUJIAN .....</b>                                | <b>86</b> |
| 7.1 Rencana Pengujian.....                                  | 86        |
| 7.2 Pengujian <i>Black-Box</i> .....                        | 88        |
| 7.2.1 Pengujian Identifikasi Pengguna .....                 | 88        |
| 7.2.2 Pengujian Manajemen Data Guru.....                    | 89        |
| 7.2.3 Pengujian Manajemen <i>User</i> .....                 | 91        |
| 7.2.4 Pengujian Manajemen Data Siswa.....                   | 92        |
| 7.2.5 Pengujian Manajemen Data Mata Pelajaran .....         | 94        |
| 7.2.6 Pengujian Manajemen Jam Pelajaran.....                | 96        |





|   |            |
|---|------------|
| 7.2.7 Pengujian Manajemen Data Kelas .....            | 97         |
| 7.2.8 Pengujian Manajemen Data Pengajar.....          | 99         |
| 7.2.9 Pengujian Manajemen Pembagian Kelas.....        | 100        |
| 7.2.10 Pengujian Manajemen Kalender Akademik.....     | 102        |
| 7.2.11 Pengujian Manajemen Jadwal Mata Pelajaran..... | 103        |
| 7.2.12 Pengujian Manajemen Kegiatan .....             | 105        |
| 7.2.13 Pengujian Manajemen Presensi Siswa.....        | 107        |
| 7.2.14 Pengujian Manajemen Wali Kelas.....            | 107        |
| 7.2.15 Pengujian Manajemen Rekapian .....             | 109        |
| 7.2.16 Pengujian Mencetak Laporan .....               | 109        |
| 7.3 Pengujian <i>Compatibility</i> .....              | 110        |
| 7.4 Analisis Hasil Pengujian.....                     | 110        |
| <b>BAB 8 PENUTUP .....</b>                            | <b>111</b> |
| 8.1 Kesimpulan.....                                   | 111        |
| 8.2 Saran .....                                       | 112        |
| <b>DAFTAR PUSTAKA.....</b>                            | <b>113</b> |



## DAFTAR TABEL

|  |     |
|--|-----|
| Tabel 4.1 Kesesuaian Alur Pengguna <i>Use Case</i> Manajemen Jadwal Mata Pelajaran ..... | 21  |
| Tabel 4.2 Harapan Alur Pengguna <i>Use Case</i> Manajemen Jadwal Mata Pelajaran          | 26  |
| Tabel 4.3 Tinjauan Navigasi <i>Use Case</i> Manajemen Jadwal Mata Pelajaran .....        | 26  |
| Tabel 4.4 Kesesuaian Alur Pengguna <i>Use Case</i> Manajemen Presensi Siswa .....        | 27  |
| Tabel 4.5 Harapan Alur Pengguna <i>Use Case</i> Manajemen Presensi Siswa .....           | 29  |
| Tabel 4.6 Tinjauan Navigasi <i>Use Case</i> Manajemen Presensi Siswa .....               | 30  |
| Tabel 4.7 Kesesuaian Alur Pengguna <i>Use Case</i> Rekap Presensi Siswa .....            | 31  |
| Tabel 4.8 Harapan Alur Pengguna <i>Use Case</i> Rekap Presensi Siswa .....               | 32  |
| Tabel 4.9 Tinjauan Navigasi <i>Use Case</i> Rekap Presensi Siswa .....                   | 32  |
| Tabel 4.10 Evaluasi Antarmuka Pengguna Alur Manajemen Jadwal Mata Pelajaran .....        | 33  |
| Tabel 5.1 Penjelasan <i>class controller</i> diagram perancangan (Fatha, 2017).....      | 36  |
| Tabel 5.2 Penjelasan <i>class model</i> diagram perancangan (Fatha, 2017) .....          | 39  |
| Tabel 6.1 Kode program <i>class</i> jadwal.....  | 45  |
| Tabel 6.2 Kode program <i>class</i> m_jadwal .....                                       | 51  |
| Tabel 7.1 Kebutuhan atau persyaratan fungsional yang akan diuji (Fatha, 2017)            | 86  |
| Tabel 7.2 Hasil pengujian Identifikasi Pengguna.....                                     | 88  |
| Tabel 7.3 Hasil pengujian manajemen data guru .....                                      | 89  |
| Tabel 7.4 Hasil pengujian manajemen <i>user</i> .....                                    | 91  |
| Tabel 7.5 Hasil pengujian manajemen data siswa .....                                     | 92  |
| Tabel 7.6 Hasil pengujian manajemen data mata pelajaran .....                            | 94  |
| Tabel 7.7 Hasil pengujian manajemen jam pelajaran .....                                  | 96  |
| Tabel 7.8 Hasil pengujian data kelas .....   | 97  |
| Tabel 7.9 Hasil pengujian manajemen data pengajar.....                                   | 99  |
| Tabel 7.10 Hasil pengujian manajemen pembagian kelas.....                                | 101 |
| Tabel 7.11 Hasil pengujian manajemen kalender akademik .....                             | 102 |
| Tabel 7.12 Hasil pengujian manajemen jadwal mata pelajaran .....                         | 104 |
| Tabel 7.13 Hasil pengujian manajemen kegiatan .....                                      | 105 |
| Tabel 7.14 Hasil pengujian manajemen presensi siswa.....                                 | 107 |

|   |     |
|---|-----|
| Tabel 7.15 Hasil pengujian manajemen wali kelas ..... | 108 |
| Tabel 7.16 Hasil pengujian manajemen rekapan.....     | 109 |
| Tabel 7.17 Hasil pengujian mencetak laporan.....      | 109 |



## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Tahapan SDLC (A.S & Shalahuddin, 2013).....  | 7  |
| Gambar 2.2 Komunikasi <i>user</i> , <i>browser</i> dan <i>web server</i> (Sunyoto, 2007)..... | 11 |
| Gambar 2.3 Ilustrasi <i>design pattern MVC</i> (Basuki, 2014) .....                           | 13 |
| Gambar 2.4 Struktur File CI (Candrawati, 2014) .....  | 14 |
| Gambar 2.5 Penulisan kode program <i>class controller</i> .....                               | 15 |
| Gambar 2.6 Penulisan kode program <i>class model</i> .....                                    | 15 |
| Gambar 3.1 Alur Penelitian .....  | 18 |
| Gambar 5.1 <i>Class controller</i> diagram perancangan (Fatha, 2017) .....                    | 36 |
| Gambar 5.2 <i>Class model</i> diagram perancangan (Fatha, 2017) .....                         | 38 |
| Gambar 5.3 <i>Class</i> diagram gabungan <i>MVC</i> perancangan (Fatha, 2017) .....           | 40 |
| Gambar 5.4 <i>Class</i> diagram <i>controller</i> implementasi.....                           | 41 |
| Gambar 5.5 <i>Class</i> diagram <i>model</i> implementasi .....                               | 42 |
| Gambar 5.6 <i>Class</i> diagram gabungan <i>MVC</i> implementasi.....                         | 43 |
| Gambar 6.1 Halaman <i>Login</i> hasil implemetasi .....                                       | 56 |
| Gambar 6.2 Halaman menambah data guru hasil implemetasi .....                                 | 56 |
| Gambar 6.3 Halaman mengubah detail guru hasil implemetasi .....                               | 57 |
| Gambar 6.4 Halaman mengubah foto guru hasil implemetasi.....                                  | 57 |
| Gambar 6.5 Halaman menghapus data guru hasil implemetasi.....                                 | 58 |
| Gambar 6.6 Halaman melihat detail guru hasil implemetasi .....                                | 58 |
| Gambar 6.7 Halaman menambah <i>user</i> hasil implemetasi.....                                | 59 |
| Gambar 6.8 Halaman mengubah <i>user</i> hasil implemetasi .....                               | 59 |
| Gambar 6.9 Halaman menghapus <i>user</i> hasil implemetasi .....                              | 60 |
| Gambar 6.10 Halaman menambah data siswa hasil implemetasi.....                                | 60 |
| Gambar 6.11 Halaman mengubah detail siswa hasil implemetasi .....                             | 61 |
| Gambar 6.12 Halaman mengubah foto siswa hasil implemetasi .....                               | 61 |
| Gambar 6.13 Halaman menghapus data siswa hasil implemetasi .....                              | 62 |
| Gambar 6.14 Halaman melihat data siswa hasil implemetasi.....                                 | 62 |
| Gambar 6.15 Halaman menambah data mata pelajaran hasil implemetasi .....                      | 63 |
| Gambar 6.16 Halaman mengubah data mata pelajaran hasil implemetasi .....                      | 63 |
| Gambar 6.17 Halaman menghapus data mata pelajaran hasil implemetasi.....                      | 64 |

|   |    |
|---|----|
| Gambar 6.18 Halaman menambah data jam pelajaran hasil implemetasi .....                                       | 64 |
| Gambar 6.19 Halaman mengubah data jam pelajaran hasil implemetasi .....                                       | 65 |
| Gambar 6.20 Halaman menghapus data jam pelajaran hasil implemetasi .....                                      | 65 |
| Gambar 6.21 Halaman menambah data kelas hasil implemetasi .....   | 66 |
| Gambar 6.22 Halaman mengubah data kelas hasil implemetasi .....   | 66 |
| Gambar 6.23 Halaman menghapus data kelas hasil implemetasi .....  | 67 |
| Gambar 6.24 Halaman melihat data kelas hasil implemetasi .....  | 67 |
| Gambar 6.25 Halaman menambah data pengajar hasil implemetasi .....  | 68 |
| Gambar 6.26 Halaman mengubah data pengajar hasil implemetasi .....  | 68 |
| Gambar 6.27 Halaman menghapus data pengajar hasil implemetasi .....   | 69 |
| Gambar 6.28 Halaman menambah pembagian kelas hasil implemetasi .....  | 69 |
| Gambar 6.29 Halaman mengubah pembagian kelas hasil implemetasi .....  | 70 |
| Gambar 6.30 Halaman menghapus pembagian kelas hasil implemetasi .....   | 70 |
| Gambar 6.31 Antarmuka menambah kalender akademik hasil implemetasi .....                                      | 71 |
| Gambar 6.32 Halaman mengubah kalender akademik hasil implemetasi .....  | 71 |
| Gambar 6.33 Halaman menghapus kalender akademik hasil implemetasi .....                                       | 72 |
| Gambar 6.34 Halaman melihat kalender akademik hasil implemetasi <i>user</i> tata usaha .....                  | 72 |
| Gambar 6.35 Halaman melihat kalender akademik hasil implemetasi <i>user</i> guru dan kepala sekolah .....     | 73 |
| Gambar 6.36 Halaman menambah jadwal mata pelajaran hasil implemetasi .....                                    | 73 |
| Gambar 6.37 Halaman mengubah jadwal mata pelajaran hasil implemetasi .....                                    | 74 |
| Gambar 6.38 Halaman menghapus jadwal mata pelajaran hasil implemetasi .....                                   | 74 |
| Gambar 6.39 Halaman melihat jadwal mata pelajaran hasil implemetasi <i>user</i> tata usaha .....              | 75 |
| Gambar 6.40 Halaman melihat jadwal mata pelajaran hasil implemetasi <i>user</i> guru dan kepala sekolah ..... | 75 |
| Gambar 6.41 Halaman menambah kegiatan hasil implemetasi .....   | 76 |
| Gambar 6.42 Halaman mengubah kegiatan hasil implemetasi .....   | 77 |
| Gambar 6.43 Halaman menghapus kegiatan hasil implemetasi .....  | 77 |
| Gambar 6.44 Halaman melihat kegiatan hasil implemetasi <i>user</i> tata usaha .....                           | 78 |
| Gambar 6.45 Halaman melihat kegiatan hasil implemetasi <i>user</i> guru dan kepala sekolah .....              | 78 |



Gambar 6.46 Halaman menambah presensi siswa hasil implemetasi ..... 79

Gambar 6.47 Halaman mengubah presensi siswa hasil implemetasi ..... 80

Gambar 6.48 Halaman menghapus presensi siswa hasil implemetasi..... 80

Gambar 6.49 Halaman melihat presensi siswa hasil implemetasi *user* tata usaha ..... 81

Gambar 6.50 Halaman melihat presensi siswa hasil implemetasi *user* guru dan kepala sekolah..... 81

Gambar 6.51 Halaman menambah wali kelas hasil implemetasi ..... 82

Gambar 6.52 Halaman mengubah wali kelas hasil implemetasi ..... 82

Gambar 6.53 Halaman menghapus wali kelas hasil implemetasi..... 83

Gambar 6.54 Halaman melihat wali kelas hasil implemetasi ..... 83

Gambar 6.55 Halaman membuat rekapan hasil implemetasi ..... 84

Gambar 6.56 Halaman mencetak rekapan hasil implemetasi ..... 85

Gambar 7.1 Hasil pengujian *Compatibility* menggunakan *SortSite 5 Trial* ..... 110



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Pemakaian teknologi komputer dan internet dalam segala bidang kehidupan sehari-hari tidak akan dapat dihindari. Perkembangan demi perkembanganpun dapat dirasakan dalam berbagai aspek kehidupan sehingga kegiatan manusia pada saat ini sering dipengaruhi oleh teknologi informasi tidak terkecuali di bidang pendidikan, khususnya di instansi pendidikan seperti sekolah.

Madrasah Tsanawiyah Muhammadiyah 1 merupakan instansi pendidikan swasta yang berlokasi di Tlogomas Kota Malang yang selalu ingin meningkatkan mutu baik dalam hal prestasi maupun pelayanan dan telah memiliki *website* resmi yang berisi tentang informasi sekolah. Akan tetapi, dalam kegiatan administrasinya, instansi tersebut belum didukung oleh teknologi informasi layaknya sistem informasi akademik yang dapat membantu kegiatan kerja akademik agar lebih efektif dan efisien sehingga dapat memunculkan beberapa permasalahan.

Berdasarkan hasil wawancara yang dilakukan oleh Fatha (2017) dengan pihak Madrasah Madrasah Tsanawiyah Muhammadiyah 1, dapat diketahui permasalahan yang muncul di lapangan diantaranya pertama, bagi pihak tata usaha, proses pembuatan jadwal mata pelajaran dengan jumlah 19 guru, 17 mata pelajaran dan 9 kelas sulit dilakukan secara konvensional (pembuatan jadwal menggunakan tulisan tangan). Proses pembuatannya sangatlah rumit dan beresiko mengalami kesalahan, sehingga proses pembuatan jadwal harus dimulai dari awal lagi dan membutuhkan waktu yang relatif lama. Kedua, permasalahan bagi wali kelas adalah presensi yang ada sekarang masih dihitung harian bukan per-jam pelajaran serta belum detail sehingga menyulitkan wali kelas untuk mengambil keputusan tentang siswa yang naik atau tidak karena bisa jadi siswa sering membolos di pertengahan kegiatan belajar mengajar lalu kembali ke kelas di saat kelas sudah akan berakhir. Ketiga, orang tua siswa biasanya kurang mengetahui kegiatan anaknya di sekolah, apakah anaknya benar-benar pergi ke sekolah atau sering membolos di tengah proses belajar mengajar atau bahkan membolos seharian penuh, dan sebagainya. Dengan adanya rekap presensi yang detail di setiap jam pelajarannya dapat dijadikan sebuah informasi untuk orang tua siswa tentang kegiatan anaknya di sekolah.

Penjadwalan merupakan kegiatan administrasi yang sangat penting bagi instansi pendidikan karena kegiatan sehari-hari yang dilakukan oleh instansi berdasarkan apa yang telah diatur di dalam jadwal. Pembuatan jadwal juga harus sangat teliti karena tidak boleh ada waktu kerja atau mengajar yang berbenturan misalnya ada seorang guru yang memiliki 2 jadwal mengajar atau lebih pada waktu yang sama dalam satu hari.

Adapun, penelitian ini bertujuan untuk membantu menyelesaikan permasalahan yang ada dalam kegiatan kerja akademik yaitu membangun sistem yang bisa membuat kegiatan kerja akademik lebih efektif dan efisien. Sistem yang dimaksud adalah sistem informasi yang mampu memanajemen jadwal mata pelajaran yang mana setiap proses transaksi (*insert* dan *update*) datanya dilakukan pengecekan identitas data sehingga data jadwal menjadi akurat atau tidak ada guru yang memiliki 2 jadwal mengajar atau lebih pada waktu yang sama dalam satu hari serta sistem yang mampu merekam presensi siswa secara detail yakni dengan merekam presensi siswa di setiap jam pelajaran berdasarkan jadwal yang telah ditentukan.

Dalam membangun sistem informasi dibutuhkan sebuah analisis kebutuhan untuk menentukan segala kebutuhan pihak terkait sebagai pemakai (pengguna) sistem dan perancangan untuk merancang bagaimana sistem akan dibangun. Sebelumnya sudah pernah ada penelitian yang berjudul “Analisis dan Perancangan Sistem Informasi Manajemen Akademik” oleh Izzul Fatha selaku mahasiswa Sistem Informasi Universitas Brawijaya pada tahun 2017 yang mana pada penelitian tersebut membahas tentang analisis kebutuhan, proses bisnis dan perancangan sistem (Fatha, 2017).

Dikarenakan penelitian tentang analisis dan perancangan sistem dilakukan tanpa implementasi, maka peneliti berniat untuk mengimplementasikan analisis dan perancangan yang sudah dilakukan menjadi sebuah sistem. Sistem ini akan dibangun berbasis *web* dan menggunakan *pattern Model-View-Controller* atau yang umumnya disebut dengan MVC, karena menurut Pop & Altar (2014), *pattern MVC* merupakan pola yang cocok untuk digunakan sebagai pengembangan aplikasi web.

MVC adalah sebuah konsep yang digunakan untuk membangun aplikasi dengan memisahkan antara *logic (Controller)* dengan tampilan (*View*) dan basis data (*Model*) (Candrawati, 2014). Dengan adanya pemisahan tersebut, dapat menjadikan aplikasi lebih fleksibel karena jika diperlukan adanya modifikasi pada salah satu komponen hanya akan berdampak minimum pada komponen lainnya (Budiharto, 2014). Konsep ini juga dapat membuat *coding logic* menjadi lebih sederhana dan memungkinkan *developer* yang memiliki keahlian berbeda untuk berkolaborasi dengan berfokus pada keterampilan inti masing-masing.

Penelitian ini dilakukan dengan harapan agar sistem informasi yang dibangun dapat mempermudah dalam mengelola serta mengontrol kegiatan belajar dan mengajar sehingga dapat tersistemasi dengan baik.

## 1.2 Rumusan Masalah

1. Bagaimana mengevaluasi hasil penelitian Analisa dan Perancangan Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang?



2. Bagaimana mengimplementasikan rancangan dari penelitian Analisa dan Perancangan Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang?
3. Bagaimana menguji Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang?
4. Bagaimana hasil pengujian Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang?

### 1.3 Tujuan

Tujuan penelitian ini adalah:

1. Mengevaluasi hasil penelitian Analisa dan Perancangan Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang agar bisa diimplementasikan menjadi sistem informasi.
2. Mengimplementasikan rancangan dari penelitian Analisa dan Perancangan Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang.
3. Menguji Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang.
4. Menganalisis hasil pengujian Sistem Informasi Manajemen Akademik Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang.

### 1.4 Manfaat

Manfaat penelitian ini adalah:

a. Bagi Peneliti:

Dapat lebih memahami tentang pengembangan aplikasi web menggunakan *pattern* MVC dan dapat lebih memahami cara untuk menemukan solusi dalam memecahkan suatu permasalahan yang ada di dunia nyata kedalam teknologi informasi.

b. Bagi Tatausaha:

Dapat memudahkan dalam memajemen jadwal mata pelajaran yang akurat serta dalam merekam kehadiran siswa dengan lebih detail.

c. Bagi Wali Kelas:

Dapat menjadi pertimbangan dalam pembuatan keputusan atas naik atau tidaknya seorang siswa yang menjadi bagian dari tanggung jawabnya.

d. Bagi Madrasah Tsanawiyah Muhammadiyah 1

Dapat meningkatkan keefektifan dan keefisienan dalam kegiatan yang ada pada sekolah.

## 1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Sistem informasi manajemen akademik ini difokuskan untuk menangani manajemen jadwal mata pelajaran yang akurat dan merekam kehadiran siswa secara detail.
- b. Yang dapat mengakses sistem informasi manajemen akademik ini adalah pihak-pihak yang berkaitan dengan kegiatan akademik antara lain, kepala sekolah, tata usaha dan guru.

## 1.6 Sistematika Pembahasan

Sistematika penulisan dalam skripsi ini sebagai berikut:

### **BAB I Pendahuluan**

Bab ini adalah bab yang menguraikan latar belakang, rumusan masalah, tujuan dan manfaat skripsi serta sistematika penyusunan laporan skripsi.

### **BAB II Landasan Kepustakaan**

Bab ini adalah bab yang menguraikan dasar teori yang menjadi referensi dalam pelaksanaan skripsi.

### **BAB III Metodologi**

Bab ini adalah bab yang menguraikan tentang alur pengerjaan skripsi.

### **BAB IV Evaluasi**

Bab IV menguraikan tentang evaluasi alur pengguna, navigasi dan antarmuka prototipe sistem informasi.

### **BAB V Perancangan**

Bab V menguraikan tentang penyesuaian perancangan sistem yang telah dilakukan pada penelitian Fatha (2017) menjadi perancangan yang dapat diimplementasikan menjadi sistem.

### **BAB VI Implementasi**

Bab VI menguraikan tentang hasil implementasi dari perancangan yang berupa kode program dan *screenshot* antarmuka hasil implementasi.

### **BAB VII Pengujian**

Bab VII menguraikan tentang pengujian terhadap fitur-fitur yang telah didefinisikan dalam kebutuhan fungsional dan non-fungsional serta analisis hasil pengujian.

### **BAB VIII Penutup**

Bab VIII menguraikan tentang kesimpulan dari penelitian serta saran penelitian yang harusnya dilakukan kedepannya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Penelitian Sebelumnya

Penelitian dengan judul “Analisis dan Perancangan Sistem Informasi Manajemen Akademik” yang dilakukan oleh Izzul Fatha selaku mahasiswa jurusan Sistem Informasi dari Fakultas Ilmu Komputer Universitas Brawijaya di tahun 2017 pada instansi pendidikan yang terletak di Tlogomas Kota Malang yakni Madrasah Tsanawiyah Muhammadiyah 1 merupakan penelitian yang dijadikan rujukan utama oleh peneliti karena penelitian tersebut telah mencakup tentang analisis proses bisnis, analisis persyaratan, perancangan dan evaluasi yang mana hal tersebut sangat dibutuhkan oleh peneliti dalam pengerjaan skripsi ini.

Analisis proses bisnis dilakukan terhadap alur proses bisnis pembuatan jadwal mata pelajaran dan perekapan presensi siswa yang masih dilakukan secara manual hingga alur proses bisnis yang diusulkan dengan menggunakan sistem informasi. Pemodelan proses bisnis pada penelitian diatas menggunakan *Business Process Model and Notation* atau yang biasa disebut dengan BPMN.

Analisis persyaratan dilakukan dengan menggunakan metode *Object Oriented Analysis Design (OOAD)* dimana pada analisis ini dibagi menjadi 2 komponen, yaitu:

1. Analisis kebutuhan pengguna dan pemangku kepentingan, dimana peneliti mengidentifikasi permasalahan yang harus diselesaikan dengan menggunakan sistem serta menjelaskan tentang alasan mengapa persyaratan sistem dibutuhkan oleh pemangku kepentingan.
2. Analisis fitur, dimana peneliti melakukan analisis terhadap kemampuan Sistem Informasi Manajemen Akademik yang akan dibangun secara umum.

Perancangan sistem dilakukan dengan pendekatan berorientasi objek. Pemodelan perancangan menggunakan diagram *Unified Modelling Language (UML)* dimana terdapat 4 komponen dalam perancangan, yaitu:

1. Rancangan Arsitektur, dimana peneliti melakukan perancangan arsitektur sistem berdasarkan hasil analisis kebutuhan sistem. Pada perancangan arsitektur ini, peneliti juga melakukan perancangan *subsystem* dan *package*.
2. Rancangan *class*, dimana rancangan ini menggambarkan tentang sistem dan relasi antar kelas seperti pewarisan dan ketergantungan antar komponen pada *class diagram*.
3. Rancangan data model yang dimodelkan dalam bentuk *physical data model diagram*.

4. Rancangan Antarmuka Pengguna dimana Antarmuka Pengguna berfungsi sebagai sarana berinteraksi antara pengguna dengan sistem.

Evaluasi dilakukan dengan tujuan untuk mengetahui kualitas perancangan yang sudah dilakukan. Pengujian dilakukan terhadap proses bisnis apakah sesuai atau tidak, serta *review* dari sisi pengguna terkait persyaratan dan hasil rancangan antarmuka pengguna sistem. Peneliti juga menggunakan matriks keruntutan perancangan sistem untuk melacak perkembangan kebutuhan dari awal hingga akhir.

## 2.2 Sistem informasi

Menurut Ardiyanto (2011), sistem informasi merupakan kumpulan dari komponen pembentuk sistem yang mana komponen-komponen tersebut mempunyai keterkaitan yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu.

Menurut Kadir (2003), sistem informasi berbasis komputer (*Computer based information system*) terdiri dari komponen-komponen sebagai berikut:

- a. Perangkat keras (*hardware*), perangkat yang terdiri dari piranti-piranti fisik seperti komputer dan printer.
- b. Perangkat lunak (*software*), merupakan program atau sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
- c. Prosedur, merupakan sekumpulan aturan yang digunakan untuk mewujudkan pemrosesan data dan keluaran yang dikehendaki.
- d. Manusia (*human*), merupakan semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
- e. Basis data (*database*), merupakan sekumpulan tabel yang berhubungan dengan penyimpanan data.
- f. Jaringan komputer dan komunikasi data, merupakan sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

## 2.3 Sistem Informasi Manajemen

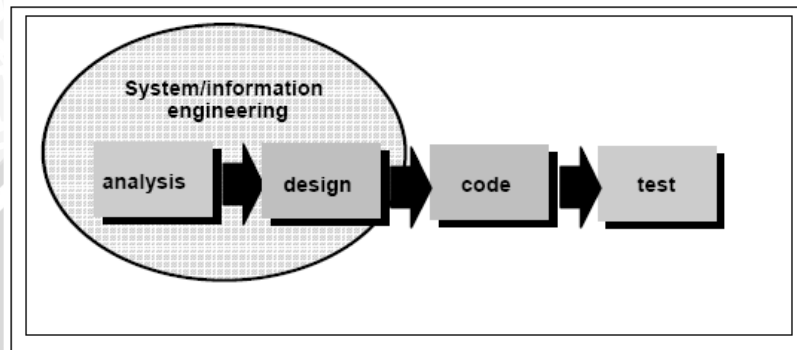
Istilah sistem informasi sering disebut dengan sistem informasi manajemen, padahal kedua istilah tersebut sebenarnya tidak sama seperti penuturan Kadir (2003) bahwa sistem informasi manajemen adalah sistem informasi yang digunakan untuk menyajikan informasi untuk mendukung operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi.

## 2.4 Sistem Informasi Manajemen Akademik

Dalam Kamus Besar Bahasa Indonesia (2013) dijelaskan bahwa akademik adalah mengenai (berhubungan dengan) akademi (lembaga pendidikan) sehingga sistem informasi manajemen akademik merupakan sistem informasi yang mengelola informasi akademik di suatu instansi pendidikan seperti informasi tentang jadwal mata pelajaran dan sebagainya.

## 2.5 Software Development Life Cycle (SDLC)

*Software Development Life Cycle* atau yang biasanya disebut dengan SDLC adalah proses mengembangkan suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (A.S & Shalahuddin, 2013). Pada umumnya, SDLC memiliki 4 tahap dasar seperti yang terlihat pada Gambar 2.1.



**Gambar 2.1** Tahapan SDLC (A.S & Shalahuddin, 2013)

Pada Gambar 2.1 dijelaskan bahwa dalam mengembangkan sebuah *software* setidaknya ada 4 tahapan yakni:

a. *Analysis*.

Ini adalah tahap dimana kebutuhan perangkat lunak dianalisis dengan cara mengumpulkan informasi secara intensif agar *software* dapat dibangun seperti apa yang diinginkan oleh pengguna.

b. *Design*.

*Design* merupakan tahap yang berfokus pada desain pembuatan *software*. Kegiatan yang dilakukan antara lain adalah merancang struktur data (*database*), arsitektur perangkat lunak, *user interface*, serta prosedur pengodean.

c. *Code*.

Tahap *code* juga bisa disebut dengan tahap *construction/implementation* yang pada tahap ini terjadi proses penerjemahan desain *software* menjadi program.

d. *Test*

Tahap ini berfokus pada pengujian *software* yang telah dibuat dari segi *logic* dan fungsional untuk memastikan bahwa semua bagian sudah berjalan sesuai dengan yang diinginkan. Hal ini dilakukan dengan tujuan untuk meminimalisir kesalahan (*error*).

Dari keempat tahap tersebut, peneliti akan berfokus pada tahap *code* atau *implementation*, karena tahap *analysis* dan *design* telah dilakukan pada penelitian sebelumnya. Peneliti juga akan membahas sebagian kecil dari tahap *Test* agar

*software* yang dihasilkan bisa diketahui tentang sudah sesuai atau tidaknya dengan apa yang diharapkan. Namun, sebelum tahap *code* dilakukan, peneliti melakukan evaluasi terhadap rancangan.

### 2.5.1 Evaluasi

Evaluasi merupakan penilaian sistematis dan objektif dari (1) sebuah sistem yang sedang atau sudah selesai dibangun, (2) perancangan, (3) implementasi dan (4) hasil jadi sebuah sistem. Tujuan dari evaluasi merupakan untuk (1) menentukan relevansi antara kebutuhan dengan persyaratan sistem dan (2) mengetahui apakah sistem sudah memenuhi tujuan pembuatan sistem itu sendiri (Norad, 2017).

#### 2.5.1.1 Verifikasi dan Validasi Perangkat Lunak

Verifikasi merupakan tahapan untuk mengetahui apakah aktivitas pengembangan sesuai dengan spesifikasi yang ditentukan (*"building the product right"*). Sedangkan validasi merupakan tahapan untuk memberikan penilaian apakah produk sesuai dengan keinginan dari pemangku kepentingan (*"building the right product"*) (Sommerville, 2009).

#### 2.5.1.2 Website Usability Evaluation (WEBUSE)

WEBUSE merupakan metode evaluasi pengembangan sistem yang berfokus pada *usability* sistem berbasis *website* dengan pendekatan tindakan subyektif yang meminta pengguna untuk mengevaluasi *website*. Pengembangan model WEBUSE sebagai standar pengukuran *usability*, dengan metode evaluasi kuesioner berbasis *web* yang memungkinkan pengguna untuk menilai *usability* dari *website* yang akan dievaluasi (Chiew & Salim, 2003). Instrumen WEBUSE tersusun dalam 4 kategori, yaitu:

1. *Content, Organization and Readability*  
Konten yang baik adalah konten yang mudah dipahami oleh pengguna, jelas, dan terorganisir dengan baik. Sedangkan, *readability* sebuah *website* diukur melalui apakah sistem berfungsi dengan benar dan memberikan informasi yang akurat.
  - a. Website memiliki konten yang menarik dan topik yang terbaru.
  - b. Saya dapat dengan mudah menemukan apa yang saya butuhkan pada *website*
  - c. Konten pada *website* terorganisir dengan baik
  - d. Mudah membaca konten pada *website*
  - e. Saya nyaman dan familiar dengan bahasa yang digunakan
2. *Navigation and Links*  
Metode yang digunakan untuk mencari dan mengakses informasi dalam situs *web* secara efektif dan efisien untuk membantu pengguna *website* disebut dengan *navigation*. Sedangkan, *links* berfungsi menghubungkan pengguna dengan cara memilih dan mengklik *links* pada halaman

*hypertext (homepage)*, yang menyebabkan terbukanya halaman baru. Berikut merupakan pertanyaan untuk mengevaluasi navigasi dan *link* :

- a. Saya dapat dengan mudah mengetahui posisi saya pada sistem.
- b. Sistem menyediakan navigasi menu dan *link* yang berguna untuk mendapatkan informasi.
- c. Sistem mudah digunakan dengan menggunakan tombol kembali atau *link*.
- d. Sistem tidak membuka banyak jendela *browser* ketika digunakan.
- e. Peletakan menu atau *link* dapat dengan mudah dikenali.

### 3. User Interface Design

Rancangan antarmuka pengguna adalah prosedur yang membutuhkan pertimbangan dengan baik saat merancang dan mengembangkan *website*. Hal yang penting dalam merancang antarmuka pengguna diantaranya menetapkan tujuan, menentukan pengguna dan menyediakan konten yang bermanfaat. Berikut merupakan pertanyaan untuk mengevaluasi rancangan antarmuka pengguna:

- a. Perancangan antarmuka pengguna sistem mudah untuk dipelajari.
- b. Sistem tidak mengandung fitur yang membutuhkan *scroll* atau animasi yang berulang.
- c. Sistem memiliki antarmuka pengguna yang konsisten.
- d. Rancangan antarmuka pengguna *website* menarik
- e. Saya nyaman dengan warna yang digunakan pada *website*

### 4. Performance and Effectiveness

*Performance website* dapat diukur dengan cara seberapa cepat suatu *website* melakukan proses atau transaksi tertentu sehingga menghasilkan kinerja pengguna yang cepat dan efisien. *Effectiveness* merupakan keberhasilan sebuah *website* menghasilkan informasi yang tepat bagi pengguna. Berikut merupakan pertanyaan untuk mengevaluasi performa dan efektifitas :

- a. Saya tidak perlu menunggu waktu yang lama untuk mengunduh file atau membuka halaman.
- b. Saya dapat dengan mudah membedakan antara *link* yang sudah diakses dan belum diakses.
- c. *Website* menyediakan informasi yang jelas dan berguna ketika saya tidak mengetahui cara untuk menggunakan *website*.
- d. Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan.
- e. *Website* dapat digunakan dengan efisien.

Pada penelitian ini, evaluasi dilakukan dengan menggunakan 3 kategori WEBUSE yaitu

- a. *Navigation and links* untuk mengevaluasi alur penggunaan sistem berdasarkan *use case scenario* yang di representasikan dalam *link* dan navigasi pada prototipe sistem informasi manajemen akademik.

- b. *User interface design* untuk mengevaluasi kemudahan dalam mempelajari antarmuka serta konsistensi desain antarmuka prototipe sistem informasi manajemen akademik.
- c. *Performance and effectiveness* untuk mengevaluasi kesesuaian respon sistem terhadap aksi yang dilakukan pada prototipe sistem informasi manajemen akademik.

### 2.5.2 Code

Tahap *code* merupakan tahap penulisan kode program sebuah perangkat lunak yang dapat dilakukan dari beberapa kemungkinan (Pressman, 2012), yakni (1) pembuatan langsung kode-kode program menggunakan bahasa pemrograman tertentu, (2) penulisan kode program secara otomatis menggunakan representasi mirip rancangan dari komponen-komponen yang akan dikembangkan, atau (3) penulisan kode program dengan menggunakan bahasa pemrograman generasi ke-4 yang langsung dapat dieksekusi.

Langkah yang harus dilakukan oleh pemrogram sebelum menulis kode program adalah (1) memahami permasalahan yang coba diselesaikan, (2) memahami prinsip dan konsep dasar perancangan, dan (3) memilih bahasa pemrograman yang sesuai dengan kebutuhan (Pressman, 2012).

Adapun saat penulisan kode dimulai, langkah yang dilakukan adalah (1) memilih struktur data sesuai dengan yang telah ditetapkan dalam rancangan yang telah dibuat, (2) memahami arsitektur program, (3) membuat antarmuka yang konsisten terhadap arsitektur program, (4) membuat logika kondisional sederhana mungkin, (5) memilih nama variabel yang bermakna dan penamaannya sesuai dengan standar penulisan bahasa pemrograman yang digunakan, dan (6) menulis kode dengan cara yang mudah dipahami (misalnya dengan menggunakan indentasi dan baris-baris kosong) (Pressman, 2012).

Ada banyak bahasa pemrograman yang bisa digunakan untuk menulis kode program diantaranya adalah JAVA, C, PHP dan lain-lain. Namun, bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa PHP. Untuk membangun *software*-pun ada konsep yang bisa digunakan diantaranya adalah konsep *Model-View-Controller* atau bisa disebut dengan MVC.

MVC merupakan konsep yang dapat digunakan untuk pemrograman berbasis objek. MVC juga memiliki beberapa *framework* seperti YII, *Laravel*, *CodeIgniter* dan lain-lain. Namun, *framework* yang digunakan dalam penelitian ini adalah *CodeIgniter*.

Setelah penulisan kode selesai, langkah yang dilakukan adalah (1) menelusuri kode yang telah ditulis untuk dilakukan pemeriksaan apakah sintaks dan logikanya sudah benar, (2) melakukan pengujian-pengujian unit, dan (3) melakukan pembetulan terhadap kesalahan yang tidak diketahui sebelumnya (Pressman, 2012).



### 2.5.2.1 PHP

PHP (*PHP Hypertext Preprocessor*) adalah sebuah bahasa pemrograman *server-side* yang mana bahasa pemrograman tersebut hanya dapat berjalan pada sebuah *web server* karena file yang dibuat menggunakan bahasa pemrograman PHP akan disimpan (diletakkan) dan diproses (dikerjakan) di server, kemudian hasil dari proses tersebut dikirim ke klien (tempat pemakai menggunakan browser).

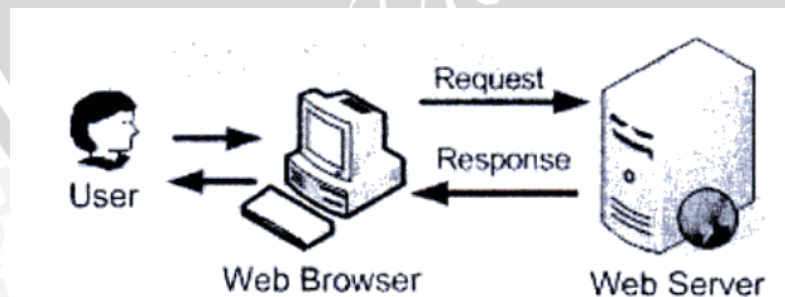
Konsep dan rancangan bahasa PHP sangat sederhana karena bisa menggunakan HTML biasa dengan menambahkan syntax yang diapit dalam tanda `<?php...?>` maka jadilah skrip PHP.

Skrip PHP disembunyikan dalam *server* dan *output* yang ditampilkan di *browser* adalah format teks seperti HTML, XHTML atau XML. PHP juga mendukung basis data populer yang sering digunakan dalam *web server*, antara lain: PostgreSQL, IBM DB2, MySQL dan lain-lain (Gilmore, 2010).

### 2.5.2.2 Prinsip Kerja PHP

Sistem Kerja dari PHP adalah diawali dengan permintaan yang berasal dari halaman *website* pada *browser*. *Browser* akan melakukan pencarian berdasarkan alamat *website* dari *web server*, lalu mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencarikan *file* yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan berkas yang diminta segera menerjemahkan kode HTML dan menampilkannya. Ketika *user* melakukan *request* pada halaman yang mengandung skrip PHP maka pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web server*, *web server* akan memeriksa file isi skrip dari halaman PHP tersebut. Apabila dalam *file* tersebut tidak mengandung skrip PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan skrip-skrip PHP dan mengolah skrip tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke *browser user* (Gilmore, 2010).

Adapun ilusi komunikasi *user*, *browser* dan *server* bisa dilihat pada Gambar 2.2:



Gambar 2.2 Komunikasi *user*, *browser* dan *web server* (Sunyoto, 2007)

### 2.5.2.3 Pemrograman Berorientasi Objek

Pendekatan pemrograman berorientasi objek adalah suatu cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya) (A.S & Shalahuddin, 2013).

Beberapa konsep dasar yang ada pada metodologi berorientasi objek:

a. Kelas (*Class*)

Kelas adalah kumpulan dari objek-objek yang memiliki karakteristik sama. Sebuah kelas akan memiliki sifat (atribut), kelakuan (operasi/metode), dan hubungan (relationship). Secara fisik, kelas merupakan berkas atau *file* yang berisi kode program, di mana kode program adalah semua hal yang berhubungan dengan nama kelas.

b. Objek (*Object*)

Objek adalah sebuah abstraksi dan sesuatu yang mewakili dunia nyata seperti manusia, benda, satuan organisasi, tempat, kejadian, atau hal-hal yang bersifat abstrak. Objek juga merupakan entitas yang mampu menyimpan informasi dan mempunyai operasi yang bisa diterapkan atau juga dapat berpengaruh terhadap status objeknya.

c. Metode (*Method*)

Metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Dalam sebuah kelas, jumlah metode bisa lebih dari satu.

d. Atribut (*Attribute*)

Atribut adalah variabel global yang dimiliki oleh sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas. Atribut dipunyai secara individual oleh sebuah objek, misalnya nama, berat, jenis dan sebagainya.

### 2.5.2.4 MVC

Konsep MVC merupakan konsep yang digunakan didalam *framework* untuk membangun aplikasi, pada konsep MVC dilakukan pemisahan antara *logic* dengan tampilan dan basis data. Manfaat konsep MVC adalah membuat *coding logic* lebih sederhana (Sidik, 2012).

a. *Model*

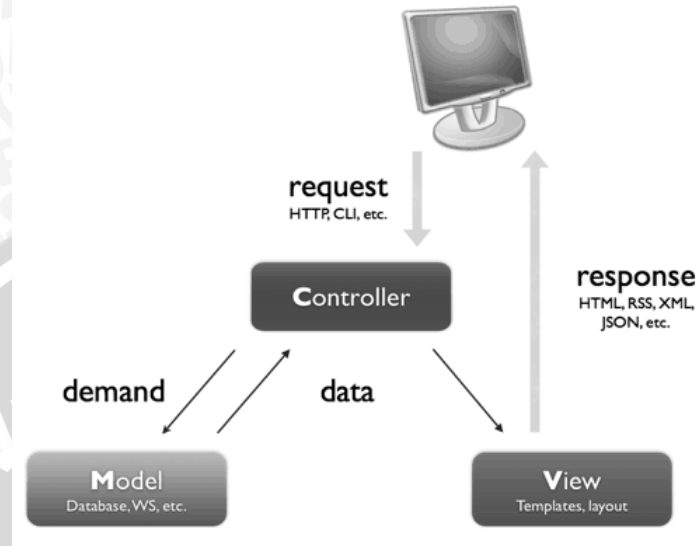
Merupakan kelas yang berisi kode untuk melakukan pengolahan data dari basis data.

b. *View*

Merupakan kelas yang berisi kode untuk tampilan suatu program. Tampilan dapat berupa *web page*, *header*, *footer*, *content* dan apa saja yang berjenis tampilan.

c. *Controller*

Merupakan kelas yang berisi kode *logic*, algoritma dan sebagai penghubung antara *model*, *view* dan sumber lain yang diperlukan untuk mengelola *HTTP request* dan *generate web page*.



**Gambar 2.3** Ilustrasi *design pattern* MVC (Basuki, 2014)

Gambar 2.3 adalah simulasi dari cara kerja MVC dimana prosesnya dimulai dengan komputer yang melakukan *request* melalui *browser* di komputer yang mana *request* tersebut akan diterima dan diproses oleh *controller* sehingga *controller* dapat memanggil (jika diperlukan) *model* untuk mengolah data, kemudian *controller* memanggil *view* yang sesuai untuk menampilkan data tersebut ke *browser* sebagai *response* dari *request* sebelumnya.

Kelebihan MVC (Budiharto, 2014):

a. Pemisahan persoalan desain

Adanya pemisahan data (*model*), tampilan (*view*) dan kontrol (*controller*) menjadikan aplikasi menjadi lebih fleksibel karena jika terjadi modifikasi pada salah satu komponen hanya berdampak minimum terhadap komponen lainnya. Contohnya adalah jika ada tampilan baru bukan berarti harus ada model baru.

b. Meningkatkan pembagian kerja

*Developer* dengan keahlian yang berbeda dapat fokus pada keterampilan inti mereka dan berkolaborasi melalui antarmuka yang jelas.

c. Lebih mudah ditangani dan diperluas

Struktur penulisan yang baik dapat mengurangi kompleksitas kode. Dengan demikian, duplikasi kode menjadi minimal.

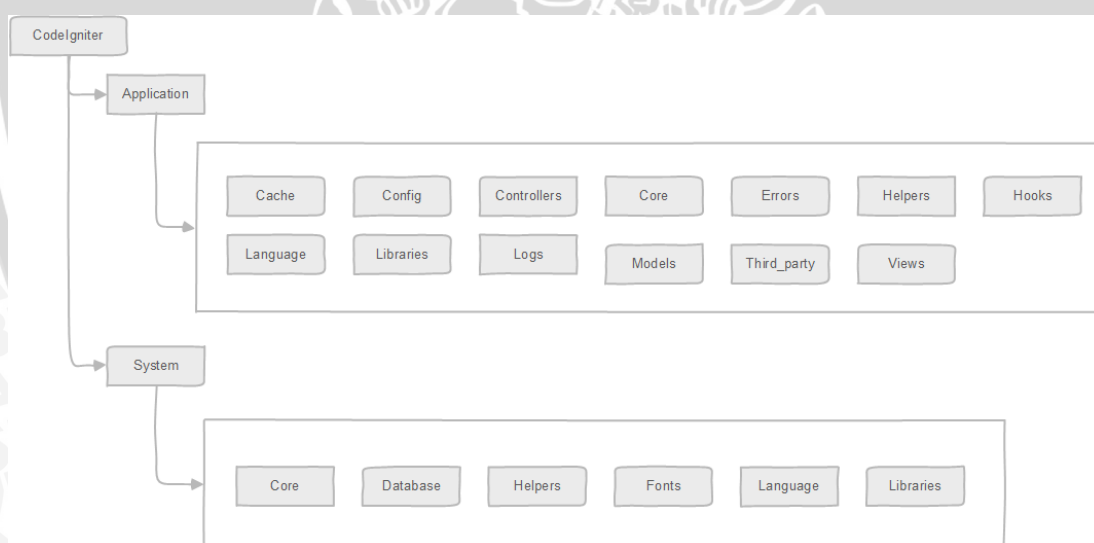
### 2.5.2.5 Framework

*Framework* adalah seumpulan perintah atau fungsi dasar yang dapat membantu dalam menyelesaikan proses-proses yang lebih kompleks, menangani berbagai masalah dalam pemrograman seperti koneksi basis data, pemanggilan variabel, dll. *Framework* memungkinkan *developer* lebih fokus dan lebih cepat dalam membangun aplikasi (Sidik, 2012).

### 2.5.2.6 CodeIgniter

*CodeIgniter* adalah salah satu *framework* PHP yang tangguh dan populer. *CodeIgniter* menerapkan *pattern MVC* yang fleksibel karena model dapat tidak digunakan (pengembang dapat hanya menggunakan *view* dan *controller* saja). Pemrogram tidak perlu membuat program dari awal (*from scratch*) karena CI menyediakan sekumpulan librari yang banyak dimana untuk mengakses librari tersebut dapat menggunakan antarmuka dan struktur logika yang sederhana. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan (Sidik, 2012).

*CodeIgniter* terdiri dari kumpulan *folder* yang berisi *file* PHP, *java script*, *css*, *txt* dan *file* berbasis web lainnya dengan *setting* tertentu untuk menggunakannya, dan menyediakan *library* serta *helper* yang dapat dimanfaatkan didalam pemrograman PHP (Candrawati, 2014). Pada Gambar 2.4 adalah struktur *file codeigniter*.



**Gambar 2.4** Struktur File CI (Candrawati, 2014)

Gambar 2.4 menunjukkan bahwa pada *CodeIgniter* terdapat 2 folder utama yaitu (1) folder *Application* yang berisi folder *Cache*, *Config*, *Controllers*, *Core*, *Errors*, *Helpers*, *Hooks*, *Language*, *Libraries*, *Logs*, *Models*, *Third\_party*, *Views*, dan (2) folder *Systems* yang berisi folder *Core*, *Database*, *Helpers*, *Fonts*, *Language*, *Libraries*.

Penulisan kode program dalam *codeigniter* tidak jauh berbeda dengan penulisan kode program dengan bahasa PHP lainnya. Namun, ada beberapa ciri khas penulisan kode program dalam *codeigniter* (EllisLab, 2016), yakni:

- a. Setiap *class controller* adalah turunan dari *class CI\_Controller* seperti pada Gambar 2.5 dibawah ini:

```

1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Welcome extends CI_Controller {
5
6      /**
7       * Index Page for this controller.
8       *
9       * Maps to the following URL
10      *   http://example.com/index.php/welcome
11      * - or -
12      *   http://example.com/index.php/welcome/index
13      * - or -
14      * Since this controller is set as the default controller in
15      * config/routes.php, it's displayed at http://example.com/
16      *
17      * So any other public methods not prefixed with an underscore will
18      * map to /index.php/welcome/<method_name>
19      * @see http://codeigniter.com/user_guide/general/urls.html
20      */
21     public function index()
22     {
23         $this->load->view('welcome_message');
24     }
25 }
26

```

**Gambar 2.5** Penulisan kode program *class controller*

Pada Gambar 2.5 merupakan contoh dari *class Welcome* yang merupakan turunan dari *class CI\_Controller* yang berarti *class Welcome* dapat mengakses seluruh *method* dan variabel yang didefinisikan dalam *class CI\_Controller*.

- b. Setiap *class model* adalah turunan dari *class CI\_Model* seperti pada Gambar 2.6 dibawah ini:

```

1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class M_Welcome extends CI_Model {
5
6
7      public function get()
8      {
9          return $this->db->get('table');
10     }
11 }
12

```

**Gambar 2.6** Penulisan kode program *class model*

Pada Gambar 2.6 merupakan contoh dari *class M\_Welcome* yang merupakan turunan dari *class CI\_Model* yang berarti *class M\_Welcome* dapat mengakses seluruh *method* dan variabel yang didefinisikan dalam *class CI\_Model*.

*Framework CodeIgniter* dipilih karena keunggulan yang dimiliki oleh *CodeIgniter*. Adapun kelebihan *CodeIgniter* adalah sebagai berikut (Basuki, 2014):

1. *CodeIgniter* termasuk *framework* sederhana dengan ukuran kecil, namun dengan kemampuan besar.
2. *CodeIgniter* dieksekusi dengan cepat dan memiliki *performance* yang handal.
3. Menerapkan konsep *MVC*.
4. Aplikasi yang dibuat dengan *CodeIgniter* menghasilkan "*clean URL*" yang *Search Engine Friendly (SEF)*.
5. Aplikasi yang dibuat dengan *CodeIgniter* kompatibel dengan standar *hosting* yang menjalankan beberapa versi PHP dan konfigurasi yang berbeda.
6. Konfigurasi minimal.
7. Menyediakan *library* dan *helper* yang cukup lengkap.
8. Fleksibel dan tidak diharuskan untuk mematuhi aturan yang terlalu ketat dalam penggunaannya.
9. Dokumentasi lengkap dan jelas.
10. Komunitas yang besar, sehingga dapat digunakan sebagai media untuk belajar.

#### **2.5.2.7 Apache Web Server**

*Apache Web Server* adalah perangkat lunak yang memberikan layanan *webite*. *Web server* menggunakan protokol yang disebut dengan HTTP (*HyperText Transfer Protocol*). Banyak pilihan *web server* di dunia *open source* yang dapat digunakan, penggunaannya bergantung pada keperluan. Apache adalah salah satu *web server* yang sangat terkenal dan menjadi standar *de facto* setiap distribusi Linux. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain (Sidik, 2012).

#### **2.5.3 Test**

Menjaga kualitas perangkat lunak sangatlah penting karena kualitas perangkat dapat mempengaruhi kepuasan pelanggan. Salah satu cara untuk menjaga kualitas perangkat lunak adalah dengan melakukan pengujian terhadap perangkat lunak sebelum perangkat lunak diberikan kepada pelanggan atau selama perangkat lunak masih dalam tahap pengembangan.

A.S dan Shalahuddin (2013) berpendapat bahwa pengujian perangkat lunak merupakan sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) yang mana verifikasi lebih berfokus kepada apakah perangkat lunak sudah dibangun dengan benar sedangkan validasi berfokus kepada apakah sudah membangun produk yang benar. Menurut A.S dan Shalahuddin (2013) pengujian berdasarkan validasi memiliki 2 pendekatan, yaitu

*black-box testing* dan *white-box testing*. Namun, metode pengujian yang digunakan dalam penelitian ini adalah *black-box testing*. Pengujian ini dilakukan terhadap fungsionalitas sistem sehingga dapat diketahui apakah keluaran sistem sudah sesuai dengan ketentuan yang telah dijabarkan dalam kebutuhan fungsional pada penelitian sebelumnya.

Metode pengujian non-fungsional dilakukan menggunakan *compatibility testing* terhadap *browser*. Pengujian ini dilakukan berdasarkan definisi dari kebutuhan non-fungsional yang telah dijabarkan dalam penelitian sebelumnya.

### 2.5.3.1 Black-Box Testing

*Black-Box Testing* merupakan pengujian perangkat lunak dari segi fungsional *software*. Pengujian ini bertujuan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari *software* sesuai dengan spesifikasi yang dibutuhkan (A.S & Shalahuddin, 2013).

Salah satu metode yang dapat digunakan untuk melakukan *black-box testing* adalah *functional testing*. *Functional testing* merupakan pengujian perangkat lunak dengan melakukan observasi pada keluaran (*output*) oleh masukan (*input*) tertentu tanpa menganalisa kode program. (Agarwal, et al., 2010).

Pengujian *black-box* dipilih karena dalam pelaksanaannya mampu menemukan kesalahan dalam kategori, (1) fungsi yang salah atau hilang, (2) kesalahan antarmuka, (3) kesalahan dalam struktur data atau akses basis data eksternal, (4) kesalahan perilaku atau kinerja, dan (5) kesalahan inisialisasi dan penghentian (Pressman, 2012).

### 2.5.3.2 Compatibility Testing

Pengujian *Compatibility* merupakan salah satu tipe pengujian non-fungsional yang bertujuan untuk memastikan bahwa perangkat lunak yang dibangun mampu berjalan dengan baik pada lingkungan yang berbeda (lingkungan pengguna). Pengujian *compatibility* dilakukan untuk mengetahui apakah sistem dapat berjalan *Hardware, Operating Systems, Software, Network, Browser, Devices, Mobile, Versions* yang berbeda (Guru99, 2016).

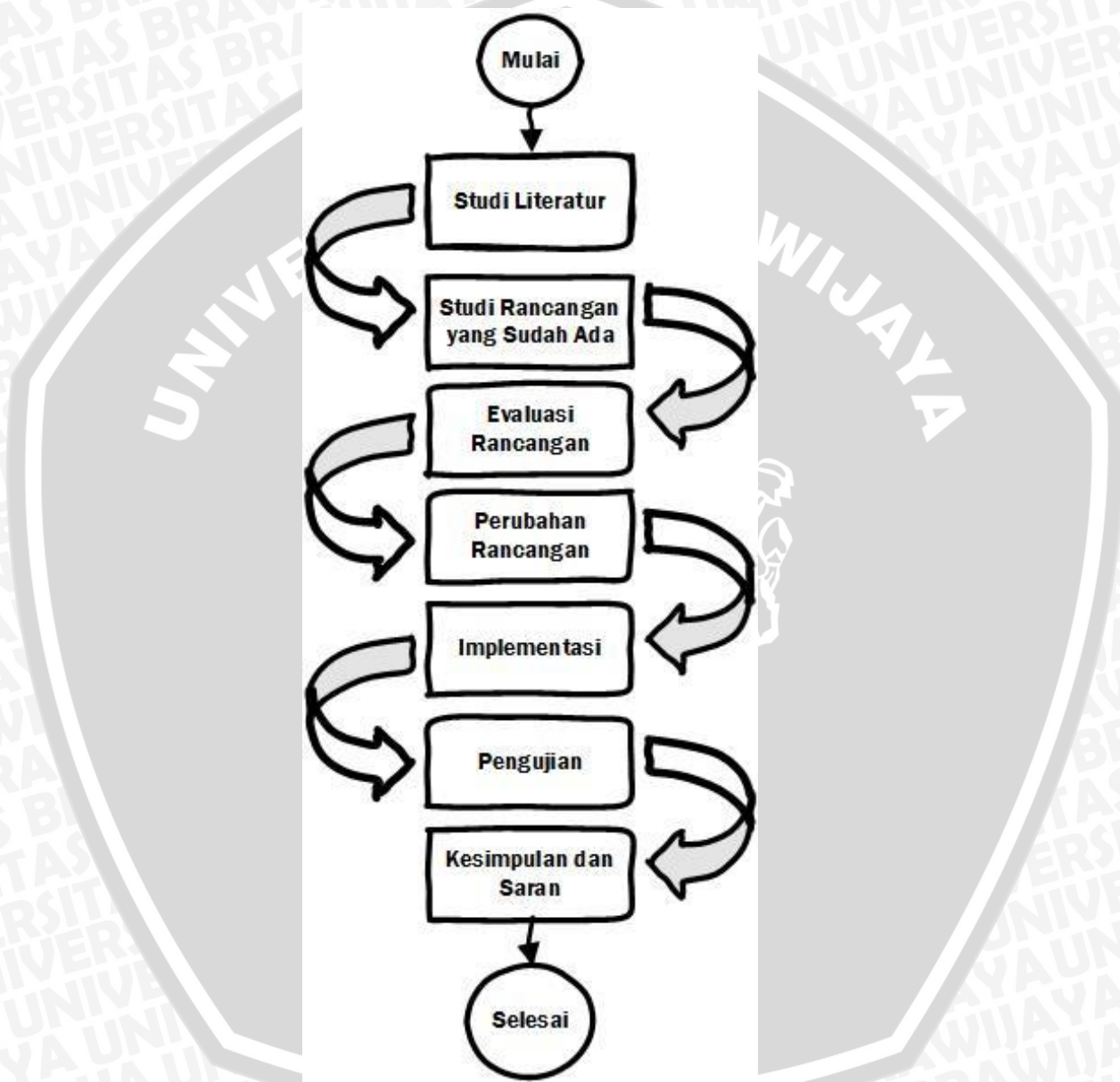
Tipe dari pengujian *compatibility* yang akan digunakan pada penelitian ini adalah tipe pengujian pada sisi *browser*. Pengujian ini dilakukan pada beberapa *browser* berbeda yang berjalan pada perangkat komputer maupun perangkat *mobile*, seperti *Mozilla Firefox, Google Chrome, Internet Explorer, Microsoft Edge, Safari, browser Android, dan browser iOS*. Pengujian *compatibility* pada sisi *browser* dapat dilakukan dengan menggunakan *tools* atau alat bantu berupa perangkat lunak, seperti *SortSite*.

*SortSite* dipilih karena mampu menguji perangkat lunak berbasis *web* secara keseluruhan pada setiap halaman yang dimiliki. Selain itu, *SortSite* mampu menguji perangkat lunak berbasis *web* pada beberapa *checkpoints* diantaranya *Accessibility, Broken Links, Compatibility, Search Engine Optimization, Privacy, Web Standart, dan Usability* (PowerMapper, 2016).

## BAB 3 METODOLOGI

### 3.1 Alur Penelitian

Penelitian ini dilaksanakan dengan alur penelitian seperti pada Gambar 3.1



Gambar 3.1 Alur Penelitian



### 3.1.1 Studi Literatur

Pada tahap ini, peneliti mempelajari literatur-literatur yang dapat mendukung atau berkaitan dengan penelitian ini karena literatur akan sangat bermanfaat bagi peneliti untuk dijadikan referensi dalam melakukan penelitian.

### 3.1.2 Studi Rancangan yang Sudah Ada

Penelitian ini adalah penelitian yang berhubungan dengan penelitian sebelumnya yakni penelitian tentang Analisis dan Perancangan Sistem Informasi Manajemen Akademik oleh Izzul Fatha. Dalam penelitian yang dilakukan oleh Izzul Fatha telah membahas tentang analisis kebutuhan, proses bisnis dan perancangan yang mana hasil dari perancangan akan dikembangkan untuk digunakan dalam implementasi sistem.

### 3.1.3 Evaluasi Rancangan

Evaluasi dilakukan terhadap rancangan yang telah dibuat dalam penelitian yang berjudul Analisis dan Perancangan Sistem Informasi Manajemen Akademik oleh Izzul Fatha pada tahun 2017.

Evaluasi dilakukan dengan menyederhanakan tinjauan *informal (walkthrough)* dan *prototyping*. Proses evaluasi dengan menggunakan tinjauan meliputi presentasi persyaratan kepada pengguna, (1) memberikan uji coba berupa *test case* kepada pengguna menggunakan prototipe untuk mengetahui kesesuaian antara alur pengguna dan spesifikasi, (2) serta melakukan review untuk mengetahui kesesuaian informasi, alur harapan pengguna, navigasi dan antarmuka sistem. Pertanyaan terkait dengan navigasi dan antarmuka pengguna sistem menggunakan WEBUSE. Tinjauan (*review*) dilakukan pada 3 *use case* yang terdapat pada prototipe Sistem Informasi Manajemen Akademik pada Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas, Malang.

### 3.1.4 Perubahan Rancangan

Rancangan merupakan sebuah panduan untuk membangun sebuah perangkat lunak. Perubahan rancangan merupakan perbaikan dari rancangan yang dihasilkan oleh penelitian sebelumnya ke dalam rancangan yang disesuaikan dengan *framework* yang digunakan pada penelitian ini agar dapat diimplementasikan ke sebuah bahasa pemrograman.

### 3.1.5 Implementasi

Sistem diimplementasikan berdasarkan perancangan yang telah diperbaiki dengan menggunakan bahasa pemrograman PHP yang didukung oleh bahasa pemrograman lain seperti CSS dan *javascript*. Perancangan tersebut diimplementasikan dengan pendekatan pemrograman berorientasi objek karena dalam penelitian perancangan sebelumnya juga menggunakan pendekatan berorientasi objek. Dalam pengimplementasiannya juga menerapkan konsep MVC dengan menggunakan *framework CodeIgniter*.

### 3.1.6 Pengujian

Pada tahap pengujian sistem, parameter yang dilakukan disesuaikan dengan perancangan sistem. Pengujian menggunakan metode *blackbox* dan pengujian kompatibilitas *browser*. Pengujian secara *black-box* menguji apakah masukan yang diberikan dan keluaran yang diberikan sesuai. Metode ini juga digunakan untuk mengetahui apakah sistem dibangun sesuai spesifikasi kebutuhan. Sedangkan pengujian kompatibilitas *browser* menguji apakah sistem dapat berjalan di berbagai macam *browser*.

### 3.1.7 Kesimpulan dan Saran

Kesimpulan diberikan setelah semua tahapan selesai dilakukan. Kesimpulan penelitian akan menjawab pertanyaan permasalahan penelitian. Hasil kesimpulan diberikan berdasarkan keseluruhan proses pembangunan sistem informasi manajemen akademik di Madrasah Tsanawiyah Muhammadiyah 1 Tlogomas Malang untuk menunjukkan apakah masalah penelitian terjawab dan tujuan dari penelitian terpenuhi. Selain kesimpulan, saran juga dibutuhkan untuk kepentingan pengembangan sistem informasi.



## BAB 4 EVALUASI RANCANGAN

### 4.1 Evaluasi Rancangan

Evaluasi rancangan dilakukan untuk mengetahui apakah rancangan yang telah dilakukan pada penelitian Fatha (2017) sudah sesuai dengan kebutuhan *stakeholder*. Evaluasi dilakukan pada 3 *use case*, yaitu:

1. Manajemen jadwal mata pelajaran.
2. Manajemen presensi siswa.
3. Rekap presensi.

#### 4.1.1 Manajemen Jadwal Mata Pelajaran

Table 4.1 merupakan tinjauan dengan menggunakan *prototype* terkait dengan kesesuaian alur penggunaan dengan spesifikasi *use case* manajemen jadwal mata pelajaran.

**Tabel 4.1** Kesesuaian Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran

| <b>Test Case</b>  | Tata usaha menambahkan jadwal mata pelajaran baru  |  |
|---|--|--|
| <b>Langkah-langkah Pengguna</b>   | <b>Langkah-langkah <i>Basic Flow</i></b>   | <b>Kesesuaian</b>                                |
| 1. Tata usaha memilih fungsi menambah jadwal mata pelajaran<br>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan. | 1. Tata usaha memilih fungsi jadwal mata pelajaran<br>2. Sistem akan menampilkan semua data jadwal yang telah di- <i>input</i> -kan<br>3. Tata usaha memilih tambah jadwal mata pelajaran<br>4. Sistem akan menampilkan form tambah data mata pelajaran<br>5. Tata usaha mengisi form tambah sesuai yang diinginkan dan menyimpan data ke sistem | Alur pengguna dan <i>basic flow</i> sudah sesuai |

**Tabel 4.1** Kesesuaian Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran (lanjutan)

|   |  |  |
|---|--|--|
|   | 6. Sistem akan menyimpan data dan menampilkan pesan data berhasil ditambahkan.   |  |
| <b>Test Case</b>  | Tata usaha menambahkan jadwal mata pelajaran baru, jadwal bentrok  |  |
| <b>Langkah-langkah Pengguna</b>   | <b>Langkah-langkah <i>Basic Flow</i> dan <i>Alternative Flow</i></b>   | <b>Kesesuaian</b>  |
| <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi menambah jadwal mata pelajaran</li> <li>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.</li> </ol> | <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi jadwal mata pelajaran</li> <li>2. Sistem akan menampilkan semua data jadwal yang telah di-<i>input</i>-kan</li> <li>3. Tata usaha memilih tambah jadwal mata pelajaran</li> <li>4. Sistem akan menampilkan form tambah data mata pelajaran</li> <li>5. Tata usaha mengisi form tambah sesuai yang diinginkan dan menyimpan data ke sistem</li> <li>6. Sistem akan menyimpan data dan menampilkan pesan data tidak berhasil ditambahkan karena ada bentrokkan jadwal guru.</li> </ol> | Alur pengguna dan <i>basic flow</i> dan <i>Alternative flow</i> sudah sesuai |

**Tabel 4.1** Kesesuaian Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran (lanjutan)

| <b>Test Case</b>  | Tata usaha menambahkan jadwal mata pelajaran baru, alokasi jam pelajaran penuh   |  |
|---|--|--|
| <b>Langkah-langkah Pengguna</b>   | <b>Langkah-langkah <i>Basic Flow</i> dan <i>Alternative Flow</i></b>   | <b>Kesesuaian</b>  |
| 1. Tata usaha memilih fungsi menambah jadwal mata pelajaran<br>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan. | 1. Tata usaha memilih fungsi jadwal mata pelajaran<br>2. Sistem akan menampilkan semua data jadwal yang telah di-input-kan<br>3. Tata usaha memilih tambah jadwal mata pelajaran<br>4. Sistem akan menampilkan form tambah data mata pelajaran<br>5. Tata usaha mengisi form tambah sesuai yang diinginkan dan menyimpan data ke sistem<br>6. Sistem akan menyimpan data dan menampilkan pesan data tidak berhasil ditambahkan karena alokasi jam pelajaran sudah penuh. | Alur pengguna dan <i>basic flow</i> dan <i>alternative flow</i> sudah sesuai |
| <b>Test Case</b>  | Tata usaha mengubah jadwal mata pelajaran  |  |
| <b>Langkah-langkah Pengguna</b>   | <b>Langkah-langkah <i>Basic Flow</i> dan <i>Alternative Flow</i></b>   | <b>Kesesuaian</b>  |
| 1. Tata usaha memilih fungsi edit jadwal mata pelajaran<br>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.     | 1. Tata usaha memilih fungsi jadwal mata pelajaran<br>2. Sistem akan menampilkan semua data jadwal yang telah di-input-kan   | Alur pengguna dan <i>basic flow</i> dan <i>alternative flow</i> sudah sesuai |

**Tabel 4.1** Kesesuaian Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran (lanjutan)

|   |   |  |
|---|---|--|
|   | <ol style="list-style-type: none"> <li>3. Tata usaha memilih edit jadwal mata pelajaran</li> <li>4. Sistem akan menampilkan form edit data mata pelajaran</li> <li>5. Tata usaha mengisi form sesuai yang diinginkan dan menyimpan data ke sistem</li> <li>6. Sistem akan menyimpan data dan menampilkan pesan data berhasil diubah.</li> </ol>   |  |
| <b>Test Case</b>  | Tata usaha mengubah jadwal mata pelajaran, jadwal bentrok   |  |
| <b>Langkah-langkah Pengguna</b>   | <b>Langkah-langkah <i>Basic Flow</i> dan <i>Alternative Flow</i></b>  | <b>Kesesuaian</b>  |
| <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi edit jadwal mata pelajaran</li> <li>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.</li> </ol> | <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi jadwal mata pelajaran</li> <li>2. Sistem akan menampilkan semua data jadwal yang telah di-<i>input</i>-kan</li> <li>3. Tata usaha memilih edit jadwal mata pelajaran</li> <li>4. Sistem akan menampilkan form edit data mata pelajaran</li> <li>5. Tata usaha mengisi form sesuai yang diinginkan dan menyimpan data ke sistem</li> <li>6. Sistem akan menyimpan data dan menampilkan pesan data tidak berhasil diubah karena ada bentrokan jadwal guru.</li> </ol> | Alur pengguna dan <i>basic flow</i> dan <i>alternative flow</i> sudah sesuai |

**Tabel 4.1** Kesesuaian Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran (lanjutan)

| <b>Test Case</b>   | Tata usaha mengubah jadwal mata pelajaran, alokasi jam pelajaran penuh  |  |
|--|---|--|
| <b>Langkah-langkah Pengguna</b>  | <b>Langkah-langkah <i>Basic Flow</i> dan <i>Alternative Flow</i></b>  | <b>Kesesuaian</b>  |
| <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi edit jadwal mata pelajaran</li> <li>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.</li> </ol>  | <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi jadwal mata pelajaran</li> <li>2. Sistem akan menampilkan semua data jadwal yang telah di-<i>input</i>-kan</li> <li>3. Tata usaha memilih edit jadwal mata pelajaran</li> <li>4. Sistem akan menampilkan form edit data mata pelajaran</li> <li>5. Tata usaha mengisi form sesuai yang diinginkan dan menyimpan data ke sistem</li> <li>6. Sistem akan menyimpan data dan menampilkan pesan data tidak berhasil diubah karena alokasi jam pelajaran sudah penuh.</li> </ol> | Alur pengguna dan <i>basic flow</i> dan <i>alternative flow</i> sudah sesuai |
| <b>Test Case</b>   | Tata usaha menghapus jadwal mata pelajaran  |  |
| <b>Langkah-langkah Pengguna</b>  | <b>Langkah-langkah <i>Basic Flow</i> dan <i>Alternative Flow</i></b>  | <b>Kesesuaian</b>  |
| <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi hapus jadwal mata pelajaran</li> <li>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.</li> </ol> | <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi jadwal mata pelajaran</li> <li>2. Sistem akan menampilkan semua data jadwal yang telah di-<i>input</i>-kan</li> <li>3. Tata usaha memilih hapus jadwal mata pelajaran</li> </ol>  | Alur pengguna dan <i>basic flow</i> dan <i>alternative flow</i> sudah sesuai |

**Tabel 4.1** Kesesuaian Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran (lanjutan)

|  |  |  |
|--|--|--|
|  | <p>4. Sistem akan menampilkan form hapus data mata pelajaran</p> <p>5. Sistem akan menghapus data dan menampilkan pesan data berhasil dihapus.</p> |  |
|--|--|--|

Tabel 4.2 merupakan tinjauan terkait dengan harapan alur pengguna pada *use case* manajemen jadwal mata pelajaran.

**Tabel 4.2** Harapan Alur Pengguna *Use Case* Manajemen Jadwal Mata Pelajaran

| Pertanyaan  | Setuju   | Netral | Tidak Setuju | Keterangan  |
|---|--|--------|--------------|---|
| Respon Sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan. | V  |        |              | Karena hasil yang ditampilkan oleh sistem sudah sesuai dengan keinginan pengguna. |
| <b>Pernyataan/Masukan dan Saran</b>                                     | Alur sudah sesuai dengan harapan pengguna dalam memanajemen jadwal mata pelajaran. |        |              |   |

Tabel 4.3 merupakan tinjauan terkait dengan navigasi *use case* manajemen jadwal mata pelajaran.

**Tabel 4.3** Tinjauan Navigasi *Use Case* Manajemen Jadwal Mata Pelajaran

| Pertanyaan   | Setuju | Netral | Tidak Setuju | Keterangan                              |
|--|--------|--------|--------------|---|
| Saya dapat dengan mudah mengetahui posisi saya pada sistem.                                | V      |        |              | Karena posisi navigasi sudah jelas.     |
| Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi. | V      |        |              | Karena isi dari navigasi sudah jelas.   |
| Sistem mudah digunakan dengan menggunakan tombol kembali atau <i>link</i> .                | V      |        |              | Tombol dan <i>link</i> dapat digunakan. |



**Tabel 4.3** Tinjauan Navigasi *Use Case* Manajemen Jadwal Mata Pelajaran (lanjutan)

|   |  |  |  |
|---|--|--|--|
| Sistem tidak membuka banyak jendela <i>browser</i> ketika digunakan | V  |  | Sistem tidak membuka banyak jendela.   |
| Peletakan menu atau <i>link</i> dapat dengan mudah dikenali.        | V  |  | Karenan dengan ukuran huruf dan tombol yang jelas, ketika membuka sistem langsung mengetahui menu dari sistem. |
| <b>Masukan dan Saran</b>  | Navigasi sudah sesuai dengan harapan pengguna dalam memanajemen jadwal mata pelajaran. |  |  |

#### 4.1.2 Manajemen Presensi Siswa

Table 4.4 merupakan tinjauan dengan menggunakan *prototype* terkait dengan kesesuaian alur penggunaan dengan spesifikasi *use case* manajemen presensi siswa.

**Tabel 4.4** Kesesuaian Alur Pengguna *Use Case* Manajemen Presensi Siswa

| <b>Test Case</b>   | Guru menambahkan presensi   |  |
|--|---|--|
| <b>Langkah-langkah Pengguna</b>  | <b>Langkah-langkah <i>Basic Flow</i></b>  | <b>Kesesuaian</b>                                |
| 1. Guru memilih fungsi menambah presensi siswa<br>2. Guru mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan. | 1. Guru memilih fungsi presensi siswa<br>2. Guru pengajar memilih operasi tambah presensi siswa<br>3. Sistem akan memproses permintaan, dan menampilkan form presensi siswa untuk ditambahkan | Alur pengguna dan <i>basic flow</i> sudah sesuai |

**Tabel 4.4** Kesesuaian Alur Pengguna *Use Case* Manajemen Presensi Siswa (lanjutan)

|  |  |  |
|--|--|--|
|  | <ol style="list-style-type: none"> <li>2. Guru pengajar mengisi presensi siswa pada form presensi</li> <li>3. Guru pengajar memilih tombol simpan jika sudah selesai melengkapi form presensi siswa.</li> <li>4. Sistem akan menyimpan data dan menampilkan pesan data berhasil ditambahkan.</li> </ol>  |  |
| <b>Test Case</b>   | Tata usaha mengubah presesensi   |  |
| <b>Langkah-langkah Pengguna</b>  | <b>Langkah-langkah <i>Basic Flow</i></b>   | <b>Kesesuaian</b>                                |
| <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi edit presensi siswa</li> <li>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.</li> </ol> | <ol style="list-style-type: none"> <li>1. Tata usaha memilih fungsi presensi siswa</li> <li>2. Tata usaha pengajar memilih operasi edit presensi siswa</li> <li>3. Sistem akan memproses permintaan, dan menampilkan form edit presensi siswa</li> <li>4. Tata usaha mengubah data presensi siswa</li> <li>5. Tata usaha menyimpan perubahan data presensi siswa.</li> <li>6. Sistem akan menyimpan data yan telah diubah dan menampilkan pesan data berhasil diubah.</li> </ol> | Alur pengguna dan <i>basic flow</i> sudah sesuai |

**Tabel 4.4** Kesesuaian Alur Pengguna *Use Case* Manajemen Presensi Siswa (lanjutan)

| <i>Test Case</i>  | Tata usaha menghapus presensi   |  |
|---|---|--|
| Langkah-langkah Pengguna  | Langkah-langkah <i>Basic Flow</i>   | Kesesuaian                                       |
| 1. Tata usaha memilih fungsi hapus presensi siswa<br>2. Tata usaha mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan. | 1. Petugas tata usaha memilih menu presensi siswa<br>2. Sistem akan memproses permintaan dan menampilkan presensi siswa.<br>3. Petugas tata usaha memilih operasi hapus presensi siswa pada data presensi yang diinginkan<br>4. Sistem akan menampilkan form hapus untuk diverifikasi<br>5. Petugas tata usaha memverifikasi hapus presensi siswa<br>6. Sistem akan menghapus presensi siswa dan menampilkan pesan data berhasil dihapus. | Alur pengguna dan <i>basic flow</i> sudah sesuai |

Tabel 4.5 merupakan tinjauan terkait dengan harapan alur pengguna pada *use case* manajemen presensi.

**Tabel 4.5** Harapan Alur Pengguna *Use Case* Manajemen Presensi Siswa

| Pertanyaan  | Setuju | Netral | Tidak Setuju | Keterangan  |
|---|--------|--------|--------------|---|
| Respon Sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan. | V      |        |              | Karena hasil yang ditampilkan oleh sistem sudah sesuai dengan keinginan pengguna. |

**Tabel 4.5** Harapan Alur Pengguna *Use Case* Manajemen Presensi Siswa (lanjutan)

|                                     |   |
|-------------------------------------|---|
| <b>Pernyataan/Masukan dan Saran</b> | Alur sudah sesuai dengan harapan pengguna dalam manajemen presensi siswa. |
|-------------------------------------|---|

Tabel 4.6 merupakan tinjauan terkait dengan navigasi *use case* manajemen presensi siswa.

**Tabel 4.6** Tinjauan Navigasi *Use Case* Manajemen Presensi Siswa

| Pertanyaan   | Setuju  | Netral | Tidak Setuju | Keterangan   |
|--|---|--------|--------------|--|
| Saya dapat dengan mudah mengetahui posisi saya pada sistem.                                | V   |        |              | Karena posisi navigasi sudah jelas.  |
| Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi. | V   |        |              | Karena isi dari navigasi sudah jelas.  |
| Sistem mudah digunakan dengan menggunakan tombol kembali atau <i>link</i> .                | V   |        |              | Tombol dan <i>link</i> dapat digunakan.  |
| Sistem tidak membuka banyak jendela <i>browser</i> ketika digunakan                        | V   |        |              | Sistem tidak membuka banyak jendela.   |
| Peletakan menu atau <i>link</i> dapat dengan mudah dikenali.                               | V   |        |              | Karenan dengan ukuran huruf dan tombol yang jelas, ketika membuka sistem langsung mengetahui menu dari sistem. |
| <b>Masukan dan Saran</b>   | Navigasi sudah sesuai dengan harapan pengguna dalam manajemen presensi siswa. |        |              |  |

#### 4.1.3 Rekap Presensi Siswa

Table 4.7 merupakan tinjauan dengan menggunakan *prototype* terkait dengan kesesuaian alur penggunaan dengan spesifikasi *use case* rekap presensi siswa.

**Tabel 4.7** Kesesuaian Alur Pengguna *Use Case* Rekap Presensi Siswa

| <i>Test Case</i>   | Guru menambahkan presensi   |  |
|--|---|--|
| Langkah-langkah Pengguna   | Langkah-langkah <i>Basic Flow</i>   | Kesesuaian                                       |
| <ol style="list-style-type: none"> <li>1. Tata usaha atau kepala sekolah memilih fungsi menambah presensi siswa</li> <li>2. Tata usaha atau kepala sekolah mengisi <i>form</i> dengan masukan sesuai yang dibutuhkan.</li> </ol> | <ol style="list-style-type: none"> <li>1. Aktor (Petugas tata usaha atau kepala sekolah) memilih menu rekap presensi siswa</li> <li>2. Sistem menampilkan seluruh presensi siswa berdasarkan tahun ajaran yang ada</li> <li>3. Aktor (Petugas tata usaha atau kepala sekolah) memilih lihat rekap</li> <li>4. Sistem akan menghitung presentasi kehadiran siswa pada kelas yang dipilih</li> <li>5. Sistem akan menampilkan semua data rekap presensi siswa pada kelas yang dipilih</li> <li>6. Petugas memilih lihat catatan untuk mengetahui detail presensi</li> <li>7. siswa (jika terdapat absen) dalam tahun ajaran tersebut</li> <li>8. Sistem menampilkan detail catatan presensi siswa.</li> <li>9. <i>Use case</i> rekap presensi selesai.</li> </ol> | Alur pengguna dan <i>basic flow</i> sudah sesuai |

Tabel 4.8 merupakan tinjauan terkait dengan harapan alur pengguna pada *use case* rekap presensi siswa.

**Tabel 4.8** Harapan Alur Pengguna *Use Case* Rekap Presensi Siswa

| Pertanyaan  | Setuju  | Netral | Tidak Setuju | Keterangan  |
|---|---|--------|--------------|---|
| Respon Sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan. | V   |        |              | Karena hasil yang ditampilkan oleh sistem sudah sesuai dengan keinginan pengguna. |
| <b>Pernyataan/Masukan dan Saran</b>                                     | Alur sudah sesuai dengan harapan pengguna dalam manajemen presensi siswa. |        |              |   |

Tabel 4.9 merupakan tinjauan terkait dengan navigasi *use case* rekap presensi siswa.

**Tabel 4.9** Tinjauan Navigasi *Use Case* Rekap Presensi Siswa

| Pertanyaan   | Setuju | Netral | Tidak Setuju | Keterangan                              |
|--|--------|--------|--------------|---|
| Saya dapat dengan mudah mengetahui posisi saya pada sistem.                                | V      |        |              | Karena posisi navigasi sudah jelas.     |
| Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi. | V      |        |              | Karena isi dari navigasi sudah jelas.   |
| Sistem mudah digunakan dengan menggunakan tombol kembali atau <i>link</i> .                | V      |        |              | Tombol dan <i>link</i> dapat digunakan. |
| Sistem tidak membuka banyak jendela <i>browser</i> ketika digunakan                        | V      |        |              | Sistem tidak membuka banyak jendela.    |

**Tabel 4.9** Tinjauan Navigasi *Use Case* Rekap Presensi Siswa (lanjutan)

|  |   |  |  |  |
|--|---|--|--|--|
| Peletakan menu atau <i>link</i> dapat dengan mudah dikenali. | V   |  |  | Karenan dengan ukuran huruf dan tombol yang jelas, ketika membuka sistem langsung mengetahui menu dari sistem. |
| <b>Masukan dan Saran</b>                                     | Navigasi sudah sesuai dengan harapan pengguna dalam rekap presensi siswa. |  |  |  |

#### 4.2 Evaluasi Antarmuka

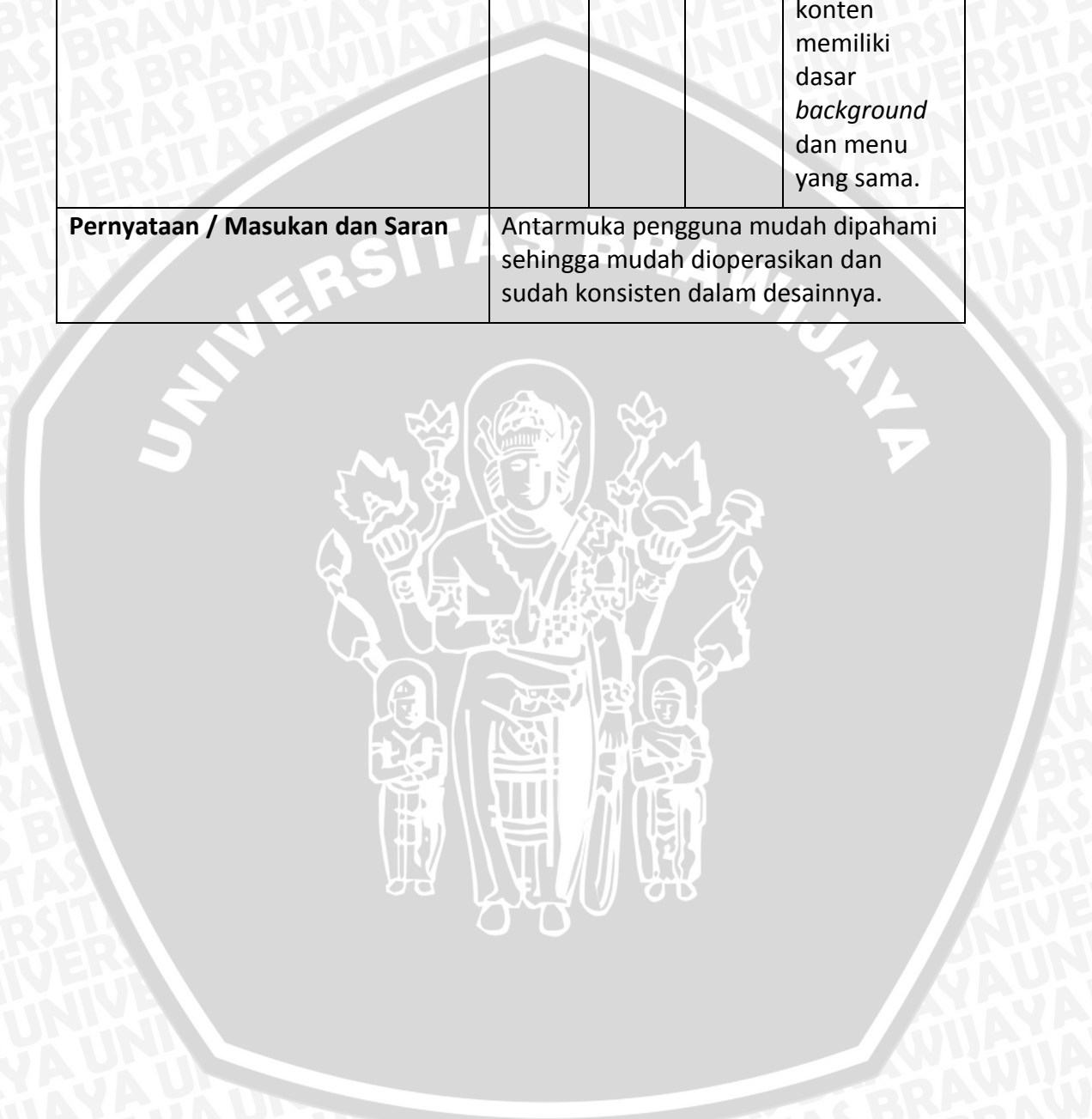
Evaluasi antarmuka pengguna yang dilakukan terhadap konsistensi dan kemudahan dalam menggunakan sistem. Berikut merupakan hasil evaluasi antarmuka prototipe Sistem Informasi Manajemen Akademik pada alur manajemen data guru, *user*, data siswa, mata pelajaran, jam pelajaran, data kelas, data pengajar, pembagian kelas, data pengajar, jadwal mata pelajaran, kegiatan, manajemen presensi siswa, wali kelas, serta rekap presensi siswa. Tabel 4.10 merupakan tinjauan antarmuka pengguna dengan alur manajemen jadwal mata pelajaran.

**Tabel 4.10** Evaluasi Antarmuka Pengguna Alur Manajemen Jadwal Mata Pelajaran

| Pertanyaan  | Setuju | Netral | Tidak Setuju | Keterangan  |
|---|--------|--------|--------------|---|
| Rancangan antarmuka pengguna sistem mudah untuk dipelajari. | V      |        |              | Karena bahasa penggunaan mudah dipahami, sehingga dapat dioperasikan oleh orang awam. |

**Tabel 4.10** Evaluasi Antarmuka Pengguna Alur Manajemen Jadwal Mata Pelajaran (lanjutan)

|  |  |  |  |  |
|--|--|--|--|--|
| Sistem memiliki antar muka pengguna yang konsisten | V  |  |  | Konsisten karena pada menu dan konten memiliki dasar <i>background</i> dan menu yang sama. |
| <b>Pernyataan / Masukan dan Saran</b>              | Antarmuka pengguna mudah dipahami sehingga mudah dioperasikan dan sudah konsisten dalam desainnya. |  |  |  |





## BAB 5 PERANCANGAN

### 5.1 Perubahan Perancangan ke Implementasi

Perubahan rancangan dilakukan untuk menyesuaikan rancangan dengan kebutuhan implementasi. Rancangan yang dimaksud adalah rancangan yang telah dibuat di penelitian Fatha (2017). Dalam penelitian tersebut terdapat beberapa hasil rancangan yang dibutuhkan dalam pembuatan perangkat lunak seperti *class diagram*, *sequence diagram*, *physical data model* dan perancangan antarmuka. Namun, rancangan tersebut tidak dapat langsung diimplementasikan kedalam sistem karena diperlukan penyesuaian dengan *framework* yang digunakan dalam pengimplementasian ataupun kebutuhan yang baru diketahui pada proses pengimplementasian.

### 5.2 Class Diagram

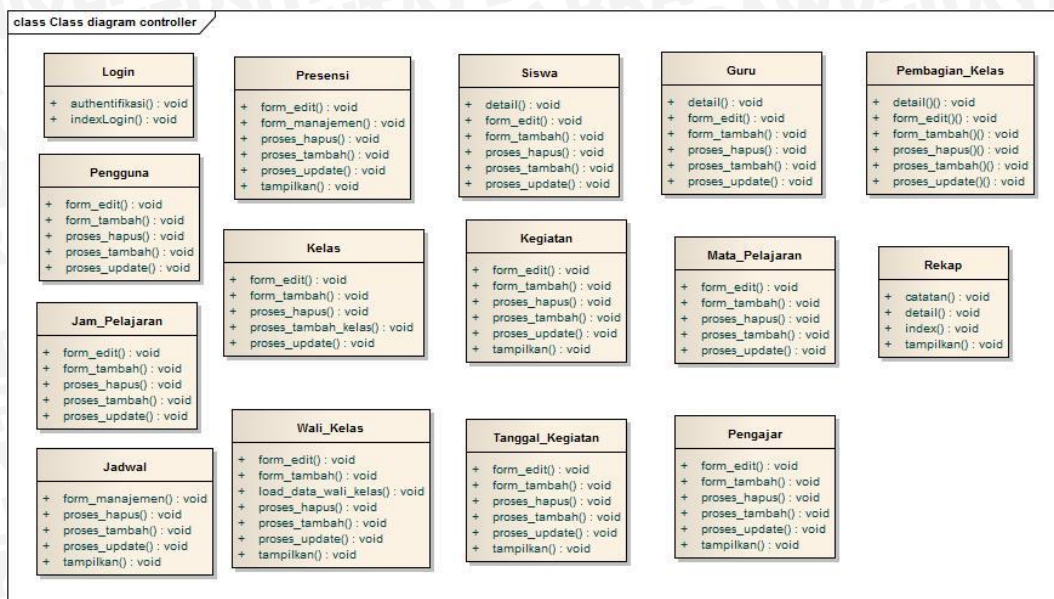
Pada sub bab ini terdapat diagram kelas perancangan dan diagram kelas implementasi yang sudah disesuaikan dengan *framework* CodeIgniter serta penjelasannya.

#### 5.2.1 Class Diagram Perancangan

Pada penelitian Fatha (2017) telah dirancang *class diagram* beorientasi objek dengan menerapkan konsep MVC (*Model, View, Controller*) secara umum atau masih murni tanpa *framework*.

Pada *class diagram* perancangan terdapat tiga *class diagram* yaitu *class controller*, *class model* serta gabungan dari *class controller*, *class model* serta *view*.

*Class controller* merupakan kelas yang memiliki sifat untuk menghubungkan antara *model* dan *view*. *Class controller* dari *class diagram* perancangan SIMAK dapat dilihat pada Gambar 5.1.



**Gambar 5.1** Class controller diagram perancangan (Fatha, 2017)

Penjelasan dari class controller diagram perancangan dapat dilihat pada Tabel 5.1.

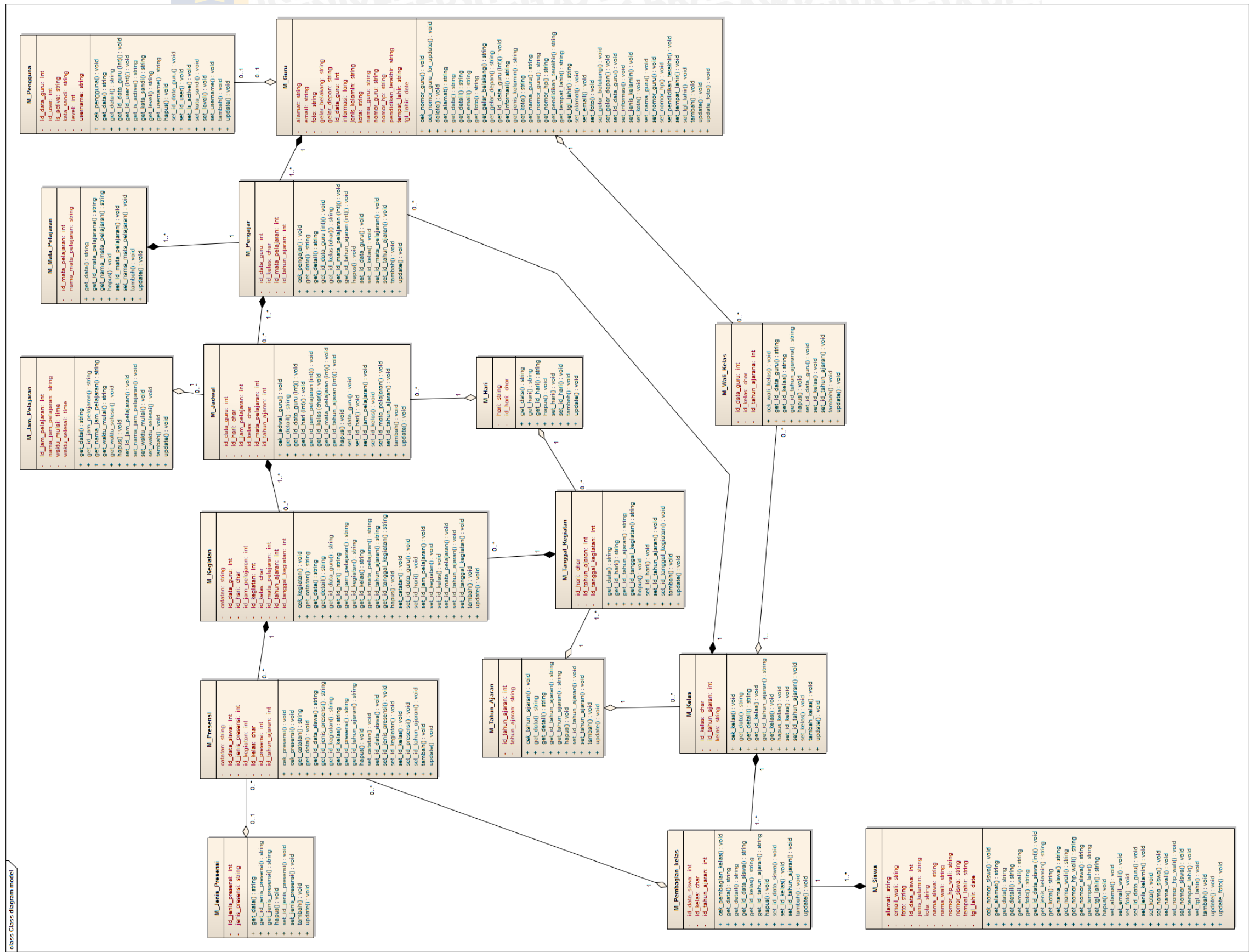
**Tabel 5.1** Penjelasan class controller diagram perancangan (Fatha, 2017)

| Nama Kelas    | Deskripsi Kelas   |
|---------------|---|
| Login         | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan validasi data untuk autentikasi pengguna |
| Pengguna      | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data pengguna                  |
| Jam_Pelajaran | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data jam pelajaran             |
| Jadwal        | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data jadwal mata pelajaran     |
| Presensi      | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data presensi siswa            |
| Kelas         | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data kelas                     |

**Tabel 5.1** Penjelasan *class controller* diagram perancangan (Fatha, 2017)  
(lanjutan)

| Nama Kelas       | Deskripsi Kelas   |
|------------------|---|
| Wali_Kelas       | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data wali kelas                              |
| Siswa            | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data siswa                                   |
| Kegiatan         | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data kegiatan                                |
| Tanggal_Kegiatan | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data tanggal kegiatan atau kalender akademik |
| Guru             | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data guru                                    |
| Mata_Pelajaran   | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data mata pelajaran                          |
| Pengajar         | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data pengajar                                |
| Pembagian_Kelas  | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan mengelola data pembagian kelas                         |
| Rekap            | Kelas yang menjembatani antara <i>entity</i> dan <i>boundary</i> yang berkaitan dengan rekap presensi   |

*Class model* merupakan kelas yang menangani hal-hal yang berhubungan dengan basis data. *Class model* dari *class diagram* perancangan SIMAK dapat dilihat pada Gambar 5.2.



Gambar 5.2 Class model diagram perancangan (Fatha, 2017)

Penjelasan dari *class model* diagram perancangan dapat dilihat pada Tabel 5.2.

**Tabel 5.2** Penjelasan *class model* diagram perancangan (Fatha, 2017)

| Nama Kelas         | Deskripsi Kelas  |
|--------------------|--|
| M_Jenis_Presensi   | Kelas yang berkaitan dengan pengelolaan data jenis presensi                  |
| M_Pembagian_Kelas  | Kelas yang berkaitan dengan pengelolaan data pembagian kelas                 |
| M_Siswa            | Kelas yang berkaitan dengan pengelolaan data siswa                           |
| M_Presensi         | Kelas yang berkaitan dengan pengelolaan data presensi                        |
| M_Tahun_Ajaran     | Kelas yang berkaitan dengan pengelolaan data tahun ajaran                    |
| M_Kelas            | Kelas yang berkaitan dengan pengelolaan data kelas                           |
| M_Kegiatan         | Kelas yang berkaitan dengan pengelolaan data kegiatan                        |
| M_Tanggal_Kegiatan | Kelas yang berkaitan dengan pengelolaan data kegiatan atau kalender akademik |
| M_Jam_Pelajaran    | Kelas yang berkaitan dengan pengelolaan data jam pelajaran                   |
| M_Jadwal           | Kelas yang berkaitan dengan pengelolaan data jadwal mata pelajaran           |
| M_Hari             | Kelas yang berkaitan dengan pengelolaan data hari                            |
| M_Wali_Kelas       | Kelas yang berkaitan dengan pengelolaan data wali kelas                      |
| M_Mata_Pelajaran   | Kelas yang berkaitan dengan pengelolaan data mata pelajaran                  |
| M_Pengajar         | Kelas yang berkaitan dengan pengelolaan data pengajar                        |
| M_Pengguna         | Kelas yang berkaitan dengan pengelolaan data pengguna                        |
| M_Guru             | Kelas yang berkaitan dengan pengelolaan data guru                            |

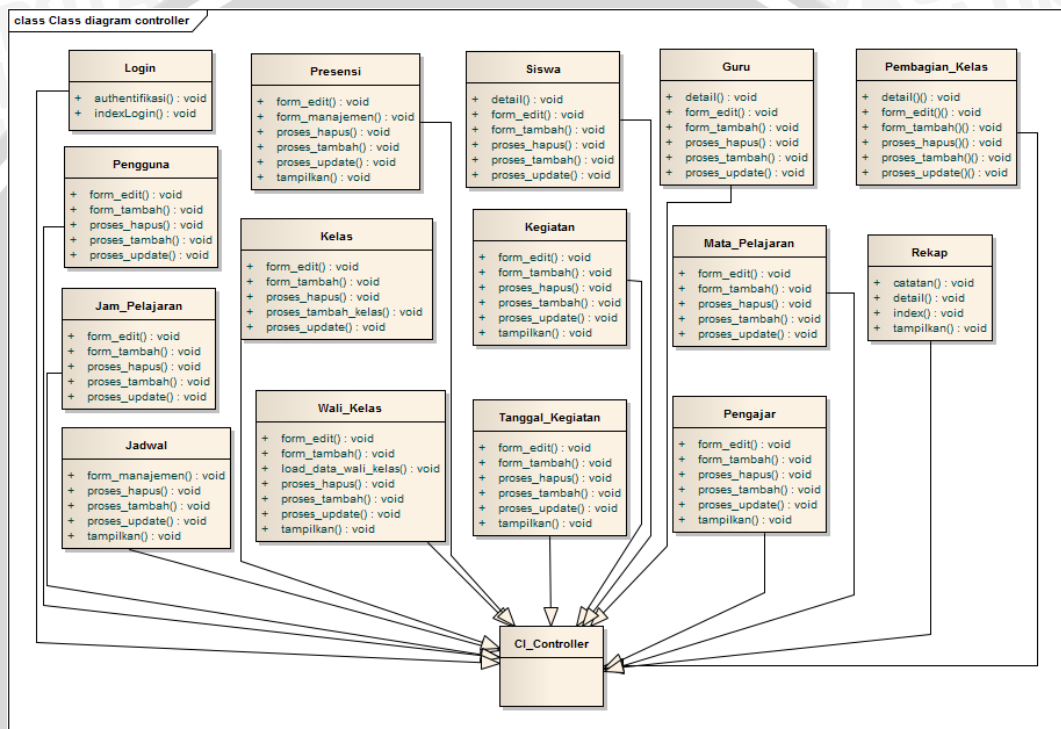
*Class* gabungan MVC merupakan *class* diagram yang menggambarkan hubungan antara *class controller*, *class model* serta *view*. *Class* gabungan MVC dari *class* diagram perancangan SIMAK dapat dilihat pada Gambar 5.3.



## 5.2.2 Class Diagram Implementasi

Pada sub bab ini dilakukan penyesuaian rancangan *class* diagram pada penelitian Fatha (2017) terhadap *framework* untuk implementasi SIMAK, yakni Codelgniter. *Class* diagram implementasi terdapat tiga *class* diagram yaitu *class controller*, *class model* serta gabungan dari *class controller*, *class model* serta *view*.

Secara umum, *class controller* dari rancangan pada penelitian Fatha (2017) mengalami perubahan yakni *class controller* pada *class* diagram implementasi merupakan *class* turunan dari *class* CI\_Controller. *Class controller* dari *class* diagram imlementasi SIMAK dapat dilihat pada Gambar 5.4.



**Gambar 5.4** Class diagram controller implementasi

*Class model* dari rancangan pada penelitian Fatha (2017) juga mengalami perubahan yakni *class model* pada *class* diagram implementasi merupakan *class* turunan dari *class* CI\_Model. *Class model* dari *class* diagram implementasi SIMAK dapat dilihat pada Gambar 5.5.







## BAB 6 IMPLEMENTASI

### 6.1 Lingkungan Implementasi

Dalam lingkungan implementasi terdapat 2 lingkungan yakni lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*).

#### 6.1.1 Lingkungan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk implementasi Sistem Informasi Manajemen Akademik adalah laptop dengan spesifikasi sebagai berikut:

1. Manufaktur : Acer
2. Model : Aspire E1-431
3. *Processor* : Intel(R) Celeron(R) CPU B820 @ 1.70GHz (2 CPUs)
4. Memori : 2048MB RAM
5. *Storage/Hard Disc* : 500GB
6. *Device pendukung* : *Mouse Digital*

#### 6.1.2 Lingkungan Perangkat Lunak (*Software*):

Perangkat lunak yang digunakan untuk implementasi Sistem Informasi Manajemen Akademik adalah sebagai berikut:

1. Sublime Text Editor 3 64-Bit
2. XAMPP V3.2.1
3. UC browser, google chrome dan mozilla firefox
4. Navicat Premium SQL

### 6.2 Hasil Implementasi

Berikut ini merupakan hasil pengimplementasian SIMAK dalam bentuk kode program dan *screenshot*.

#### 6.2.1 Kode Program

Pada sub bab ini akan menjabarkan tentang kode program dari SIMAK. Berikut ini contoh kode program dari *class* yang ada pada SIMAK.

##### 6.2.1.1 Kode Program *Class* Jadwal

*Class* jadwal merupakan *class* yang bertipe *controller*. Dalam *class* jadwal terdapat fungsi untuk menampilkan data tiap tahun, menambahkan data, mengubah data dan menghapus data. Kode program dapat dilihat pada Tabel 6.1.

Tabel 6.1 Kode program *class* jadwal

| No | Syntax  |
|----|---|
| 1  | <?php   |
| 2  | defined('BASEPATH') OR exit('No direct script access allowed');   |
| 3  |   |
| 4  | class Jadwal extends CI_Controller {  |
| 5  |   |
| 6  | function __construct(){   |
| 7  | parent::__construct();  |
| 8  | \$this->load->model('m_jadwal', 'jadwal');  |
| 9  | \$this->load->model('m_tahun_ajaran', 'tahun_ajaran');  |
| 10 | \$this->load->model('m_hari', 'hari');  |
| 11 | \$this->load-   |
| 12 | >model('m_jam_pelajaran','jam_pelajaran');  |
| 13 | \$this->load->model('m_pengajar', 'pengajar');  |
| 14 | }   |
| 15 | public function index()   |
| 16 | {   |
| 17 | \$sesi = \$this->session->userdata('simak_session');  |
| 18 |   |
| 19 | if(!empty(\$sesi)){   |
| 20 | if (\$sesi['level'] == 'tusaha') {  |
| 21 | \$header['sesi'] = \$sesi;  |
| 22 | \$header['menu_active'] = 'jadwal';   |
| 23 | \$header['sub_menu_active'] = '';   |
| 24 | \$data['sesi'] = \$sesi;  |
| 25 | \$this->jadwal->set_additional_query("WHERE is_default='true' ORDER BY h.id hari;");  |
| 26 | \$data['jadwal'] = \$this->jadwal-  |
| 27 | >get_data()->result();  |
| 28 | \$data['menu_tahun_ajaran'] = \$this-   |
| 29 | >tahun_ajaran->get_data()->result();  |
| 30 | \$this->load->view('tusaha/v_header',   |
| 31 | \$header);  |
| 32 | \$this->load-   |
| 33 | >view('tusaha/jadwal/v_index', \$data);   |
| 34 | \$this->load->view('tusaha/v_footer');  |
| 35 | }elseif (\$sesi['level'] == 'guru') {   |
| 36 | \$header['sesi'] = \$sesi;  |
| 37 | \$header['menu_active'] = 'jadwal';   |
| 38 | \$header['sub_menu_active'] = '';   |
| 39 | \$data['sesi'] = \$sesi;  |
| 40 | \$this->jadwal->set_additional_query("WHERE is_default='true' AND jd.id_data_guru='\$sesi[id_data_guru]' ORDER BY h.id hari;"); |
| 41 | \$data['jadwal'] = \$this->jadwal-  |
| 42 | >get_data()->result();  |
| 43 | \$data['menu_tahun_ajaran'] = \$this-   |
| 44 | >tahun_ajaran->get_data()->result();  |
| 45 | \$this->load->view('guru/v_header',   |
| 46 | \$header);  |
| 47 | \$this->load->view('guru/jadwal/v_index',   |
| 48 | \$data);  |
| 49 | \$this->load->view('guru/v_footer');  |
| 50 | }elseif (\$sesi['level'] == 'kepsek') {   |
| 51 | \$header['sesi'] = \$sesi;  |
| 52 | \$header['menu_active'] = 'jadwal';   |
| 53 | \$header['sub_menu_active'] = '';   |
| 54 | \$data['sesi'] = \$sesi;  |
| 55 | \$this->jadwal->set_additional_query("WHERE is_default='true' AND jd.id_data_guru='\$sesi[id_data_guru]' ORDER BY h.id hari;"); |
| 56 |   |
| 57 |   |



**Tabel 6.1** Kode program *class* jadwal (lanjutan)

| No | Syntax   |
|----|--|
| 48 | <code>\$data['jadwal'] = \$this-&gt;jadwal-&gt;get_data()-&gt;result();</code>                     |
| 49 | <code>\$data['menu_tahun_ajaran'] = \$this-&gt;tahun_ajaran-&gt;get_data()-&gt;result();</code>    |
| 50 | <code>\$this-&gt;load-&gt;view('kepsek/v_header', \$header);</code>                                |
| 51 | <code>\$this-&gt;load-&gt;view('kepsek/jadwal/v_index', \$data);</code>                            |
| 52 | <code>\$this-&gt;load-&gt;view('kepsek/v_footer');</code>  |
| 53 | <code>}else{</code>  |
| 54 | <code>\$statement = "Anda tidak memiliki hak mengakses halaman tersebut";</code>                   |
| 55 | <code>\$data_alert = \$this-&gt;alert-&gt;display('danger', \$statement);</code>                   |
| 56 | <code>\$this-&gt;session-&gt;set_flashdata(array('simak_script' =&gt; \$data_alert));</code>       |
| 57 | <code>redirect(base_url());</code>   |
| 58 | <code>}</code>   |
| 59 | <code>}else{</code>  |
| 60 | <code>\$this-&gt;load-&gt;view('v_login');</code>  |
| 61 | <code>}</code>   |
| 62 | <code>}</code>   |
| 63 | <code>public function tampilkan()</code>   |
| 64 | <code>{</code>   |
| 65 | <code>\$sesi = \$this-&gt;session-&gt;userdata('simak_session');</code>                            |
| 66 | <code>if(!empty(\$sesi)){</code>   |
| 67 | <code>if (\$sesi['level'] == 'tusaha') {</code>  |
| 68 | <code>\$header['sesi'] = \$sesi;</code>  |
| 69 | <code>\$header['menu_active'] = 'jadwal';</code>   |
| 70 | <code>\$header['sub_menu_active'] = '';</code>   |
| 71 | <code>\$data['sesi'] = \$sesi;</code>  |
| 72 | <code>\$id_tahun_ajaran = \$this-&gt;input-&gt;post('id_tahun_ajaran');</code>                     |
| 73 | <code>if(!empty(\$id_tahun_ajaran)){</code>  |
| 74 | <code>\$this-&gt;jadwal-&gt;set_additional_query("ta.id_tahun_ajaran='\$id_tahun_ajaran';")</code> |
| 75 | <code>WHERE</code>   |
| 76 | <code>\$data['jadwal'] = \$this-&gt;jadwal-&gt;get_data()-&gt;result();</code>                     |
| 77 | <code>\$data['menu_tahun_ajaran'] = \$this-&gt;tahun_ajaran-&gt;get_data()-&gt;result();</code>    |
| 78 | <code>\$data['id_tahun_ajaran_active'] = \$id_tahun_ajaran;</code>                                 |
| 79 | <code>\$this-&gt;load-&gt;view('tusaha/v_header', \$header);</code>                                |
| 80 | <code>\$this-&gt;load-&gt;view('tusaha/jadwal/v_tampilkan', \$data);</code>                        |
| 81 | <code>\$this-&gt;load-&gt;view('tusaha/v_footer');</code>  |
| 82 | <code>}else{</code>  |
| 83 | <code>redirect(base_url('jadwal'));</code>   |
| 84 | <code>}</code>   |
| 85 | <code>}elseif (\$sesi['level'] == 'guru') {</code>   |
| 86 | <code>\$header['sesi'] = \$sesi;</code>  |
| 87 | <code>\$header['menu_active'] = 'jadwal';</code>   |
| 88 | <code>}</code>   |





**Tabel 6.1** Kode program *class* jadwal (lanjutan)

| No  | Syntax  |
|-----|---|
| 126 | <code>}else{</code>   |
| 127 | <code>    \$this-&gt;load-&gt;view('v_login');</code>                                     |
| 128 | <code>  }</code>  |
| 129 | <code>}</code>  |
| 130 |   |
| 131 | <code>public function form_manajemen()</code>   |
| 132 | <code>{</code>  |
| 133 | <code>    \$sesi = \$this-&gt;session-&gt;userdata('simak_session');</code>               |
| 134 |   |
| 135 | <code>    if(!empty(\$sesi)){</code>  |
| 136 | <code>        if (\$sesi['level'] == 'tusaha') {</code>                                   |
| 137 | <code>            \$header['sesi'] = \$sesi;</code>                                       |
| 138 | <code>            \$header['menu_active'] = 'jadwal';</code>                              |
| 139 | <code>            \$header['sub_menu_active'] = '';</code>                                |
| 140 | <code>            \$data['sesi'] = \$sesi;</code>   |
| 141 | <code>            \$data['tahun_ajaran'] = \$this-&gt;</code>                             |
| 142 | <code>            &gt;tahun_ajaran-&gt;get_data()-&gt;result();</code>                    |
| 143 | <code>            \$data['hari'] = \$this-&gt;hari-&gt;get_data()-&gt;</code>             |
| 144 | <code>            &gt;result();</code>  |
| 145 | <code>            \$this-&gt;load-&gt;view('tusaha/v_header',</code>                      |
| 146 | <code>            \$header);</code>   |
| 147 | <code>            \$this-&gt;load-&gt;view('tusaha/jadwal/v_tambah', \$data);</code>      |
| 148 | <code>            \$this-&gt;load-&gt;view('tusaha/v_footer');</code>                     |
| 149 | <code>            }else{</code>   |
| 150 | <code>                \$statement = "Anda tidak memiliki hak</code>                       |
| 151 | <code>                mengakses halaman tersebut";</code>                                 |
| 152 | <code>                \$data_alert = \$this-&gt;alert-&gt;</code>                         |
| 153 | <code>                &gt;display('danger', \$statement);</code>                          |
| 154 | <code>                \$this-&gt;session-&gt;</code>                                      |
| 155 | <code>                &gt;set_flashdata(array('simak_script' =&gt; \$data_alert));</code> |
| 156 | <code>                redirect(base_url());</code>  |
| 157 | <code>            }</code>  |
| 158 | <code>            }else{</code>   |
| 159 | <code>                \$this-&gt;load-&gt;view('v_login');</code>                         |
| 160 | <code>            }</code>  |
| 161 | <code>        }</code>  |
| 162 | <code>    }</code>  |
| 163 | <code>public function proses_tambah()</code>  |
| 164 | <code>{</code>  |
| 165 | <code>    \$sesi = \$this-&gt;session-&gt;userdata('simak_session');</code>               |
| 166 |   |
| 167 | <code>    if(!empty(\$sesi)){</code>  |
| 168 | <code>        if (\$sesi['level'] == 'tusaha') {</code>                                   |
| 169 | <code>            // pengecekan jumlah jam mengajar</code>                                |
| 170 | <code>            \$get_jumlah_jam = \$this-&gt;pengajar-&gt;</code>                      |
| 171 | <code>            &gt;get_detail()-&gt;row();</code>                                      |
| 172 | <code>            \$cek_jumlah_jam = \$this-&gt;jadwal-&gt;</code>                        |
| 173 | <code>            &gt;cek_jumlah_jam()-&gt;row();</code>                                  |
| 174 | <code>            if(\$cek_jumlah_jam-&gt;jumlah_jam &lt;</code>                          |
| 175 | <code>            \$get_jumlah_jam-&gt;jumlah_jam){</code>                                |
| 176 | <code>                // pengecekan jadwal guru</code>                                    |
| 177 | <code>                \$cek_jadwal_guru = \$this-&gt;jadwal-&gt;</code>                   |
| 178 | <code>                &gt;cek_jadwal_guru()-&gt;num_rows();</code>                        |
| 179 | <code>                if(\$cek_jadwal_guru &gt; 0){</code>                                |
| 180 | <code>                    \$statement = "Data tidak</code>                                |
| 181 | <code>                    berhasil ditambahkan karena ada bentrokan jadwal guru";</code>  |

**Tabel 6.1** Kode program *class* jadwal (lanjutan)

| No  | Syntax  |
|-----|---|
| 173 | <code>\$data['alert'] = \$this-</code>  |
| 174 | <code>&gt;alert-&gt;display('danger', \$statement);</code><br><code>\$this-&gt;load-</code>   |
| 175 | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code><br><code>}else{</code>  |
| 176 | <code>\$do_insert = \$this-</code>  |
| 177 | <code>&gt;jadwal-&gt;tambah();</code><br><code>if(\$do_insert &gt; 0){</code>   |
| 178 | <code>\$statement = "Data</code><br><code>berhasil ditambahkan";</code>   |
| 179 | <code>\$data['alert']</code><br><code>= \$this-&gt;alert-&gt;display('success', \$statement);</code>  |
| 180 | <code>\$this-&gt;load-</code><br><code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>  |
| 181 | <code>}else{</code>   |
| 182 | <code>\$statement = "Data</code><br><code>tidak berhasil ditambahkan";</code>   |
| 183 | <code>\$data['alert']</code><br><code>= \$this-&gt;alert-&gt;display('danger', \$statement);</code>   |
| 184 | <code>\$this-&gt;load-</code><br><code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>  |
| 185 | <code>}</code>  |
| 186 | <code>}</code>  |
| 187 | <code>}else{</code>   |
| 188 | <code>\$statement = "Data tidak berhasil</code><br><code>ditambahkan karena alokasi jam pelajaran sudah penuh";</code>                              |
| 189 | <code>\$data['alert'] = \$this-&gt;alert-</code><br><code>&gt;display('danger', \$statement);</code>  |
| 190 | <code>\$this-&gt;load-</code><br><code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>  |
| 191 | <code>}</code>  |
| 192 | <code>}else{</code>   |
| 193 | <code>\$statement = "Anda tidak memiliki hak</code><br><code>mengakses halaman tersebut";</code>  |
| 194 | <code>\$data_alert = \$this-&gt;alert-</code><br><code>&gt;display('danger', \$statement);</code>   |
| 195 | <code>\$this-&gt;session-</code><br><code>&gt;set_flashdata(array('simak_script' =&gt; \$data_alert));</code><br><code>redirect(base_url());</code> |
| 196 | <code>}</code>  |
| 197 | <code>}</code>  |
| 198 | <code>}else{</code>   |
| 199 | <code>\$this-&gt;load-&gt;view('v_login');</code>   |
| 200 | <code>}</code>  |
| 201 | <code>}</code>  |
| 202 |   |
| 203 | <code>public function proses_update()</code>  |
| 204 | <code>{</code>  |
| 205 | <code>\$sesi = \$this-&gt;session-&gt;userdata('simak_session');</code>   |
| 206 |   |
| 207 | <code>if(!empty(\$sesi)){</code>  |
| 208 | <code>if (\$sesi['level'] == 'tusaha') {</code>   |
| 209 | <code>// pengecekan jumlah jam mengajar</code><br><code>\$get_jumlah_jam = \$this-&gt;pengajar-</code>  |
| 210 | <code>&gt;get_detail()-&gt;row();</code>  |
| 211 | <code>\$cek_jumlah_jam = \$this-&gt;jadwal-</code>  |
| 212 | <code>&gt;cek_jumlah_jam()-&gt;row();</code>  |
| 213 | <code>if(\$cek_jumlah_jam-&gt;jumlah_jam &lt;</code><br><code>\$get_jumlah_jam-&gt;jumlah_jam){</code>  |
| 214 | <code>// pengecekan jadwal guru</code>  |

Tabel 6.1 Kode program *class* jadwal (lanjutan)

| No  | Syntax   |
|-----|--|
| 215 | <code>\$cek_jadwal_guru = \$this-&gt;jadwal-</code>                          |
| 216 | <code>&gt;cek_jadwal_guru()-&gt;num_rows();</code>                           |
| 217 | <code>if(\$cek_jadwal_guru &gt; 0){</code>                                   |
| 218 | <code>    \$statement = "Data tidak</code>                                   |
|     | <code>berhasil diubah karena ada bentrokan jadwal guru";</code>              |
| 219 | <code>    \$data['alert'] = \$this-</code>                                   |
|     | <code>&gt;alert-&gt;display('danger', \$statement);</code>                   |
| 220 | <code>    \$this-&gt;load-</code>  |
|     | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>                      |
| 221 | <code>    }else{</code>  |
| 222 | <code>        \$do_insert = \$this-</code>                                   |
|     | <code>&gt;jadwal-&gt;update();</code>  |
| 223 | <code>        if(\$do_insert &gt; 0){</code>                                 |
| 224 | <code>            \$statement = "Data</code>                                 |
|     | <code>berhasil diubah";</code>   |
| 225 | <code>            \$data['alert']</code>                                     |
|     | <code>        = \$this-&gt;alert-&gt;display('success', \$statement);</code> |
| 226 | <code>            \$this-&gt;load-</code>                                    |
|     | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>                      |
| 227 | <code>        }else{</code>  |
| 228 | <code>            \$statement = "Data</code>                                 |
|     | <code>tidak berhasil diubah";</code>   |
| 229 | <code>            \$data['alert']</code>                                     |
|     | <code>        = \$this-&gt;alert-&gt;display('danger', \$statement);</code>  |
| 230 | <code>            \$this-&gt;load-</code>                                    |
|     | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>                      |
| 231 | <code>        }</code>   |
| 232 | <code>    }</code>   |
| 233 | <code>    }else{</code>  |
| 234 | <code>        \$statement = "Data tidak berhasil</code>                      |
|     | <code>ditambahkan karena alokasi jam pelajaran sudah penuh";</code>          |
| 235 | <code>        \$data['alert'] = \$this-&gt;alert-</code>                     |
|     | <code>&gt;display('danger', \$statement);</code>                             |
| 236 | <code>        \$this-&gt;load-</code>  |
|     | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code>                      |
| 237 | <code>    }</code>   |
| 238 | <code>    }else{</code>  |
| 239 | <code>        \$statement = "Anda tidak memiliki hak</code>                  |
|     | <code>mengakses halaman tersebut";</code>                                    |
| 240 | <code>        \$data_alert = \$this-&gt;alert-</code>                        |
|     | <code>&gt;display('danger', \$statement);</code>                             |
| 241 | <code>        \$this-&gt;session-</code>                                     |
|     | <code>&gt;set_flashdata(array('simak_script' =&gt; \$data_alert));</code>    |
| 242 | <code>        redirect(base_url());</code>                                   |
| 243 | <code>    }</code>   |
| 244 | <code>    }else{</code>  |
| 245 | <code>        \$this-&gt;load-&gt;view('v_login');</code>                    |
| 246 | <code>    }</code>   |
| 247 | <code>}</code>   |
| 248 |  |
| 249 | <code>public function proses_hapus()</code>                                  |
| 250 | <code>{</code>   |
| 251 | <code>    \$sesi = \$this-&gt;session-&gt;userdata('simak_session');</code>  |
| 252 |  |
| 253 | <code>    if(!empty(\$sesi)){</code>   |
| 254 | <code>        if (\$sesi['level'] == 'tusaha') {</code>                      |
| 255 | <code>            \$do_delete = \$this-&gt;jadwal-&gt;hapus();</code>        |
| 256 | <code>            if(\$do_delete &gt; 0){</code>                             |



**Tabel 6.1** Kode program *class* jadwal (lanjutan)

| No  | Syntax  |
|-----|---|
| 257 | <code>\$statement = "Data berhasil</code>   |
| 258 | <code>dihapus";</code><br><code>\$data['alert'] = \$this-</code>  |
| 259 | <code>&gt;alert-&gt;display('success', \$statement);</code><br><code>\$this-&gt;load-</code>                    |
| 260 | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code><br><code>}else{</code>                                  |
| 261 | <code>\$statement = "Data tidak berhasil</code>   |
| 262 | <code>dihapus";</code><br><code>\$data['alert'] = \$this-</code>  |
| 263 | <code>&gt;alert-&gt;display('danger', \$statement);</code><br><code>\$this-&gt;load-</code>                     |
| 264 | <code>&gt;view('tusaha/jadwal/v_alert', \$data);</code><br><code>}</code>                                       |
| 265 | <code>}else{</code>   |
| 266 | <code>\$statement = "Anda tidak memiliki hak</code><br><code>mengakses halaman tersebut";</code>                |
| 267 | <code>\$data_alert = \$this-&gt;alert-</code>   |
| 268 | <code>&gt;display('danger', \$statement);</code><br><code>\$this-&gt;session-</code>                            |
| 269 | <code>&gt;set_flashdata(array('simak_script' =&gt; \$data_alert));</code><br><code>redirect(base_url());</code> |
| 270 | <code>}</code>  |
| 271 | <code>}else{</code>   |
| 272 | <code>\$this-&gt;load-&gt;view('v_login');</code>   |
| 273 | <code>}</code>  |
| 274 | <code>}</code>  |

**6.2.1.2 Kode Program Class M\_Jadwal**

*Class* *m\_jadwal* merupakan *class* yang bertipe *model*. Dalam *class* *m\_jadwal* terdapat fungsi untuk mengolah (menambah, mengubah, menghapus dan menampilkan) data yang berhubungan dengan data jadwal. Kode program dapat dilihat pada Table 6.2.

**Tabel 6.2** Kode program *class* *m\_jadwal*

| No | Syntax   |
|----|--|
| 1  | <code>&lt;?php</code>  |
| 2  | <code>defined('BASEPATH') OR exit('No direct script access allowed');</code> |
| 3  |  |
| 4  | <code>class M_Jadwal extends CI_Model {</code>                               |
| 5  |  |
| 6  | <code>private \$id_data_guru;</code>   |
| 7  | <code>private \$id_kelas;</code>   |
| 8  | <code>private \$id_tahun_ajaran;</code>                                      |
| 9  | <code>private \$id_mata_pelajaran;</code>                                    |
| 10 | <code>private \$id_hari;</code>  |
| 11 | <code>private \$id_jam_pelajaran;</code>                                     |
| 12 | <code>private \$additional_query;</code>                                     |
| 13 |  |
| 14 | <code>function set_id_data_guru(\$id_data_guru)</code>                       |
| 15 | <code>{</code>   |
| 16 | <code>\$this-&gt;id_data_guru = \$id_data_guru;</code>                       |
| 17 | <code>}</code>   |
| 18 |  |
| 19 | <code>public function set id kelas(\$id kelas)</code>                        |



Tabel 6.2 Kode program *class m\_jadwal* (lanjutan)

| No | Syntax   |
|----|--|
| 20 | {  |
| 21 | \$this->id_kelas = \$id_kelas;                             |
| 22 | }  |
| 23 |  |
| 24 | public function set_id_tahun_ajaran(\$id_tahun_ajaran)     |
| 25 | {  |
| 26 | \$this->id_tahun_ajaran = \$id_tahun_ajaran;               |
| 27 | }  |
| 28 |  |
| 29 | public function set_id_mata_pelajaran(\$id_mata_pelajaran) |
| 30 | {  |
| 31 | \$this->id_mata_pelajaran = \$id_mata_pelajaran;           |
| 32 | }  |
| 33 |  |
| 34 | public function set_id_hari(\$id_hari)                     |
| 35 | {  |
| 36 | \$this->id_hari = \$id_hari;                               |
| 37 | }  |
| 38 |  |
| 39 | public function set_id_jam_pelajaran(\$id_jam_pelajaran)   |
| 40 | {  |
| 41 | \$this->id_jam_pelajaran = \$id_jam_pelajaran;             |
| 42 | }  |
| 43 |  |
| 44 | public function set_additional_query(\$additional_query)   |
| 45 | {  |
| 46 | \$this->additional_query = \$additional_query;             |
| 47 | }  |
| 48 |  |
| 49 | public function get_id_data_guru()                         |
| 50 | {  |
| 51 | return \$this->id_data_guru;                               |
| 52 | }  |
| 53 |  |
| 54 | public function get_id_kelas()                             |
| 55 | {  |
| 56 | return \$this->id_kelas;                                   |
| 57 | }  |
| 58 |  |
| 59 | public function get_id_tahun_ajaran()                      |
| 60 | {  |
| 61 | return \$this->id_tahun_ajaran;                            |
| 62 | }  |
| 63 |  |
| 64 | public function get_id_mata_pelajaran()                    |
| 65 | {  |
| 66 | return \$this->id_mata_pelajaran;                          |
| 67 | }  |
| 68 |  |
| 69 | public function get_id_hari()                              |
| 70 | {  |
| 71 | return \$this->id_hari;                                    |
| 72 | }  |
| 73 |  |
| 74 | public function get_id_jam_pelajaran()                     |
| 75 | {  |
| 76 | return \$this->id_jam_pelajaran;                           |
| 77 | }  |
| 78 |  |

Tabel 6.2 Kode program *class m\_jadwal* (lanjutan)

| No  | Syntax  |
|-----|---|
| 79  | <code>public function get_data()</code>                           |
| 80  | <code>{</code>  |
| 81  | <code>    return \$this-&gt;db-&gt;query("SELECT DISTINCT</code>  |
| 82  | <code>        h.id_hari,</code>                                   |
| 83  | <code>        h.hari,</code>                                      |
| 84  | <code>        jd.id_tahun_ajaran,</code>                          |
| 85  | <code>        ta.tahun_ajaran,</code>                             |
| 86  | <code>        jd.id_data_guru,</code>                             |
| 87  | <code>        dg.gelar_depan,</code>                              |
| 88  | <code>        dg.nama_guru,</code>                                |
| 89  | <code>        dg.gelar_belakang,</code>                           |
| 90  | <code>        jd.id_kelas,</code>                                 |
| 91  | <code>        k.keterangan,</code>                                |
| 92  | <code>        jd.id_mata_pelajaran,</code>                        |
| 93  | <code>        mp.nama_mata_pelajaran,</code>                      |
| 94  | <code>        jd.id_jam_pelajaran,</code>                         |
| 95  | <code>        jp.nama_jam_pelajaran,</code>                       |
| 96  | <code>        jp.waktu_mulai,</code>                              |
| 97  | <code>        jp.waktu_selesai</code>                             |
| 98  | <code>    FROM</code>   |
| 99  | <code>    jadwal jd</code>  |
| 100 | <code>    JOIN tahun_ajaran ta ON ta.id_tahun_ajaran =</code>     |
| 101 | <code>jd.id_tahun_ajaran</code>                                   |
| 102 | <code>    JOIN data_guru dg ON dg.id_data_guru =</code>           |
| 103 | <code>jd.id_data_guru</code>                                      |
| 104 | <code>    JOIN kelas k ON k.id_kelas = jd.id_kelas</code>         |
| 105 | <code>    JOIN mata_pelajaran mp ON mp.id_mata_pelajaran</code>   |
| 106 | <code>= jd.id_mata_pelajaran</code>                               |
| 107 | <code>    JOIN hari h ON h.id_hari = jd.id_hari</code>            |
| 108 | <code>    JOIN jam_pelajaran jp ON jp.id_jam_pelajaran =</code>   |
| 109 | <code>jd.id_jam_pelajaran</code>                                  |
| 110 | <code>    \$this-&gt;additional_query");</code>                   |
| 111 | <code>    }</code>  |
| 112 | <code>public function get_detail()</code>                         |
| 113 | <code>{</code>  |
| 114 | <code>    return \$this-&gt;db-&gt;query(</code>                  |
| 115 | <code>        "SELECT</code>                                      |
| 116 | <code>            h.id_hari,</code>                               |
| 117 | <code>            h.hari,</code>                                  |
| 118 | <code>            jd.id_tahun_ajaran,</code>                      |
| 119 | <code>            ta.tahun_ajaran,</code>                         |
| 120 | <code>            jd.id_data_guru,</code>                         |
| 121 | <code>            dg.gelar_depan,</code>                          |
| 122 | <code>            dg.nama_guru,</code>                            |
| 123 | <code>            dg.gelar_belakang,</code>                       |
| 124 | <code>            jd.id_kelas,</code>                             |
| 125 | <code>            k.keterangan,</code>                            |
| 126 | <code>            jd.id_mata_pelajaran,</code>                    |
| 127 | <code>            mp.nama_mata_pelajaran,</code>                  |
| 128 | <code>            jd.id_jam_pelajaran,</code>                     |
| 129 | <code>            jp.nama_jam_pelajaran,</code>                   |
| 130 | <code>            jp.waktu_mulai,</code>                          |
| 131 | <code>            jp.waktu_selesai</code>                         |
| 132 | <code>        FROM</code>   |
| 133 | <code>        jadwal jd</code>                                    |
| 134 | <code>        JOIN tahun_ajaran ta ON ta.id_tahun_ajaran =</code> |
| 135 | <code>jd.id_tahun_ajaran</code>                                   |
| 136 | <code>        JOIN data_guru dg ON dg.id_data_guru =</code>       |
| 137 | <code>jd.id_data_guru</code>                                      |

Tabel 6.2 Kode program *class m\_jadwal* (lanjutan)

| No  | Syntax  |
|-----|---|
| 127 | JOIN kelas k ON k.id_kelas = jd.id_kelas  |
| 128 | JOIN mata_pelajaran mp ON mp.id_mata_pelajaran<br>= jd.id_mata_pelajaran  |
| 129 | JOIN hari h ON h.id_hari = jd.id_hari   |
| 130 | JOIN jam_pelajaran jp ON jp.id_jam_pelajaran =<br>jd.id_jam_pelajaran   |
| 131 | WHERE jd.id_jam_pelajaran='\$this-><br>>id_jam_pelajaran'   |
| 132 | AND jd.id_tahun_ajaran='\$this-><br>>id_tahun_ajaran'   |
| 133 | AND jd.id_kelas='\$this->id_kelas'  |
| 134 | AND jd.id_hari='\$this->id_hari';");  |
| 135 | }   |
| 136 |   |
| 137 | public function cek_jadwal()  |
| 138 | {   |
| 139 | return \$this->db->query(<br>140 "SELECT * FROM jadwal<br>141 WHERE id_tahun_ajaran='\$this->id_tahun_ajaran'<br>142 AND id_kelas='\$this->id_kelas'<br>143 AND id_hari='\$this->id_hari'<br>144 AND id_jam_pelajaran='\$this->id_jam_pelajaran'<br>145 \$this->additional_query;");                                      |
| 146 | }   |
| 147 |   |
| 148 | public function cek_jadwal_guru()   |
| 149 | {   |
| 150 | return \$this->db->query(<br>151 "SELECT * FROM jadwal<br>152 WHERE id_tahun_ajaran='\$this->id_tahun_ajaran'<br>153 AND id_hari='\$this->id_hari'<br>154 AND id_jam_pelajaran='\$this->id_jam_pelajaran'<br>155 AND id_data_guru='\$this->id_data_guru';");  |
| 156 | }   |
| 157 |   |
| 158 | public function cek_jumlah_jam()  |
| 159 | {   |
| 160 | return \$this->db->query(<br>161 "SELECT count(*) as jumlah_jam FROM jadwal<br>162 WHERE id_tahun_ajaran='\$this->id_tahun_ajaran'<br>163 AND id_kelas='\$this->id_kelas'<br>164 AND id_mata_pelajaran='\$this-><br>>id_mata_pelajaran';"<br>165 );   |
| 166 | }   |
| 167 |   |
| 168 | public function cek_jadwal_for_update()   |
| 169 | {   |
| 170 | return \$this->db->query(<br>171 "SELECT * FROM jadwal<br>172 WHERE id_data_guru = '\$this->id_data_guru'<br>173 AND id_tahun_ajaran='\$this->id_tahun_ajaran'<br>174 AND id_kelas='\$this->id_kelas'<br>175 AND id_mata_pelajaran='\$this-><br>>id_mata_pelajaran'<br>176 AND id_data_guru <> '\$this->id_data_guru';"); |
| 177 | }   |
| 178 |   |
| 179 | public function tambah()  |
| 180 | {   |

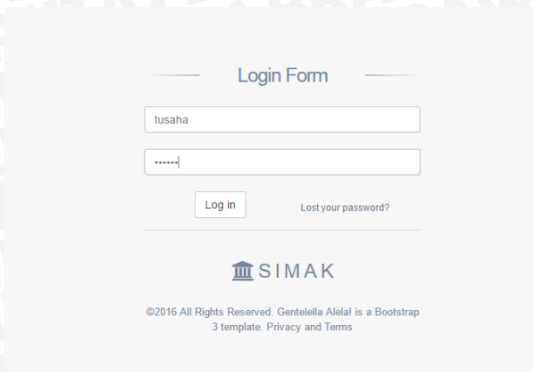
Tabel 6.2 Kode program *class m\_jadwal* (lanjutan)

| No  | Syntax  |
|-----|---|
| 181 | <code>\$data = array(</code>  |
| 182 | <code>    'id_tahun_ajaran' =&gt; \$this-&gt;id_tahun_ajaran,</code>      |
| 183 | <code>    'id_kelas' =&gt; \$this-&gt;id_kelas,</code>                    |
| 184 | <code>    'id_hari' =&gt; \$this-&gt;jadwal-&gt;id_hari,</code>           |
| 185 | <code>    'id_jam_pelajaran' =&gt; \$this-</code>                         |
|     | <code>&gt;id_jam_pelajaran,</code>  |
| 186 | <code>    'id_data_guru' =&gt; \$this-&gt;id_data_guru,</code>            |
| 187 | <code>    'id_mata_pelajaran' =&gt; \$this-</code>                        |
|     | <code>&gt;id_mata_pelajaran</code>  |
| 188 | <code>);</code>   |
| 189 | <code>return \$this-&gt;db-&gt;insert('jadwal', \$data);</code>           |
| 190 | <code>}</code>  |
| 191 |   |
| 192 | <code>public function update()</code>                                     |
| 193 | <code>{</code>  |
| 194 | <code>    \$data = array(</code>  |
| 195 | <code>        'id_data_guru' =&gt; \$this-&gt;id_data_guru,</code>        |
| 196 | <code>        'id_mata_pelajaran' =&gt; \$this-</code>                    |
|     | <code>&gt;id_mata_pelajaran</code>  |
| 197 | <code>    );</code>   |
| 198 | <code>    \$where = array(</code>   |
| 199 | <code>        'id_tahun_ajaran' =&gt; \$this-&gt;id_tahun_ajaran,</code>  |
| 200 | <code>        'id_kelas' =&gt; \$this-&gt;id_kelas,</code>                |
| 201 | <code>        'id_hari' =&gt; \$this-&gt;jadwal-</code>                   |
|     | <code>&gt;id_hari,</code>   |
| 202 | <code>        'id_jam_pelajaran' =&gt; \$this-&gt;id_jam_pelajaran</code> |
| 203 | <code>    );</code>   |
| 204 | <code>return \$this-&gt;db-&gt;update("jadwal", \$data, \$where);</code>  |
| 205 | <code>}</code>  |
| 206 |   |
| 207 | <code>public function hapus()</code>                                      |
|     | <code>{</code>  |
| 208 | <code>    \$where = array(</code>   |
| 209 | <code>        'id_tahun_ajaran' =&gt; \$this-&gt;id_tahun_ajaran,</code>  |
| 210 | <code>        'id_kelas' =&gt; \$this-&gt;id_kelas,</code>                |
| 211 | <code>        'id_hari' =&gt; \$this-&gt;jadwal-</code>                   |
|     | <code>&gt;id_hari,</code>   |
| 212 | <code>        'id_jam_pelajaran' =&gt; \$this-&gt;id_jam_pelajaran</code> |
| 213 | <code>    );</code>   |
| 214 | <code>return \$this-&gt;db-&gt;delete("jadwal", \$where);</code>          |
| 215 | <code>}</code>  |
| 216 | <code>}</code>  |
| 217 |   |

## 6.2.2 Screenshot

### 6.2.2.1 Halaman Identifikasi Pengguna

Halaman *login* merupakan halaman pertama yang akan ditampilkan ketika *web* diakses dan belum dalam keadaan *login*. Halaman *login* dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman *login* hasil implementasi dapat dilihat pada Gambar 6.1.



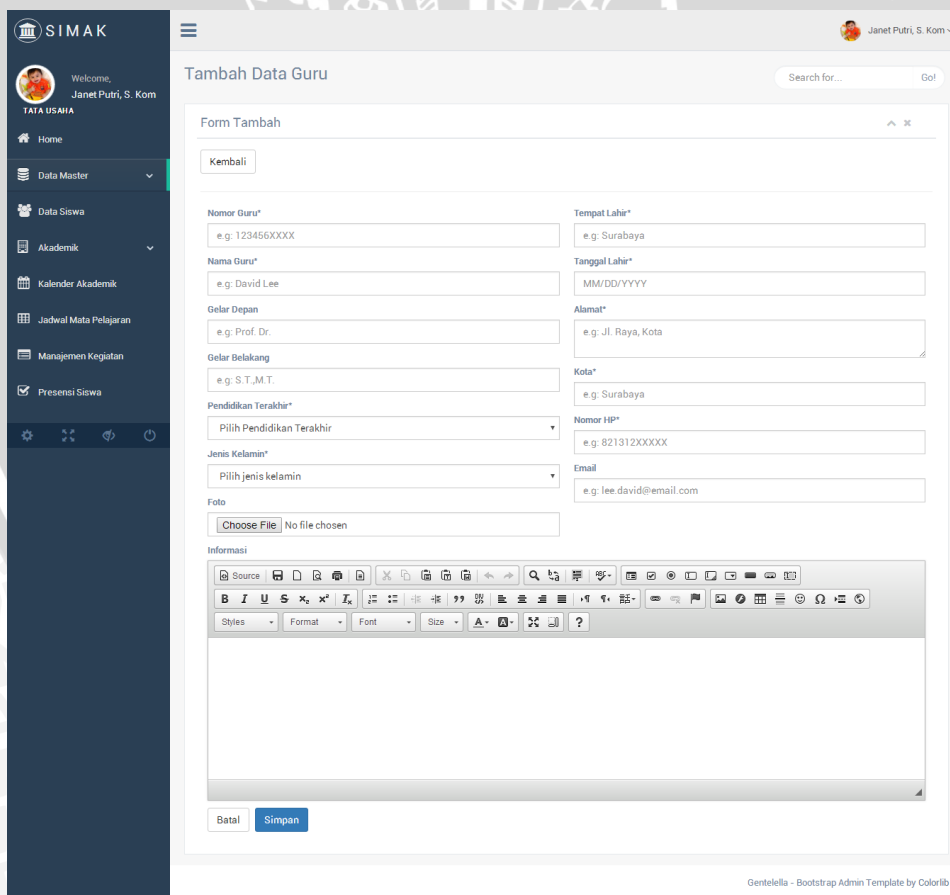
Gambar 6.1 Halaman *Login* hasil implemetasi

### 6.2.2.2 Halaman Manajemen Data Guru

Fitur manajemen data guru merupakan fitur yang hanya dapat diakses oleh *user* tata usaha. Fitur manajemen data guru terdapat 4 fungsional, yakni:

a. Menambah data guru

Halaman untuk menambah data guru dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.2.

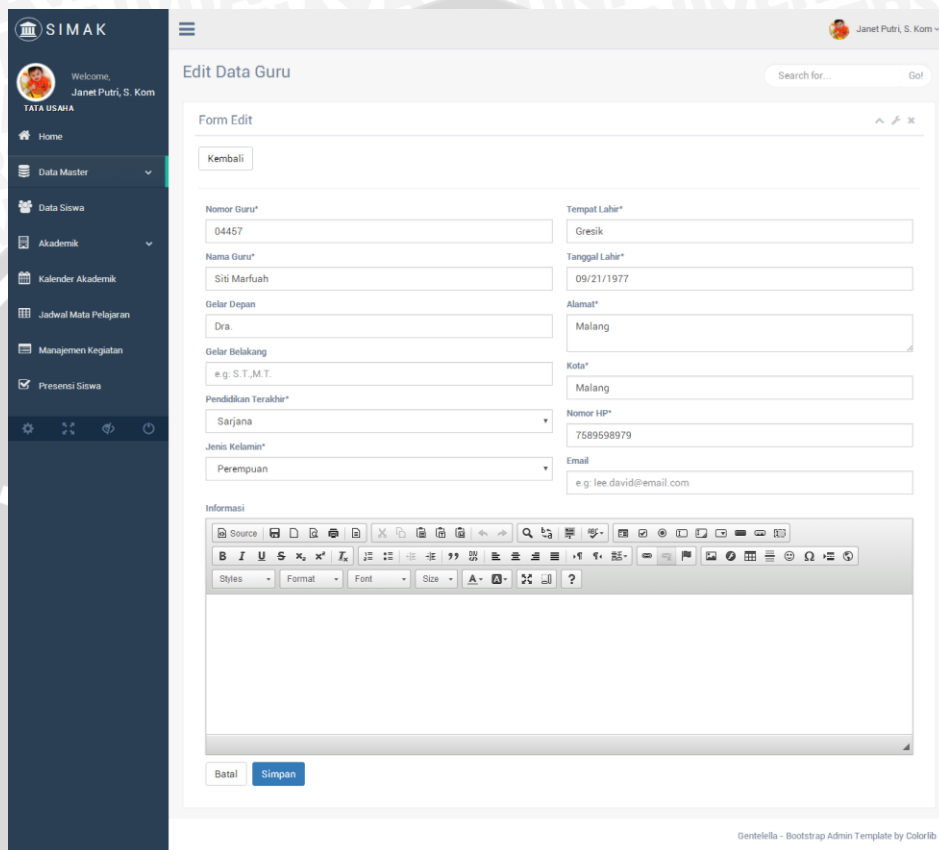


Gambar 6.2 Halaman menambah data guru hasil implemetasi

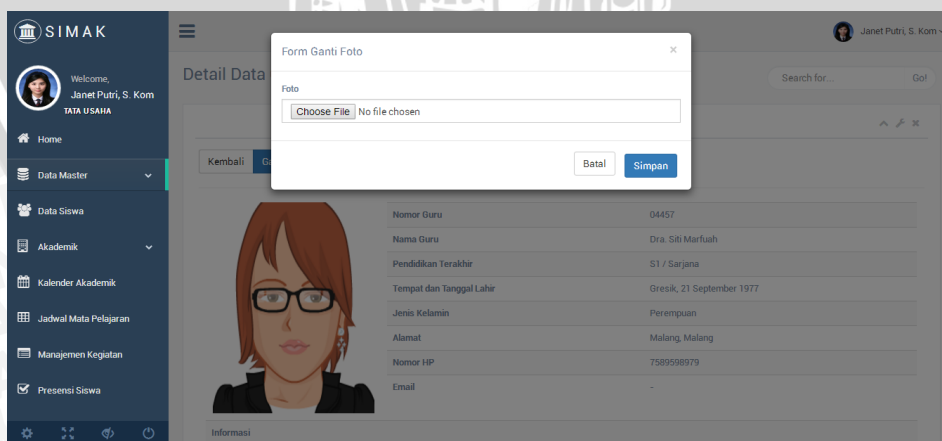


b. Mengubah data guru

Halaman untuk mengubah data guru dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman untuk mengubah data guru terdapat 2 halaman, yakni halaman mengubah detail guru dan halaman mengubah foto guru. Halaman hasil implementasi dapat dilihat pada Gambar 6.3 dan 6.4.



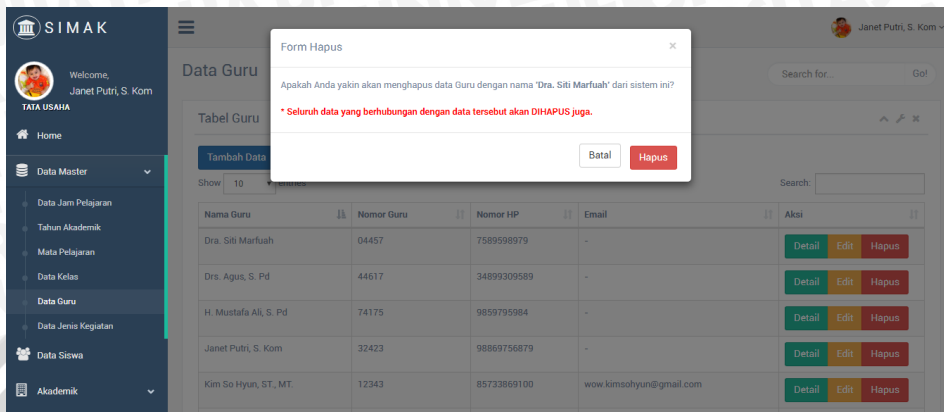
Gambar 6.3 Halaman mengubah detail guru hasil implemetasi



Gambar 6.4 Halaman mengubah foto guru hasil implemetasi

c. Menghapus data guru

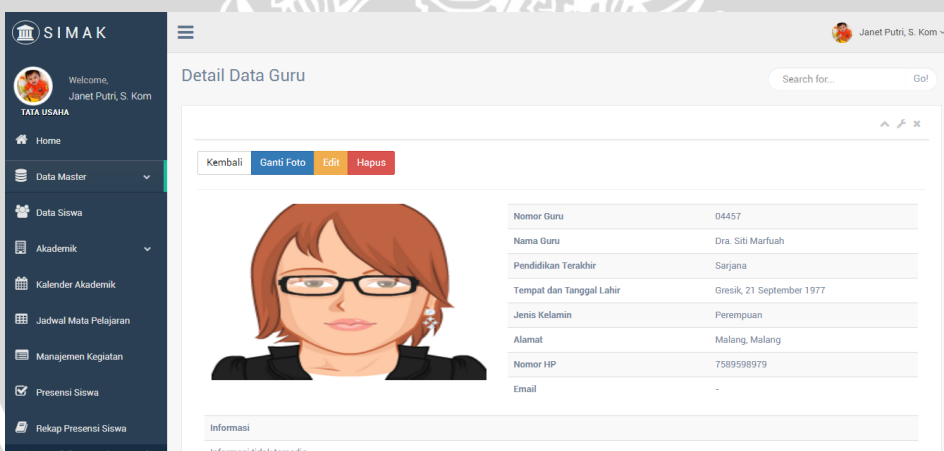
Halaman untuk menghapus data guru dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.5.



Gambar 6.5 Halaman menghapus data guru hasil implemetasi

d. Melihat detail guru

Halaman untuk melihat detail guru dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.6.



Gambar 6.6 Halaman melihat detail guru hasil implemetasi

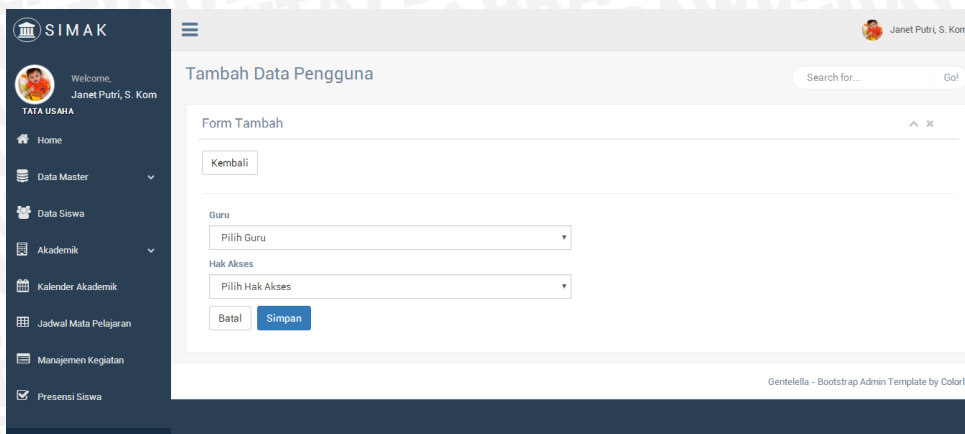
6.2.2.3 Halaman Manajemen User

Fitur manajemen *user* merupakan fitur yang hanya dapat diakses oleh *user* tata usaha. Fitur manajemen *user* terdapat 3 fungsional, yakni:

a. Menambah *user*

Halaman untuk menambah *user* dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.7.

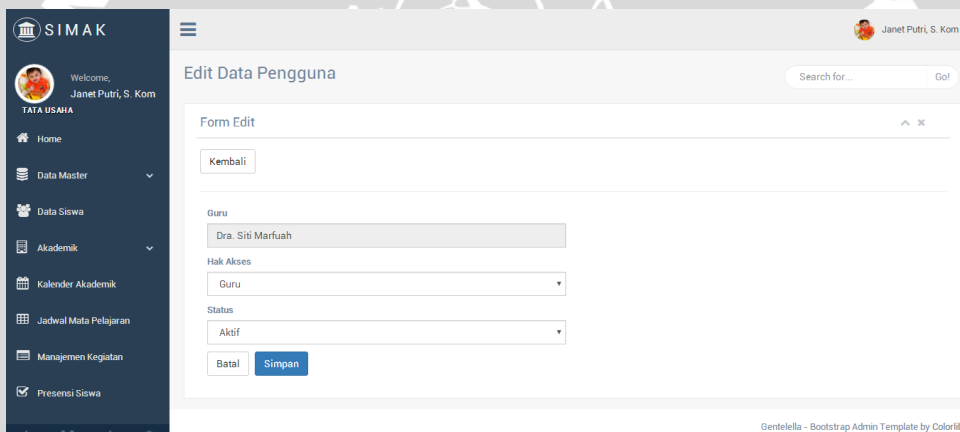




**Gambar 6.7** Halaman menambah *user* hasil implemetasi

b. Mengubah *user*

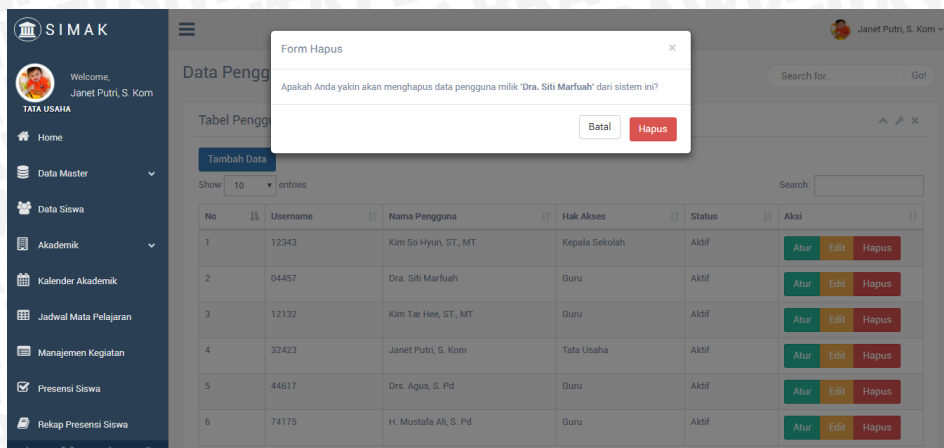
Halaman untuk mengubah *user* dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.8.



**Gambar 6.8** Halaman mengubah *user* hasil implemetasi

c. Menghapus *user*

Halaman untuk menghapus *user* dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.9.



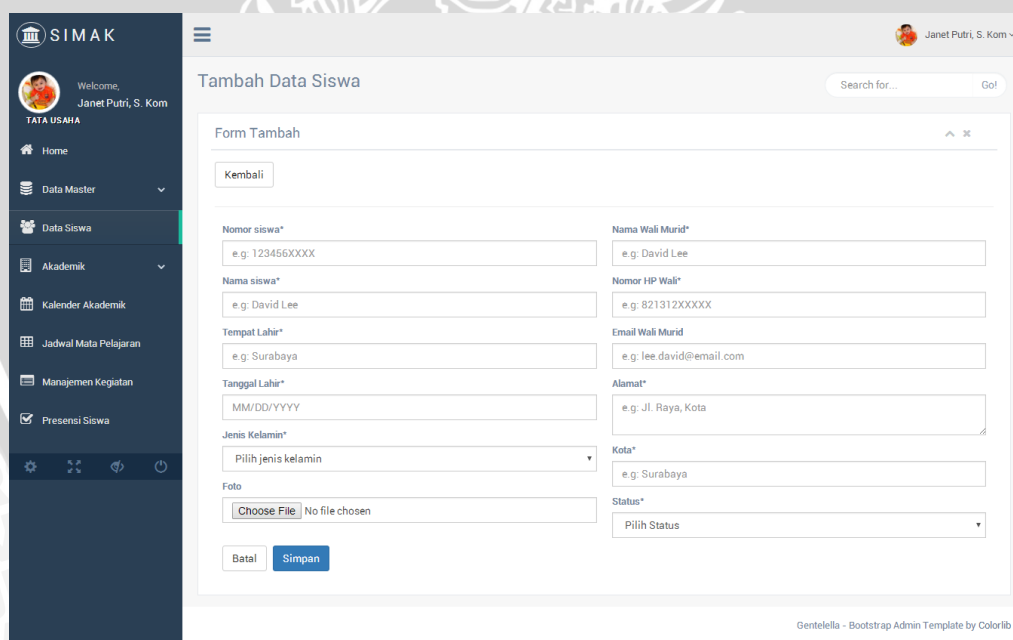
**Gambar 6.9** Halaman menghapus user hasil implemetasi

### 6.2.2.4 Halaman Manajemen Data Siswa

Fitur manajemen data siswa merupakan fitur yang hanya dapat diakses oleh user tata usaha. Fitur manajemen data siswa terdapat 4 fungsional, yakni:

a. Menambah data siswa

Halaman untuk menambah data siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.10.



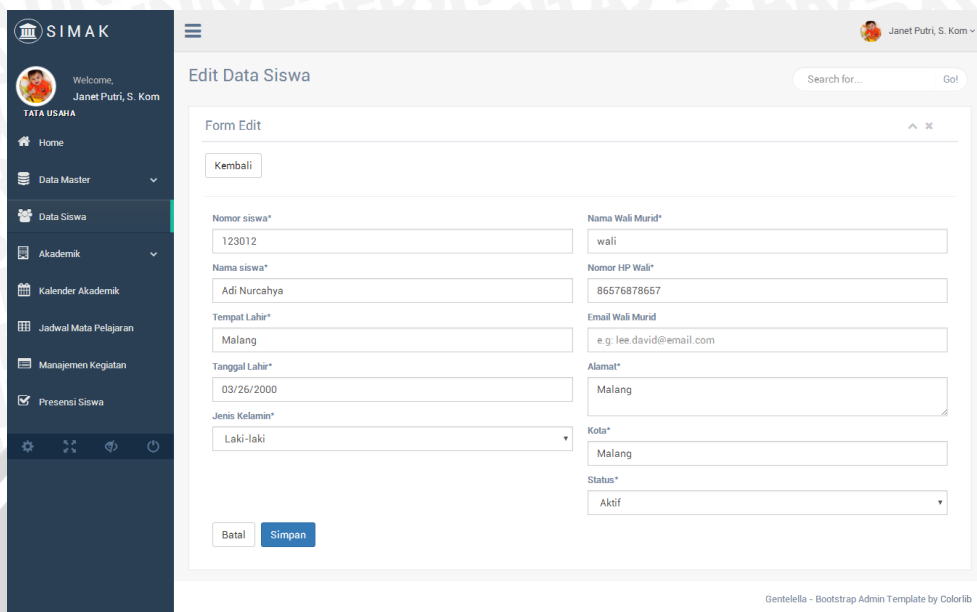
**Gambar 6.10** Halaman menambah data siswa hasil implemetasi

b. Mengubah data siswa

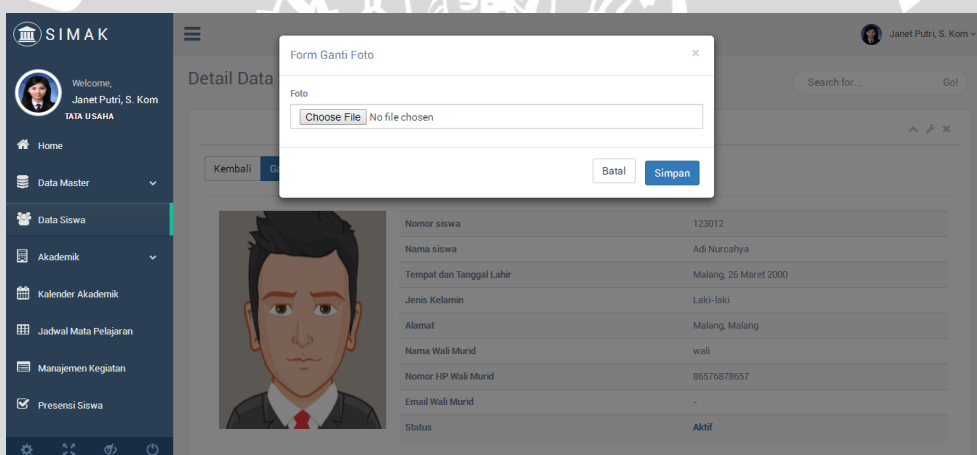
Halaman untuk mengubah data siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman untuk mengubah data siswa terdapat 2 halaman, yakni halaman mengubah detail siswa dan



halaman mengubah foto siswa. Halaman hasil implementasi dapat dilihat pada Gambar 6.11 dan 6.12.



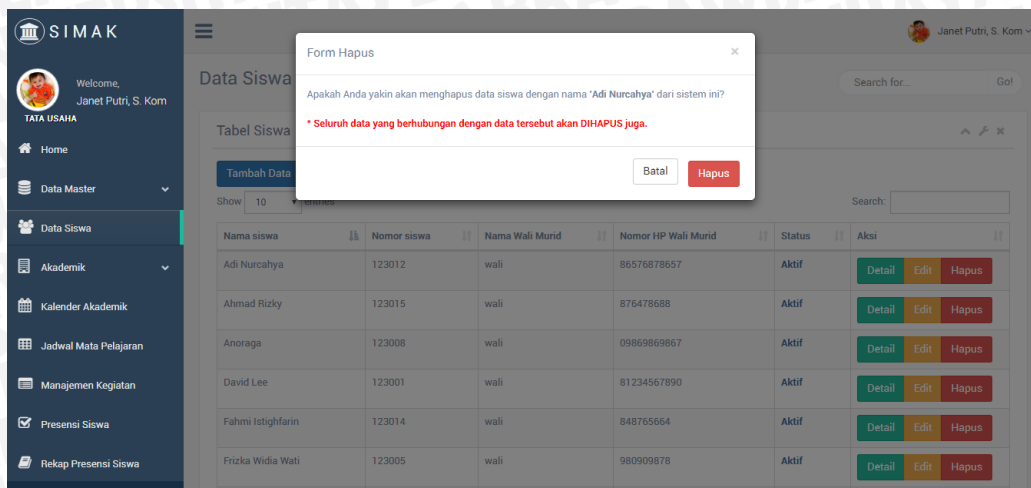
**Gambar 6.11** Halaman mengubah detail siswa hasil implemetasi



**Gambar 6.12** Halaman mengubah foto siswa hasil implemetasi

c. Menghapus data siswa

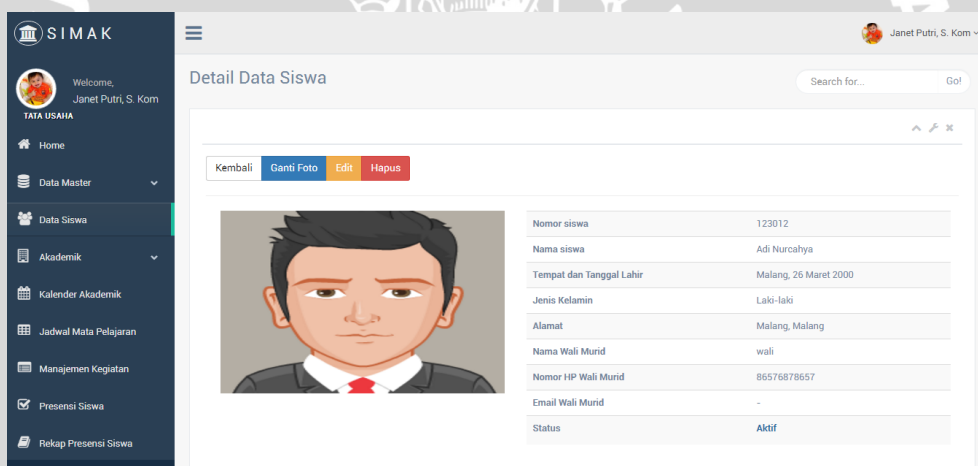
Halaman untuk menghapus data siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.13.



Gambar 6.13 Halaman menghapus data siswa hasil implemetasi

d. Melihat data siswa

Halaman untuk melihat data siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.14.



Gambar 6.14 Halaman melihat data siswa hasil implemetasi

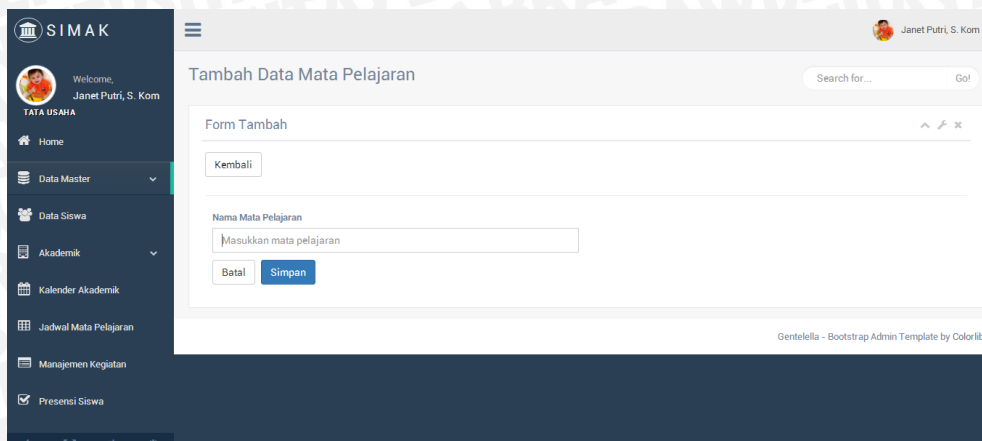
6.2.2.5 Halaman Manajemen Data Mata Pelajaran

Fitur manajemen data mata pelajaran merupakan fitur yang hanya dapat diakses oleh user tata usaha. Fitur manajemen data mata pelajaran terdapat 3 fungsional, yakni:

a. Menambah mata pelajaran

Halaman untuk menambah data mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.15.

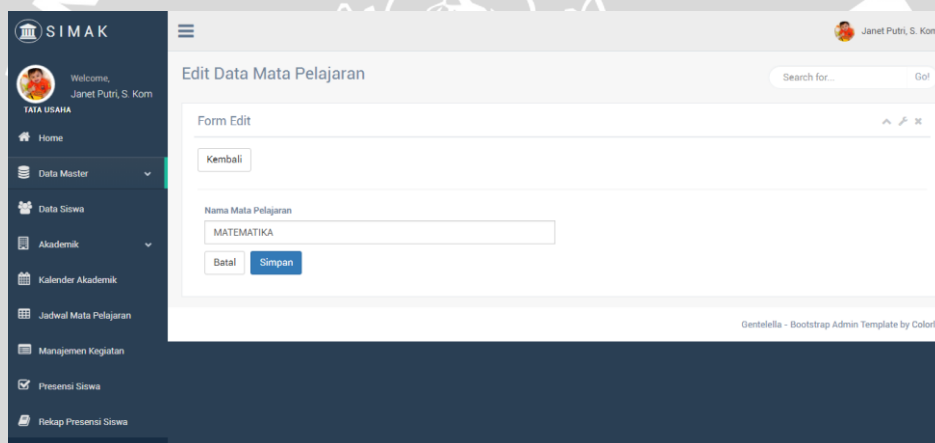




**Gambar 6.15** Halaman menambahkan data mata pelajaran hasil implemetasi

b. Mengubah data mata pelajaran

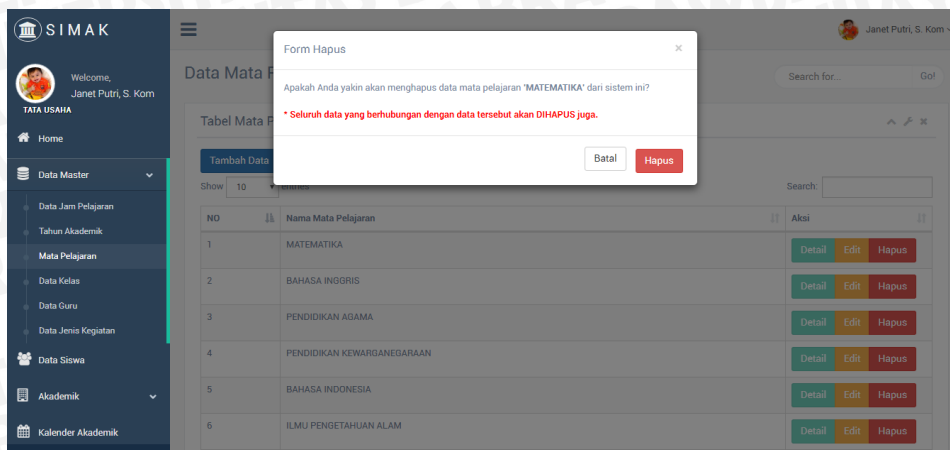
Halaman untuk mengubah data mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.16.



**Gambar 6.16** Halaman mengubah data mata pelajaran hasil implemetasi

c. Menghapus data mata pelajaran

Halaman untuk menghapus data mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.17.



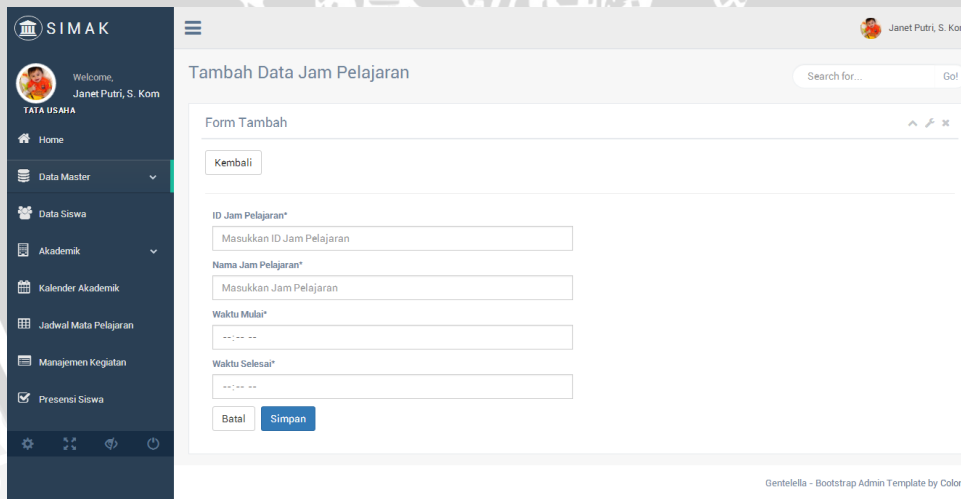
**Gambar 6.17** Halaman menghapus data mata pelajaran hasil implemetasi

### 6.2.2.6 Halaman Manajemen Jam Pelajaran

Fitur manajemen data jam pelajaran merupakan fitur yang hanya dapat diakses oleh *user* tata usaha. Fitur manajemen jam pelajaran terdapat 3 fungsional, yakni:

a. Menambah data jam pelajaran

Halaman untuk menambah data kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.18.

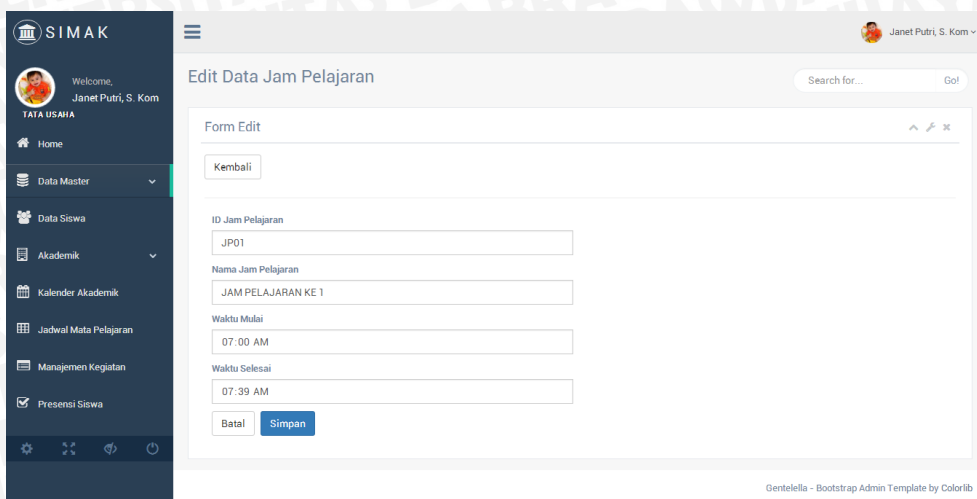


**Gambar 6.18** Halaman menambah data jam pelajaran hasil implemetasi

b. Mengubah data jam pelajaran

Halaman untuk mengubah data jam pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.19.

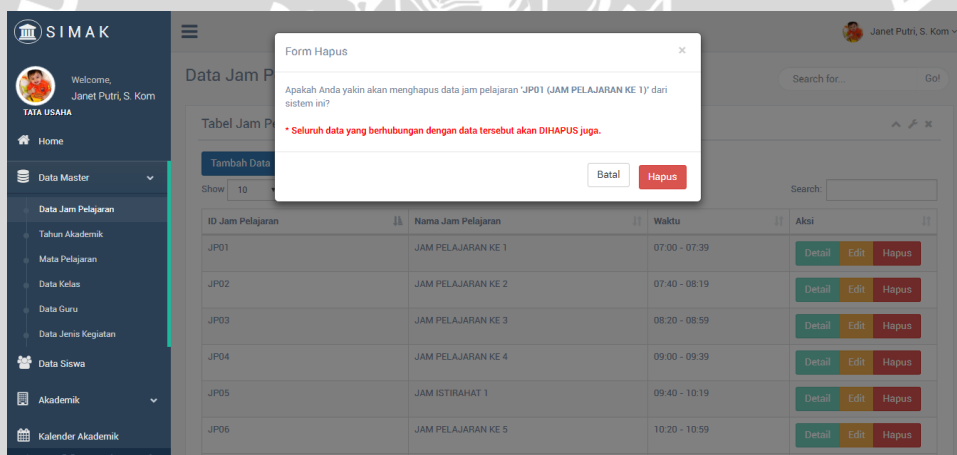




**Gambar 6.19** Halaman mengubah data jam pelajaran hasil implemetasi

c. Menghapus data jam pelajaran

Halaman untuk menghapus data jam pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.20.



**Gambar 6.20** Halaman menghapus data jam pelajaran hasil implemetasi

**6.2.2.7 Halaman Manajemen Data Kelas**

Fitur manajemen data kelas merupakan fitur yang hanya dapat diakses oleh user tata usaha. Fitur manajemen data kelas terdapat 4 fungsional, yakni:

a. Menambah data kelas

Halaman untuk menambah data kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.21.



**Gambar 6.21** Halaman menambah data kelas hasil implemetasi

b. Mengubah data kelas

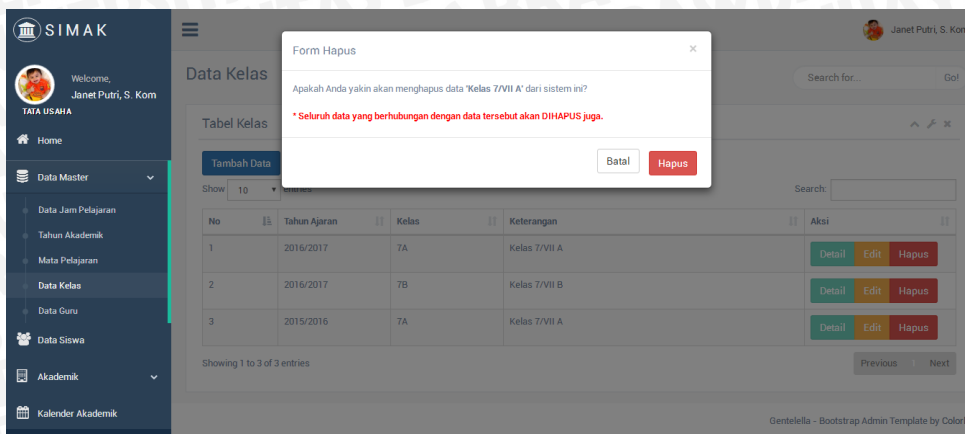
Halaman untuk mengubah data jam pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.22.

**Gambar 6.22** Halaman mengubah data kelas hasil implemetasi

c. Menghapus data kelas

Halaman untuk menghapus data kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.23.

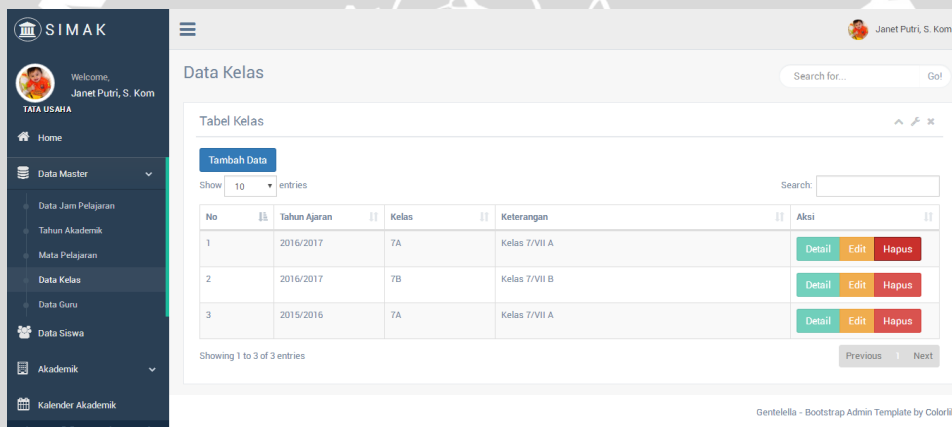




**Gambar 6.23** Halaman menghapus data kelas hasil implemetasi

d. Melihat data kelas

Halaman untuk melihat data guru dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.24.



**Gambar 6.24** Halaman melihat data kelas hasil implemetasi

### 6.2.2.8 Halaman Manajemen Data Pengajar

Fitur manajemen data pengajar merupakan fitur yang hanya dapat diakses oleh *user* tata usaha. Fitur manajemen data pengajar terdapat 3 fungsional, yakni:

a. Menambah data pengajar

Halaman untuk menambah data pengajar dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.25.

**Tambah Data pengajar**

Form Tambah

Kembali

Nama Guru\*  
Pilih Guru

Tahun Ajaran\*  
2016/2017 (Default)

Tingkat\*  
VII

Subtingkat\*  
A

Mata Pelajaran\*  
Pilih Mata Pelajaran

Jumlah Jam

Batal Simpan

Gentelella - Bootstrap Admin Template by Colorlib

**Gambar 6.25** Halaman menambah data pengajar hasil implemetasi

b. Mengubah data pengajar

Halaman untuk mengubah data pengajar dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.26.

**Edit Data Pengajar**

Form Edit

Kembali

Nama Guru\*  
Kim Tae Hee, ST., MT.

Tahun Ajaran\*  
2016/2017 (Default)

Tingkat\*  
VII

Subtingkat\*  
A

Mata Pelajaran\*  
BAHASA INGGRIS

Jumlah Jam  
4

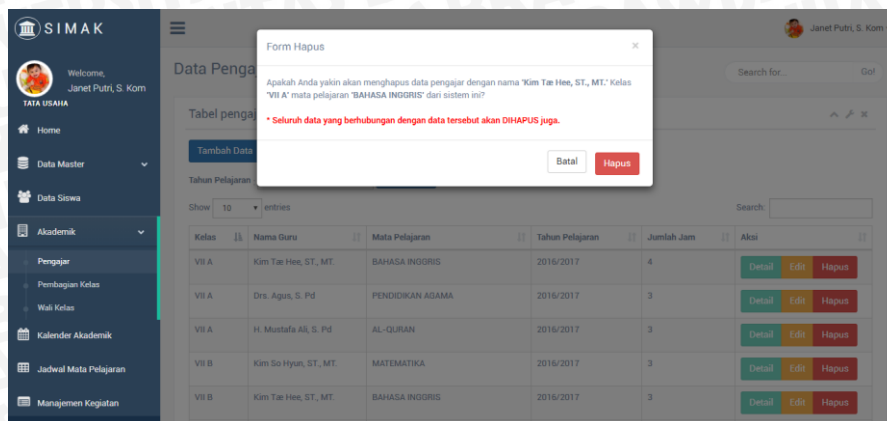
Batal Simpan

Gentelella - Bootstrap Admin Template by Colorlib

**Gambar 6.26** Halaman mengubah data pengajar hasil implemetasi

c. Menghapus data pengajar

Halaman untuk menghapus data pengajar dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.27.



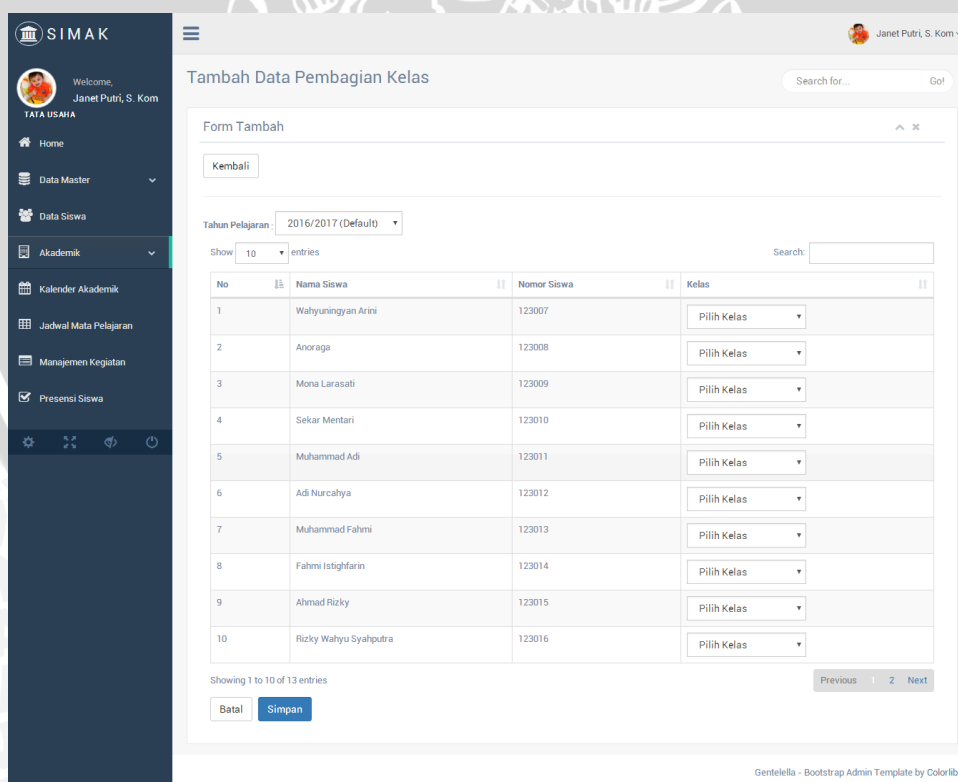
Gambar 6.27 Halaman menghapus data pengajar hasil implemetasi

### 6.2.2.9 Halaman Manajemen Pembagian Kelas

Fitur manajemen pembagian kelas merupakan fitur yang hanya dapat diakses oleh user tata usaha. Fitur manajemen pembagian kelas terdapat 3 fungsional, yakni:

a. Menambah pembagian kelas

Halaman untuk menambah pembagian kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.28.

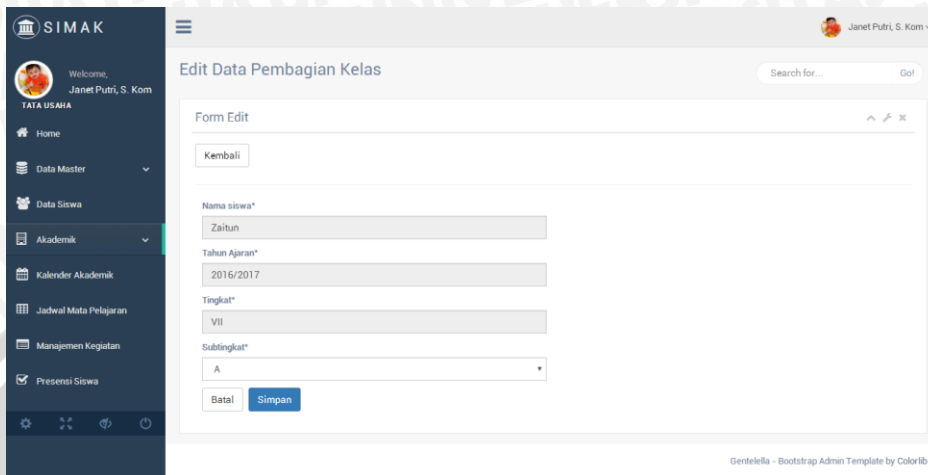


Gambar 6.28 Halaman menambah pembagian kelas hasil implemetasi



b. Mengubah pembagian kelas

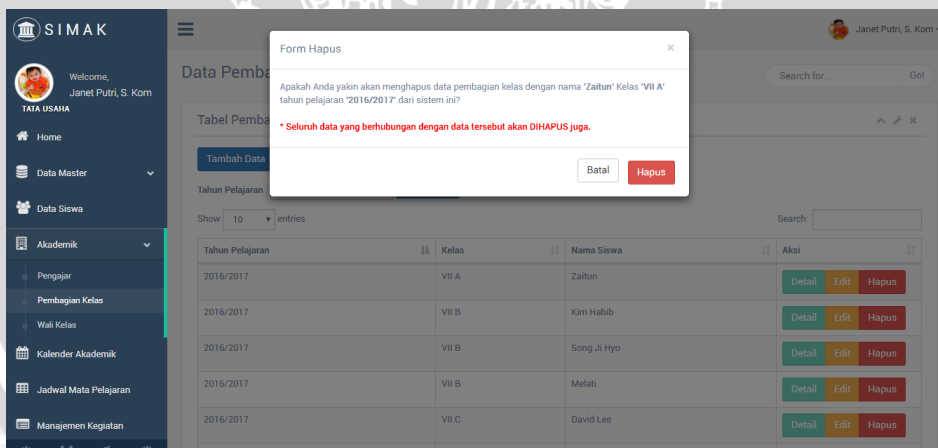
Halaman untuk mengubah pembagian kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.29.



**Gambar 6.29** Halaman mengubah pembagian kelas hasil implemetasi

c. Menghapus pembagian kelas

Halaman untuk menghapus pembagian kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.30.



**Gambar 6.30** Halaman menghapus pembagian kelas hasil implemetasi

**6.2.2.10 Halaman Manajemen Kalender Akademik**

Fitur manajemen kalender akademik merupakan fitur yang dapat diakses oleh semua *user*. Namun, *user* guru dan kepala sekolah hanya bisa mengakses fungsional melihat kalender akademik dari 4 fungsional yang terdapat pada fitur manajemen kalender akademik. Adapun, 4 fungsional yang terdapat pada fitur manajemen kalender akademik, yakni:



a. Menambah kalender akademik

Halaman untuk menambah kalender akademik dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.31.

**Gambar 6.31** Antarmuka menambah kalender akademik hasil implemetasi

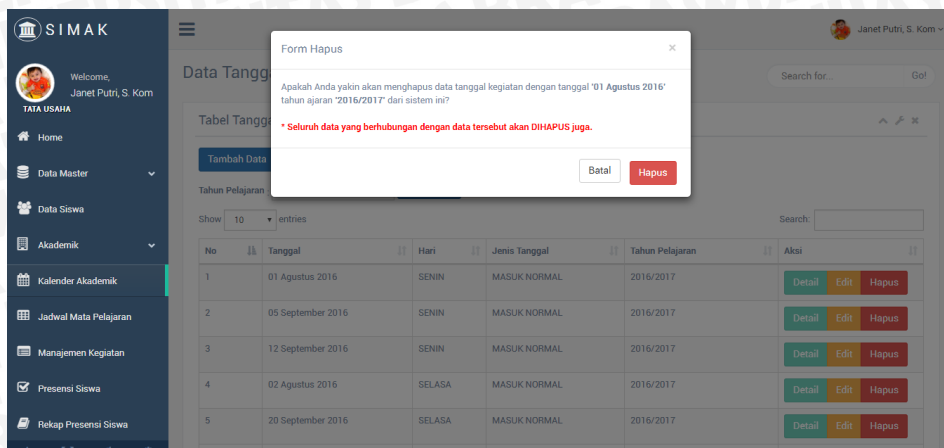
b. Mengubah kalender akademik

Halaman untuk mengubah kalender akademik dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.32.

**Gambar 6.32** Halaman mengubah kalender akademik hasil implemetasi

c. Menghapus kalender akademik

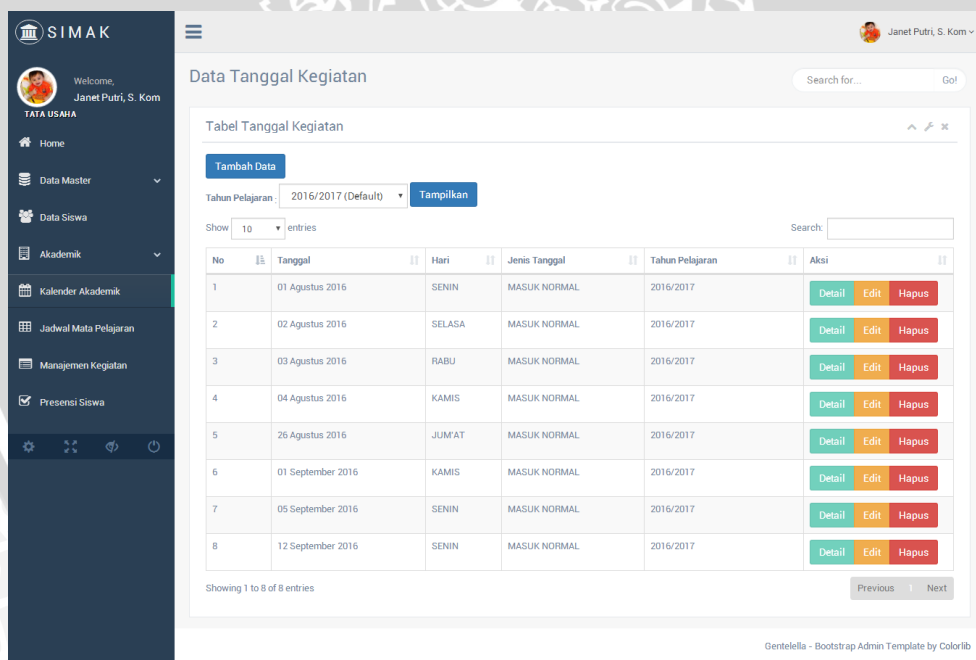
Halaman untuk menghapus kalender akademik dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.33.



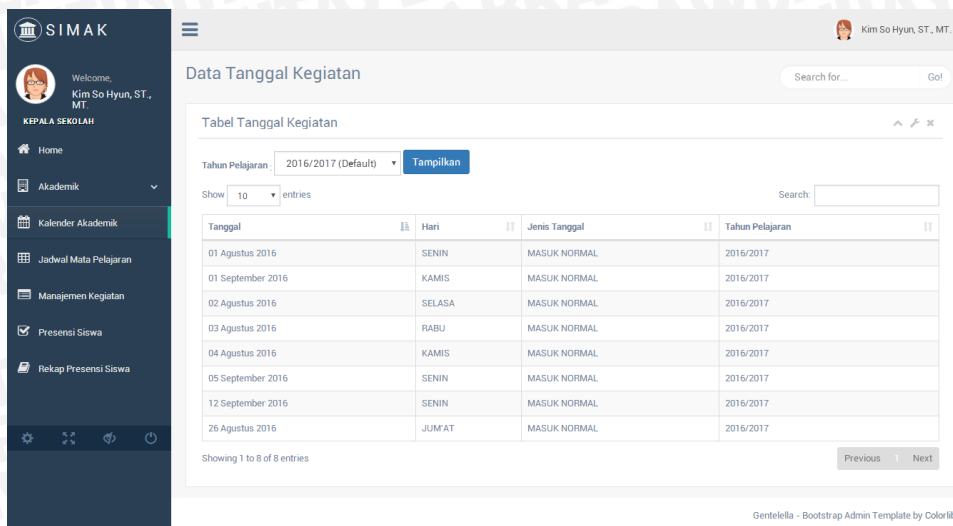
**Gambar 6.33** Halaman menghapus kalender akademik hasil implemetasi

d. Melihat kalender akademik

Halaman untuk melihat kalender akademik dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.34 untuk *user* tata usaha dan Gambar 6.35 untuk *user* guru dan kepala sekolah. Halaman melihat kalender akademik untuk *user* guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



**Gambar 6.34** Halaman melihat kalender akademik hasil implemetasi *user* tata usaha



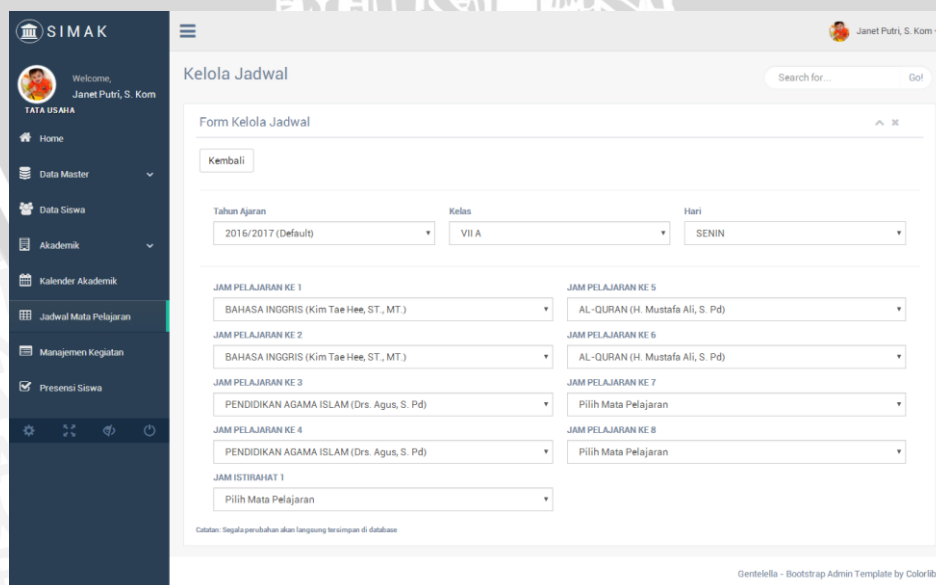
**Gambar 6.35** Halaman melihat kalender akademik hasil implemetasi *user* guru dan kepala sekolah

### 6.2.2.11 Halaman Manajemen Jadwal Mata Pelajaran

Fitur manajemen jadwal mata pelajaran merupakan fitur yang dapat diakses oleh semua *user*. Namun, *user* guru dan kepala sekolah hanya bisa mengakses fungsional melihat jadwal mata pelajaran dari 4 fungsional yang terdapat pada fitur manajemen jadwal mata pelajaran. Adapun, 4 fungsional yang terdapat pada fitur manajemen jadwal mata pelajaran, yakni:

a. Menambah jadwal mata pelajaran

Halaman untuk menambah jadwal mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.36.

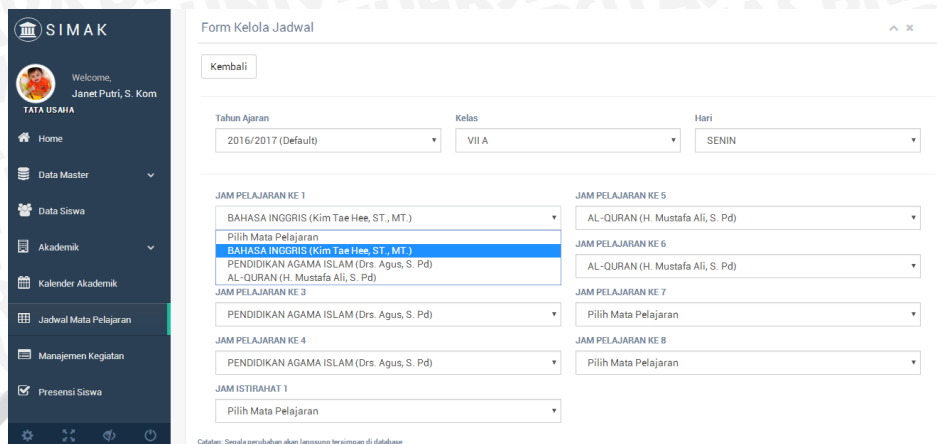


**Gambar 6.36** Halaman menambah jadwal mata pelajaran hasil implemetasi

b. Mengubah jadwal mata pelajaran



Halaman untuk mengubah jadwal mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.37.



**Gambar 6.37** Halaman mengubah jadwal mata pelajaran hasil implemetasi

Untuk mengubah jadwal mata pelajaran, *user* tata usaha hanya perlu mengganti mata pelajaran terpilih dengan mata pelajaran lain dengan cara memilih mata pelajaran yang ada pada daftar maka data mata pelajaran terpilih akan langsung tersimpan di *database* sistem.

### c. Menghapus jadwal mata pelajaran

Halaman untuk menghapus jadwal mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.38.



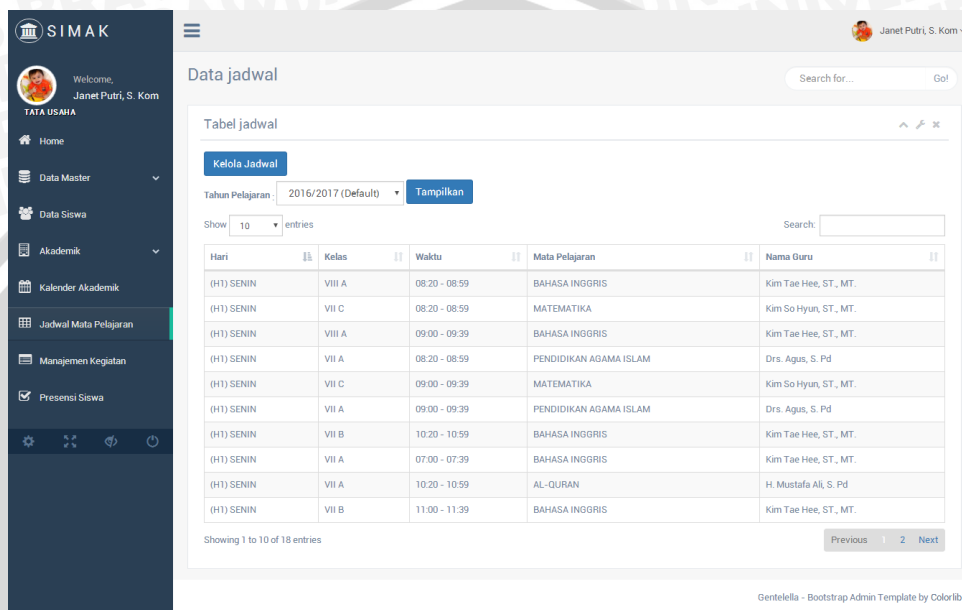
**Gambar 6.38** Halaman menghapus jadwal mata pelajaran hasil implemetasi

Untuk menghapus jadwal mata pelajaran, *user* tata usaha hanya perlu mengganti mata pelajaran terpilih dengan pilihan 'Pilih Mata Pelajaran' pada daftar maka data mata pelajaran terpilih akan langsung dihapus dari *database* sistem.

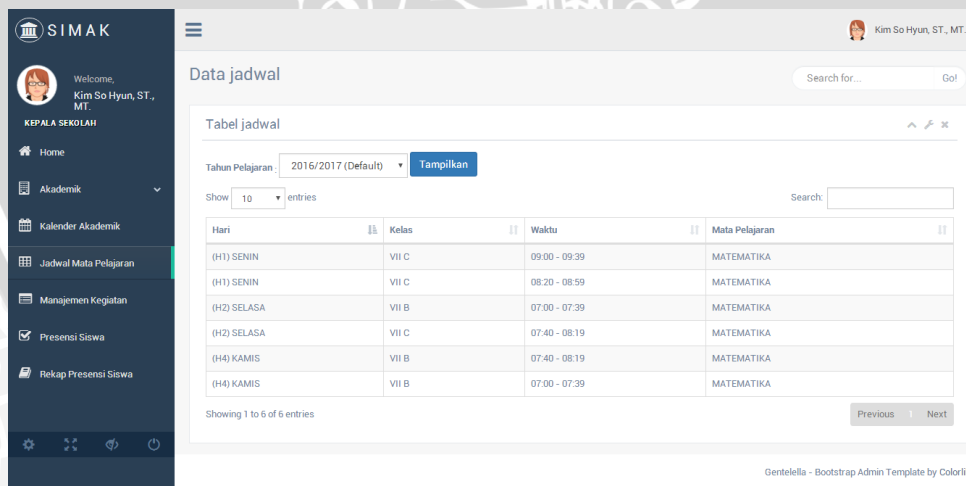


d. Melihat jadwal mata pelajaran

Halaman untuk melihat jadwal mata pelajaran dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.39 untuk *user* tata usaha dan Gambar 6.40 untuk *user* guru dan kepala sekolah. Halaman melihat jadwal mata pelajaran untuk *user* guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



Gambar 6.39 Halaman melihat jadwal mata pelajaran hasil implemetasi *user* tata usaha



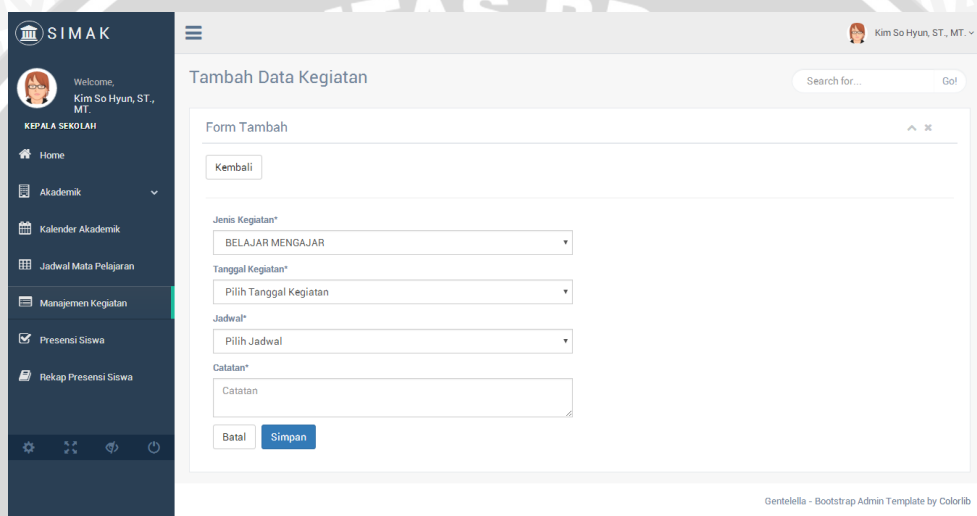
Gambar 6.40 Halaman melihat jadwal mata pelajaran hasil implemetasi *user* guru dan kepala sekolah

### 6.2.2.12 Halaman Manajemen Kegiatan

Fitur manajemen kegiatan merupakan fitur yang dapat diakses oleh semua *user*. Namun, tidak semua *user* memiliki hak akses terhadap fungsional yang sama. Fitur manajemen kegiatan terdapat 4 fungsional, yakni:

#### a. Menambah kegiatan

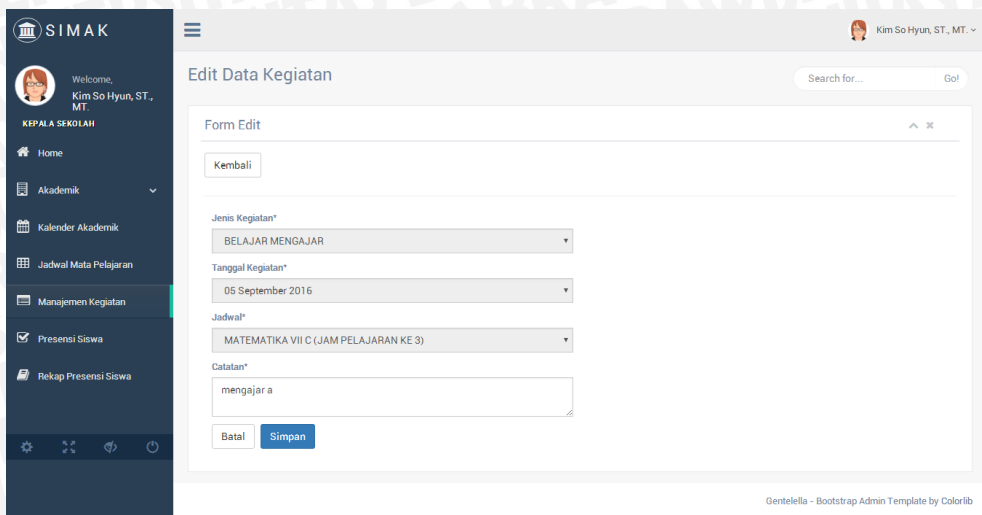
Halaman untuk menambah kegiatan hanya dapat diakses oleh *user* guru dan kepala sekolah. Halaman untuk menambah kegiatan dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.41. Halaman menambah kegiatan untuk *user* guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



**Gambar 6.41** Halaman menambah kegiatan hasil implemetasi

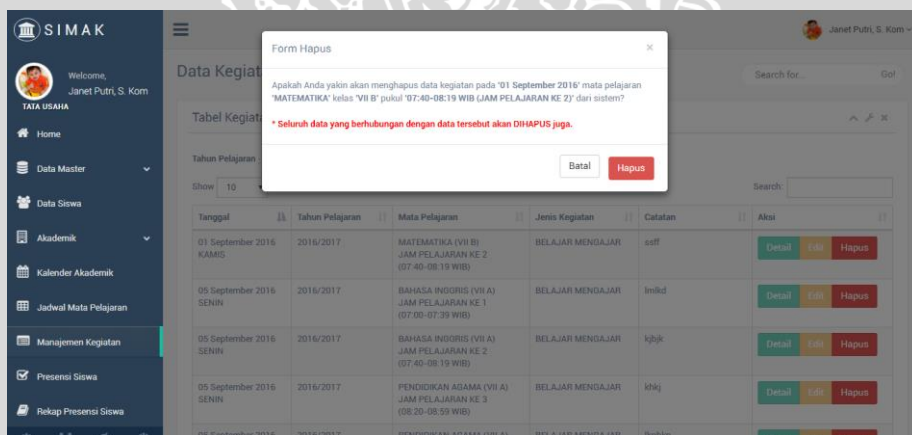
#### b. Mengubah kegiatan

Halaman untuk mengubah kegiatan hanya dapat diakses oleh *user* guru dan kepala sekolah. Halaman untuk mengubah kegiatan dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.42. Halaman mengubah kegiatan untuk *user* guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



**Gambar 6.42** Halaman mengubah kegiatan hasil implemetasi  
c. Menghapus kegiatan

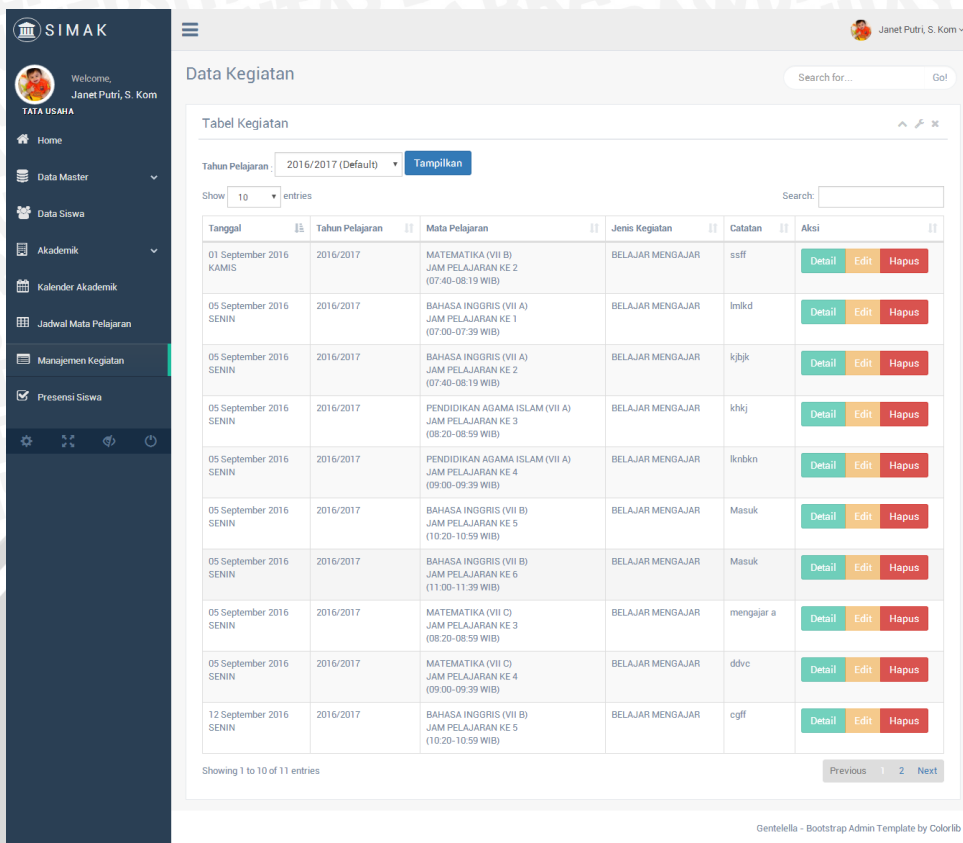
Halaman untuk menghapus kegiatan hanya dapat diakses oleh *user* tata usaha. Halaman untuk menghapus kegiatan dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.43.



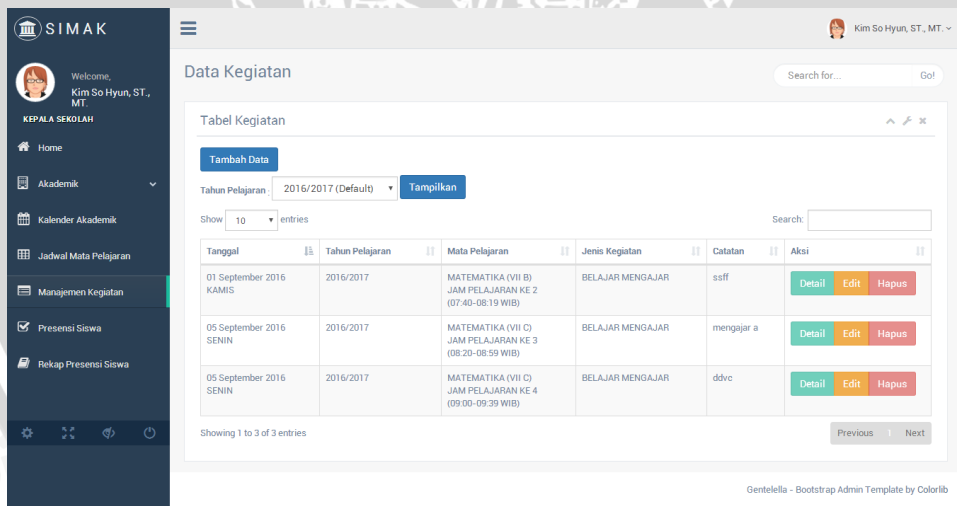
**Gambar 6.43** Halaman menghapus kegiatan hasil implemetasi  
d. Melihat kegiatan

Halaman untuk melihat kegiatan dapat diakses oleh semua *user*. Halaman untuk melihat kegiatan dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.44 untuk *user* tata usaha dan Gambar 6.45 untuk *user* guru dan kepala sekolah. Halaman melihat kegiatan untuk *user* guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.





Gambar 6.44 Halaman melihat kegiatan hasil implemetasi *user* tata usaha



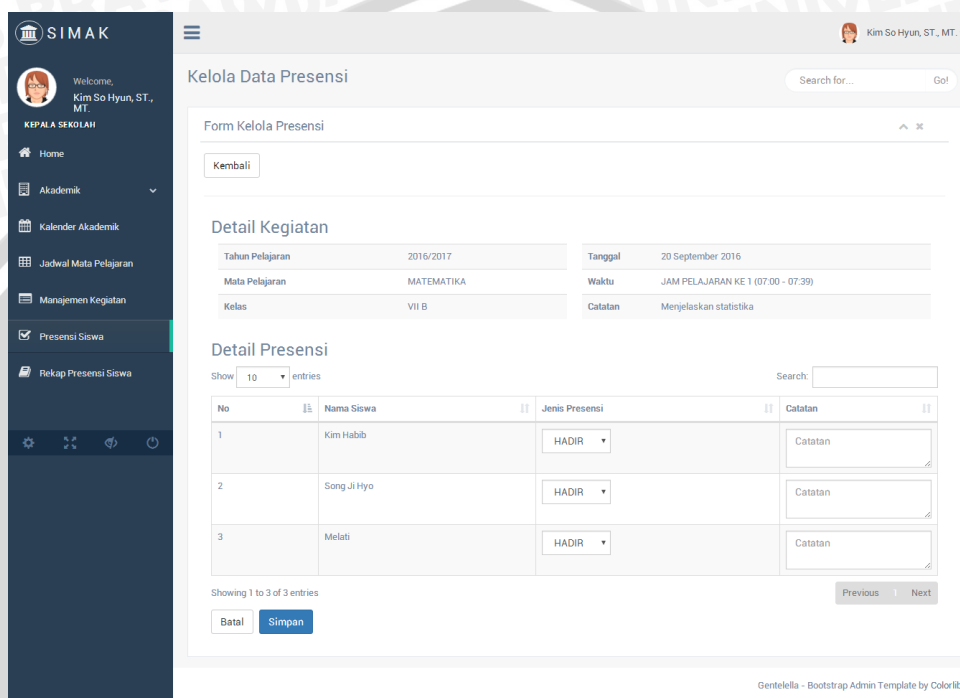
Gambar 6.45 Halaman melihat kegiatan hasil implemetasi *user* guru dan kepala sekolah

### 6.2.2.13 Halaman Manajemen Presensi Siswa

Fitur manajemen presensi siswa merupakan fitur yang dapat diakses oleh semua *user*. Namun, tidak semua *user* memiliki hak akses terhadap fungsional yang sama. Fitur manajemen presensi siswa terdapat 4 fungsional, yakni:

a. Menambah presensi siswa

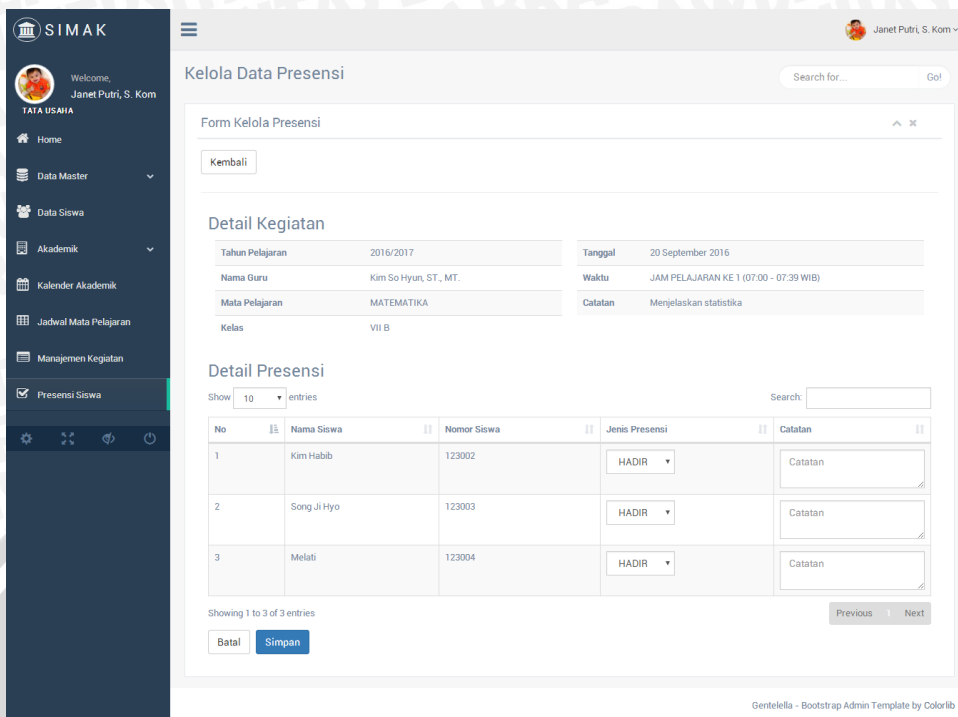
Halaman untuk menambah presensi siswa hanya dapat diakses oleh *user* guru dan kepala sekolah. Halaman untuk menambah presensi siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.46. Halaman menambah presensi siswa untuk *user* guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



**Gambar 6.46** Halaman menambah presensi siswa hasil implemetasi

b. Mengubah presensi siswa

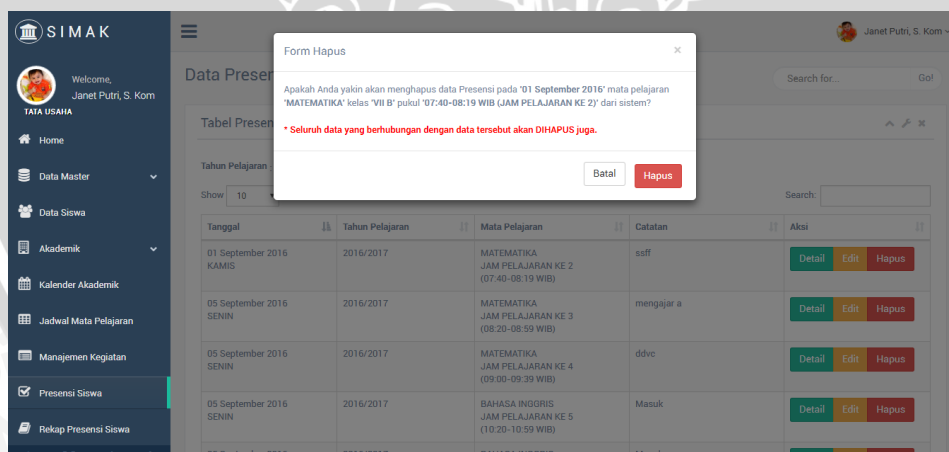
Halaman untuk mengubah presensi siswa hanya dapat diakses oleh *user* tata usaha. Halaman untuk mengubah presensi siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.47.



**Gambar 6.47** Halaman mengubah presensi siswa hasil implemetasi

c. Menghapus presensi siswa

Halaman untuk menghapus presensi siswa hanya dapat diakses oleh *user* tata usaha. Halaman untuk menghapus presensi siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.48.

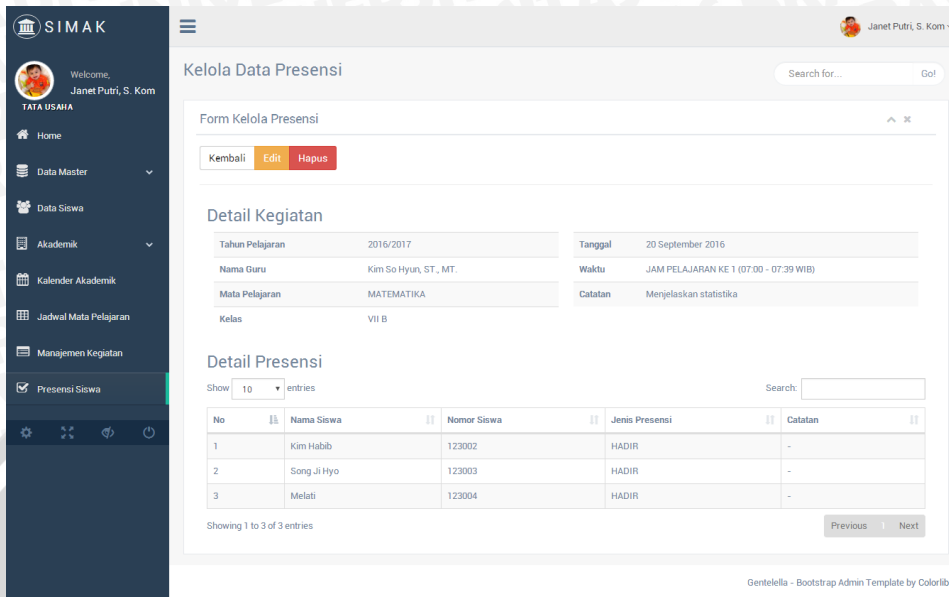


**Gambar 6.48** Halaman menghapus presensi siswa hasil implemetasi

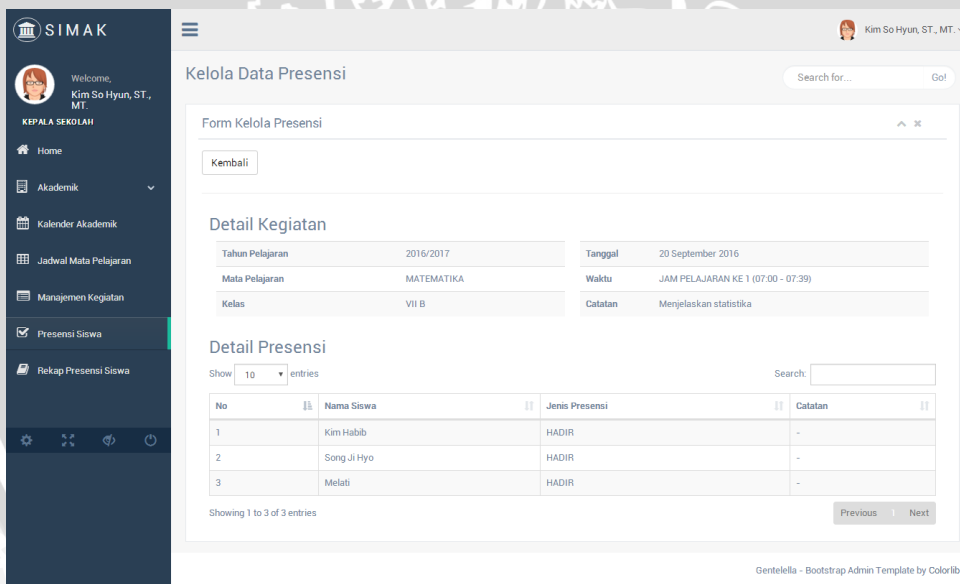
d. Melihat presensi siswa

Halaman untuk melihat presensi siswa dapat diakses oleh semua *user*. Halaman untuk melihat presensi siswa dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.49 untuk *user* tata usaha dan Gambar 6.50 untuk *user* guru dan kepala sekolah. Halaman melihat presensi siswa untuk

user guru dan kepala sekolah adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



**Gambar 6.49** Halaman melihat presensi siswa hasil implemetasi *user* tata usaha



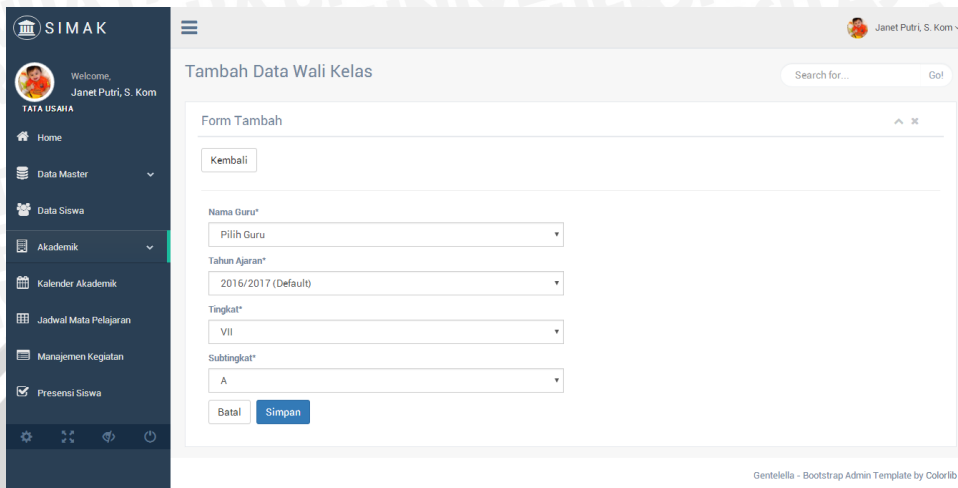
**Gambar 6.50** Halaman melihat presensi siswa hasil implemetasi *user* guru dan kepala sekolah

#### 6.2.2.14 Halaman Manajemen Wali Kelas

Fitur manajemen wali kelas merupakan fitur yang hanya dapat diakses oleh *user* tata usaha. Fitur manajemen wali kelas terdapat 4 fungsional, yakni:

a. Menambah wali kelas

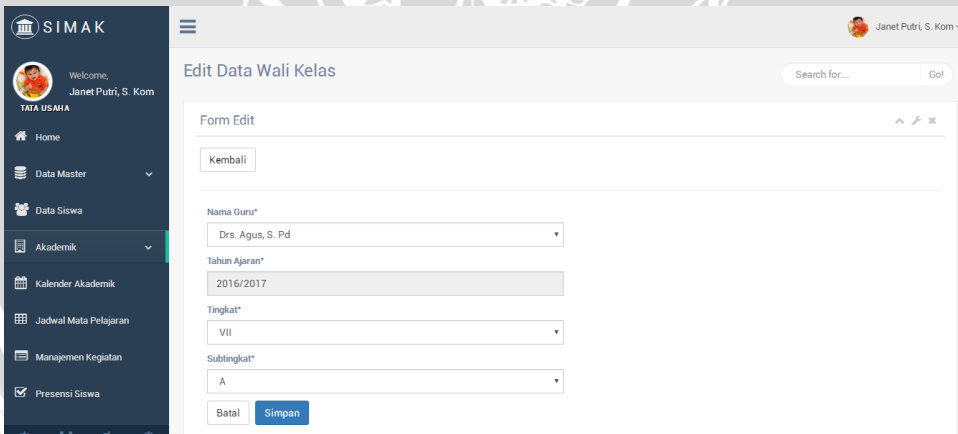
Halaman untuk menambah wali kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.51.



**Gambar 6.51** Halaman menambah wali kelas hasil implemetasi

b. Mengubah wali kelas

Halaman untuk mengubah wali kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.52.

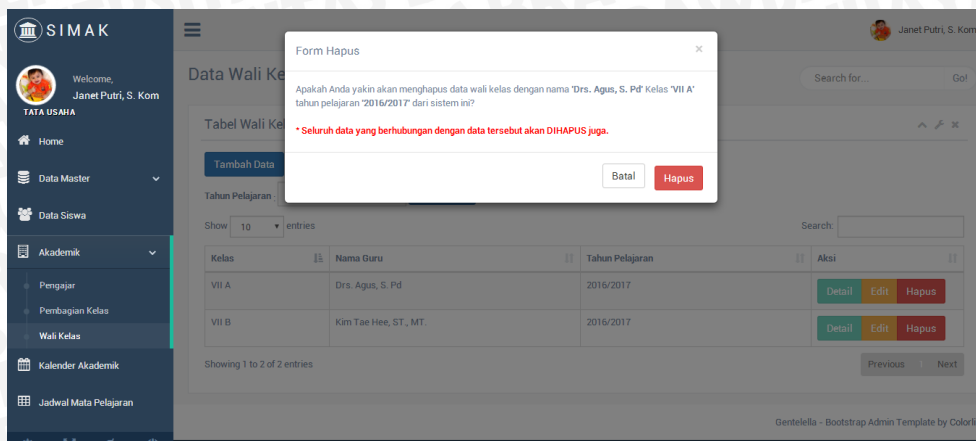


**Gambar 6.52** Halaman mengubah wali kelas hasil implemetasi

c. Menghapus wali kelas

Halaman untuk menghapus wali kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.53.

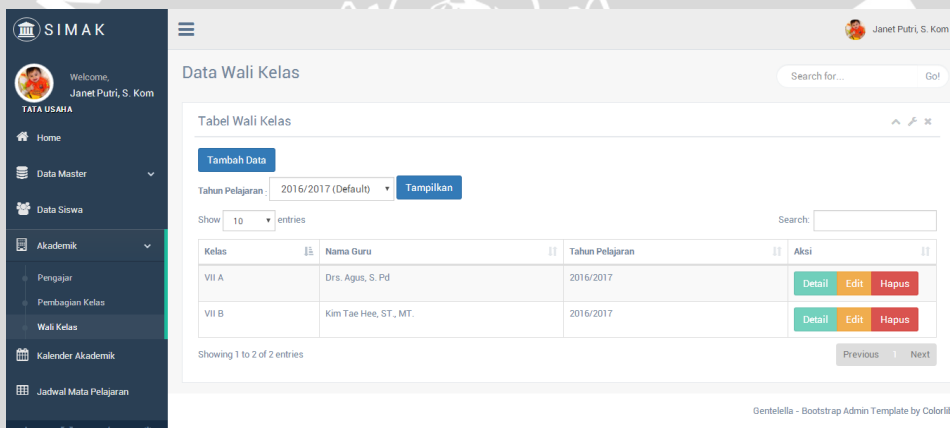




**Gambar 6.53** Halaman menghapus wali kelas hasil implemetasi

d. Melihat wali kelas

Halaman untuk melihat wali kelas dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.54.



**Gambar 6.54** Halaman melihat wali kelas hasil implemetasi

**6.2.2.15 Halaman Manajemen Rekanan**

Fitur manajemen rekanan merupakan fitur yang dapat diakses oleh semua user. Fitur manajemen rekanan hanya memiliki 1 fungsional, yakni:

a. Membuat rekanan

Halaman untuk membuat rekanan dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.55. Halaman membuat rekanan untuk semua user adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.



Detail Rekap Presensi Siswa

Tabel Rekap Presensi Siswa

Kembali

Detail Periode

|                 |                       |
|-----------------|-----------------------|
| Tahun Pelajaran | 2016/2017             |
| Nama Guru       | Kim Tae Hee, ST., MT. |
| Nomor Guru      | 12132                 |
| Kelas           | VII B                 |

Detail Presensi

Show 10 entries Search:

| No | Nama Siswa  | Hadir (Jam) | Tidak Hadir (Jam) | Tatap Muka (Jam) | Prosentase (%) | Aksi                |
|----|-------------|-------------|-------------------|------------------|----------------|---------------------|
| 1  | Kim Habib   | 2           | 2                 | 4                | 50             | Lihat Catatan Cetak |
| 2  | Song Ji-Hyo | 2           | 2                 | 4                | 50             | Lihat Catatan Cetak |
| 3  | Melati      | 3           | 0                 | 3                | 100            | Lihat Catatan Cetak |

Showing 1 to 3 of 3 entries Previous Next

Gentelella - Bootstrap Admin Template by Colorlib

**Gambar 6.55** Halaman membuat rekap hasil implemetasi

#### 6.2.2.16 Halaman Mencetak Laporan

Fitur mencetak laporan merupakan fitur yang dapat diakses oleh semua *user*. Fitur mencetak laporan hanya memiliki 1 fungsional, yakni:

##### a. Mencetak rekap

Halaman untuk mencetak rekap dibuat berdasarkan rancangan antarmuka yang telah dibuat (Fatha, 2017). Halaman hasil implementasi dapat dilihat pada Gambar 6.56. Halaman mencetak rekap untuk semua *user* adalah sama persis. Oleh karena itu, contoh hasil implementasi dipilih salah satu.

Print  
Total: 1 page

---

Destination

Pages  All  
 e.g. 1-5, 8, 11-13

Layout

Paper size

Margins

Options  Simplify page  
 Headers and footers  
 Background graphics

**Madrasah Muhammadiyah 1 Tlogomas Malang**  
Laporan Rekap Presensi

**A. Detail Siswa**

|                 |                       |
|-----------------|-----------------------|
| Tahun Pelajaran | 2016/2017             |
| Nama Wali Kelas | Kim Tan Hui, ST., MT. |
| Nama Siswa      | Kim Hobb              |
| Nomor Siswa     | 123002                |
| Nama Wali Murid | uik                   |
| Kelas           | VI B                  |

**B. Perhitungan Presensi**

| Hadir (Jam) | Absen (Jam) | Tatap Muka (Jam) | Presensiase (%) |
|-------------|-------------|------------------|-----------------|
| 2           | 2           | 4                | 50              |

**C. Detail Catatan**

| No | Tanggal Kegiatan  | Mata Pelajaran | Pengajar              | Jam Pelajaran                    | Catatan        |
|----|-------------------|----------------|-----------------------|----------------------------------|----------------|
| 1  | 01 September 2016 | MATEMATIKA     | Kim Tan Hui, ST., MT. | JAM PELAJARAN/KE 2 0740-0810 WIB | Dobe           |
| 2  | 02 September 2016 | Bahasa Inggris | Kim Tan Hui, ST., MT. | JAM PELAJARAN/KE 5 0820-1020 WIB | Titik Di Manja |

Malang, 28 September 2016

Gambar 6.56 Halaman mencetak rekap hasil implemetasi



## BAB 7 PENGUJIAN

### 7.1 Rencana Pengujian

Pengujian sistem informasi dilakukan pada kebutuhan fungsional dan non-fungsional. Pada Tabel 7.1 berisi tentang kebutuhan fungsional yang akan diuji.

**Tabel 7.1** Kebutuhan atau persyaratan fungsional yang akan diuji (Fatha, 2017)

| No | Kode SRS       | Butir Uji                                   | Jenis Pengujian  |
|----|----------------|---|------------------|
| 1  | PF-SIMAK-001.1 | Pengujian <i>login</i>                      | <i>Black box</i> |
| 2  | PF-SIMAK-002.1 | Pengujian menambah data guru                | <i>Black box</i> |
| 3  | PF-SIMAK-002.2 | Pengujian mengubah data guru.               | <i>Black box</i> |
| 4  | PF-SIMAK-002.3 | Pengujian menghapus data guru.              | <i>Black box</i> |
| 5  | PF-SIMAK-002.4 | Pengujian melihat detail data guru          | <i>Black box</i> |
| 6  | PF-SIMAK-003.1 | Pengujian menambah user                     | <i>Black box</i> |
| 7  | PF-SIMAK-003.2 | Pengujian mengubah user                     | <i>Black box</i> |
| 8  | PF-SIMAK-003.3 | Pengujian menghapus user                    | <i>Black box</i> |
| 9  | PF-SIMAK-003.4 | Pengujian melihat semua data user           | <i>Black box</i> |
| 10 | PF-SIMAK-004.1 | Pengujian menambah data siswa               | <i>Black box</i> |
| 11 | PF-SIMAK-004.2 | Pengujian mengubah data siswa               | <i>Black box</i> |
| 12 | PF-SIMAK-004.3 | Pengujian menghapus data siswa              | <i>Black box</i> |
| 13 | PF-SIMAK-004.4 | Pengujian melihat semua data siswa          | <i>Black box</i> |
| 14 | PF-SIMAK-005.1 | Pengujian menambah data mata pelajaran      | <i>Black box</i> |
| 15 | PF-SIMAK-005.2 | Pengujian mengubah data mata pelajaran      | <i>Black box</i> |
| 16 | PF-SIMAK-005.3 | Pengujian menghapus data mata pelajaran     | <i>Black box</i> |
| 17 | PF-SIMAK-005.4 | Pengujian melihat data mata pelajaran       | <i>Black box</i> |
| 18 | PF-SIMAK-006.1 | Pengujian menambah jam pelajaran            | <i>Black box</i> |
| 19 | PF-SIMAK-006.2 | Pengujian mengubah detail jam pelajaran     | <i>Black box</i> |
| 20 | PF-SIMAK-006.3 | Pengujian menghapus data jam pelajaran      | <i>Black box</i> |
| 21 | PF-SIMAK-006.4 | Pengujian melihat semua data mata pelajaran | <i>Black box</i> |
| 22 | PF-SIMAK-007.1 | Pengujian menambah data kelas               | <i>Black box</i> |
| 23 | PF-SIMAK-007.2 | Pengujian mengubah data kelas               | <i>Black box</i> |
| 24 | PF-SIMAK-007.3 | Pengujian menghapus data kelas              | <i>Black box</i> |
| 25 | PF-SIMAK-007.4 | Pengujian melihat data kelas                | <i>Black box</i> |
| 26 | PF-SIMAK-008.1 | Pengujian menambah data pengajar            | <i>Black box</i> |
| 27 | PF-SIMAK-008.2 | Pengujian mengubah data pengajar            | <i>Black box</i> |
| 28 | PF-SIMAK-008.3 | Pengujian menghapus data pengajar           | <i>Black box</i> |
| 29 | PF-SIMAK-008.4 | Pengujian menampilkan data pengajar         | <i>Black box</i> |

**Tabel 7.1** Kebutuhan atau persyaratan fungsional yang akan diuji (Fatha, 2017)(lanjutan)

| No | Kode SRS       | Butir Uji   | Jenis Pengujian  |
|----|----------------|---|------------------|
| 30 | PF-SIMAK-009.1 | Pengujian untuk menambah data siswa baru pada kelas-kelas yang telah dibuat | <i>Black box</i> |
| 31 | PF-SIMAK-009.2 | Pengujian mengubah data siswa pada kelas yang telah diinputkan.             | <i>Black box</i> |
| 32 | PF-SIMAK-009.3 | Pengujian menghapus data siswa dari kelas yang ada.                         | <i>Black box</i> |
| 33 | PF-SIMAK-009.4 | Pengujian melihat semua data siswa dalam satu kelas.                        | <i>Black box</i> |
| 34 | PF-SIMAK-010.1 | Pengujian menambah kalender akademik  | <i>Black box</i> |
| 35 | PF-SIMAK-010.2 | Pengujian mengubah kalender akademik  | <i>Black box</i> |
| 36 | PF-SIMAK-010.3 | Pengujian menghapus kalender akademik                                       | <i>Black box</i> |
| 37 | PF-SIMAK-010.4 | Pengujian melihat kalender akademik   | <i>Black box</i> |
| 38 | PF-SIMAK-011.1 | Pengujian menambah jadwal mata pelajaran                                    | <i>Black box</i> |
| 39 | PF-SIMAK-011.2 | Pengujian mengubah jadwal mata pelajaran                                    | <i>Black box</i> |
| 40 | PF-SIMAK-011.3 | Pengujian menghapus jadwal mata pelajaran                                   | <i>Black box</i> |
| 41 | PF-SIMAK-011.4 | Pengujian melihat jadwal mata pelajaran                                     | <i>Black box</i> |
| 42 | PF-SIMAK-012.1 | Pengujian menambah data kegiatan  | <i>Black box</i> |
| 43 | PF-SIMAK-012.2 | Pengujian mengubah data kegiatan  | <i>Black box</i> |
| 44 | PF-SIMAK-012.3 | Pengujian menghapus kegiatan  | <i>Black box</i> |
| 45 | PF-SIMAK-012.4 | Pengujian melihat data kegiatan   | <i>Black box</i> |
| 46 | PF-SIMAK-013.1 | Pengujian menambah data presensi siswa                                      | <i>Black box</i> |
| 47 | PF-SIMAK-013.2 | Pengujian mengubah data presensi  | <i>Black box</i> |
| 48 | PF-SIMAK-013.3 | Pengujian menghapus data presensi siswa                                     | <i>Black box</i> |
| 49 | PF-SIMAK-013.4 | Pengujian melihat data presensi siswa                                       | <i>Black box</i> |
| 50 | PF-SIMAK-014.1 | Pengujian menambah data wali kelas  | <i>Black box</i> |
| 51 | PF-SIMAK-014.2 | Pengujian mengubah data wali kelas  | <i>Black box</i> |
| 52 | PF-SIMAK-014.3 | Pengujian menghapus data wali kelas   | <i>Black box</i> |
| 53 | PF-SIMAK-014.4 | Pengujian melihat data wali kelas   | <i>Black box</i> |
| 54 | PF-SIMAK-015.1 | Pengujian membuat data rekapan  | <i>Black box</i> |
| 55 | PF-SIMAK-016.1 | Pengujian mencetak rekapan  | <i>Black box</i> |

Kebutuhan non-fungsional yang diuji adalah *compability* pada *browser*. Pengujian dilakukan dengan menggunakan *SortSite 5 Trial*.

## 7.2 Pengujian *Black-Box*

Pengujian *black-box* adalah pengujian yang dilakukan dari segi fungsional tanpa menguji desain dan kode program. Berikut hasil pengujian *black-box* terhadap fitur-fitur SIMAK.

### 7.2.1 Pengujian Identifikasi Pengguna

Hasil pengujian *use case* identifikasi pengguna atau *login* dapat dilihat pada Tabel 7.2.

**Tabel 7.2** Hasil pengujian Identifikasi Pengguna

| Identifikasi Pengguna |                          |  |
|-----------------------|--------------------------|--|
| Test case 1           | Skenario 1               | Pengujian <i>login</i> dengan <i>username</i> dan <i>password</i> yang valid |
|                       | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Selamat. Anda berhasil login.</i> "       |
|                       | Keluaran                 | Sistem menampilkan notifikasi " <i>Selamat. Anda berhasil login.</i> "       |
| Test case 2           | Skenario 2               | Pengujian <i>login</i> dengan <i>password</i> yang tidak valid               |
|                       | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Maaf. Password tidak sesuai.</i> "        |
|                       | Keluaran                 | Sistem menampilkan notifikasi " <i>Maaf. Password tidak sesuai.</i> "        |
| Test case 3           | Skenario 3               | Pengujian <i>login</i> dengan <i>username</i> tidak valid                    |
|                       | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Maaf. Data tidak ditemukan.</i> "         |
|                       | Keluaran                 | Sistem menampilkan notifikasi " <i>Maaf. Data tidak ditemukan.</i> "         |
| Test case 4           | Skenario 4               | Pengujian <i>login</i> dengan <i>username</i> atau <i>password</i> kosong    |
|                       | Keluaran yang diharapkan | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong"              |
|                       | Keluaran                 | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong"              |

Dari 4 *test case* yang diuji pada Tabel 6.2, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi *login*.

## 7.2.2 Pengujian Manajemen Data Guru

Hasil pengujian *use case* manajemen data guru dapat dilihat pada Tabel 7.3.

**Tabel 7.3** Hasil pengujian manajemen data guru

| Manajemen Data Guru |                          |   |
|---------------------|--------------------------|---|
| Test case 5         | Skenario 1               | Pengujian menambah data guru dengan data valid dan nomor guru belum ada di database sistem  |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
|                     | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
| Test case 6         | Skenario 2               | Pengujian menambah data guru dengan data valid tapi nomor guru sudah ada di database sistem   |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Nomor guru telah digunakan</i> "   |
|                     | Keluaran                 | Sistem menampilkan notifikasi " <i>Nomor guru telah digunakan</i> "   |
| Test case 7         | Skenario 3               | Pengujian menambah data guru dengan mengosongkan form   |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong*" atau "Pilih <Nama kolom>**"<br>* Form berupa masukan<br>** Form berupa pilihan |
|                     | Keluaran                 | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong*" atau "Pilih <Nama kolom>**"<br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 8         | Skenario 4               | Pengujian mengubah data guru dengan data valid dan nomor guru tidak sama dengan nomor guru lain   |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
|                     | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |

**Tabel 7.3** Hasil pengujian manajemen data guru (lanjutan)

| Manajemen Data Guru |                          |   |
|---------------------|--------------------------|---|
| Test case 9         | Skenario 5               | Pengujian mengubah data guru dengan data valid tapi nomor guru sama dengan nomor guru lain  |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Nomor guru telah digunakan</i> "   |
|                     | Keluaran                 | Sistem menampilkan notifikasi " <i>Nomor guru telah digunakan</i> "   |
| Test case 10        | Skenario 6               | Pengujian mengubah data guru dengan mengosongkan form   |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
|                     | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 11        | Skenario 7               | Pengujian menghapus data guru   |
|                     | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
|                     | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
| Test case 12        | Skenario 8               | Pengujian melihat data guru   |
|                     | Keluaran yang diharapkan | Sistem menampilkan data guru dalam bentuk tabel   |
|                     | Keluaran                 | Sistem menampilkan data guru dalam bentuk tabel   |

Dari 8 test case yang diuji pada Tabel 7.3 didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada use case spesifikasi manajemen data guru.



### 7.2.3 Pengujian Manajemen User

Hasil pengujian *use case* manajemen *user* dapat dilihat pada Tabel 7.4.

**Tabel 7.4** Hasil pengujian manajemen *user*

| Manajemen User |                          |  |
|----------------|--------------------------|--|
| Test case 13   | Skenario 1               | Pengujian menambah data <i>user</i> dengan data valid dan data belum ada di database sistem  |
|                | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                           |
|                | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                           |
| Test case 14   | Skenario 2               | Pengujian menambah data <i>user</i> dengan data valid tapi data sudah ada di database sistem |
|                | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data guru tersebut telah digunakan</i> "                  |
|                | Keluaran                 | Sistem menampilkan notifikasi " <i>Data guru tersebut telah digunakan</i> "                  |
| Test case 15   | Skenario 3               | Pengujian menambah data <i>user</i> dengan mengosongkan form                                 |
|                | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "                            |
|                | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "                            |
| Test case 16   | Skenario 4               | Pengujian mengubah data <i>user</i> dengan data valid  |
|                | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                                |
|                | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                                |
| Test case 17   | Skenario 5               | Pengujian mengubah data <i>user</i> dengan mengosongkan form                                 |
|                | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "                            |
|                | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "                            |
| Test case 18   | Skenario 6               | Pengujian menghapus data <i>user</i>   |
|                | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                               |
|                | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                               |

**Tabel 7.4** Hasil pengujian manajemen *user* (lanjutan)

| Manajemen <i>User</i> |                          |  |
|-----------------------|--------------------------|--|
| Test case 19          | Skenario 6               | Pengujian melihat data <i>user</i>                     |
|                       | Keluaran yang diharapkan | Sistem menampilkan data <i>user</i> dalam bentuk tabel |
|                       | Keluaran                 | Sistem menampilkan data <i>user</i> dalam bentuk tabel |

Dari 7 *test case* yang diuji pada Tabel 7.4, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen *user*.

### 7.2.4 Pengujian Manajemen Data Siswa

Hasil pengujian *use case* manajemen data siswa dapat dilihat pada Tabel 7.5.

**Tabel 7.5** Hasil pengujian manajemen data siswa

| Manajemen Data Siswa |                          |   |
|----------------------|--------------------------|---|
| Test case 20         | Skenario 1               | Pengujian menambah data siswa dengan data valid dan nomor siswa belum ada di database sistem  |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
| Test case 21         | Skenario 2               | Pengujian menambah data siswa dengan data valid tapi nomor siswa sudah ada di database sistem   |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Nomor siswa telah digunakan</i> "  |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Nomor siswa telah digunakan</i> "  |
| Test case 22         | Skenario 3               | Pengujian menambah data siswa dengan mengosongkan form  |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong*" atau "Pilih <Nama kolom>**"<br><br>* Form berupa masukan<br>** Form berupa pilihan |

**Tabel 7.5** Hasil pengujian manajemen data siswa (lanjutan)

| Manajemen Data Siswa |                          |   |
|----------------------|--------------------------|---|
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 23         | Skenario 4               | Pengujian mengubah data siswa dengan data valid dan nomor siswa tidak sama dengan nomor siswa lain  |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
| Test case 24         | Skenario 5               | Pengujian mengubah data siswa dengan data valid tapi nomor siswa sama dengan nomor siswa lain   |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Nomor guru telah digunakan</i> "   |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Nomor guru telah digunakan</i> "   |
| Test case 25         | Skenario 6               | Pengujian mengubah data siswa dengan mengosongkan form  |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 26         | Skenario 7               | Pengujian menghapus data siswa  |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |

**Tabel 7.5** Hasil pengujian manajemen data siswa (lanjutan)

| Manajemen Data Siswa |                          |  |
|----------------------|--------------------------|--|
| Test case 27         | Skenario 8               | Pengujian melihat data siswa                     |
|                      | Keluaran yang diharapkan | Sistem menampilkan data siswa dalam bentuk tabel |
|                      | Keluaran                 | Sistem menampilkan data siswa dalam bentuk tabel |

Dari 8 test case yang diuji pada Tabel 7.5, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada use case spesifikasi manajemen data siswa.

### 7.2.5 Pengujian Manajemen Data Mata Pelajaran

Hasil pengujian use case manajemen data mata pelajaran dapat dilihat pada Tabel 7.6.

**Tabel 7.6** Hasil pengujian manajemen data mata pelajaran

| Manajemen Data Mata Pelajaran |                          |   |
|-------------------------------|--------------------------|---|
| Test case 28                  | Skenario 1               | Pengujian menambah data mata pelajaran dan nama mata pelajaran belum ada di database sistem |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                          |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                          |
| Test case 29                  | Skenario 2               | Pengujian menambah data mata pelajaran tapi mata pelajaran sudah ada di database sistem     |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Nama mata pelajaran telah digunakan</i> "                |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>Nama mata pelajaran telah digunakan</i> "                |
| Test case 30                  | Skenario 3               | Pengujian menambah data mata pelajaran dengan mengosongkan form                             |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> "              |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> "              |

**Tabel 7.6** Hasil pengujian manajemen data mata pelajaran (lanjutan)

| Manajemen Data Mata Pelajaran |                          |   |
|-------------------------------|--------------------------|---|
| Test case 31                  | Skenario 4               | Pengujian mengubah data mata pelajaran dan nama mata pelajaran tidak sama dengan nama mata pelajaran lain |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
| Test case 32                  | Skenario 5               | Pengujian mengubah data mata pelajaran tapi nama mata pelajaran sama dengan nama mata pelajaran lain      |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Nama mata pelajaran telah digunakan</i> "                              |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>Nama mata pelajaran telah digunakan</i> "                              |
| Test case 33                  | Skenario 6               | Pengujian mengubah data mata pelajaran dengan mengosongkan form   |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> "                            |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> "                            |
| Test case 34                  | Skenario 7               | Pengujian menghapus data mata pelajaran   |
|                               | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
|                               | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
| Test case 35                  | Skenario 8               | Pengujian melihat data mata pelajaran   |
|                               | Keluaran yang diharapkan | Sistem menampilkan data mata pelajaran dalam bentuk tabel   |
|                               | Keluaran                 | Sistem menampilkan data mata pelajaran dalam bentuk tabel   |

Dari 8 test case yang diuji pada Tabel 7.6, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada use case spesifikasi manajemen data mata pelajaran.

## 7.2.6 Pengujian Manajemen Jam Pelajaran

Hasil pengujian *use case* manajemen jam pelajaran dapat dilihat pada Tabel 7.7.

**Tabel 7.7** Hasil pengujian manajemen jam pelajaran

| Manajemen Jam Pelajaran |                          |   |
|-------------------------|--------------------------|---|
| Test case 36            | Skenario 1               | Pengujian menambah jam pelajaran dan nama mata pelajaran belum ada di database sistem   |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                      |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                      |
| Test case 37            | Skenario 2               | Pengujian menambah jam pelajaran tapi jam pelajaran sudah ada di <i>database</i> sistem |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Jam pelajaran telah digunakan</i> "                  |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Jam pelajaran telah digunakan</i> "                  |
| Test case 38            | Skenario 3               | Pengujian menambah jam pelajaran dengan mengosongkan form                               |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> "          |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> "          |
| Test case 39            | Skenario 4               | Pengujian mengubah jam pelajaran dan jam pelajaran tidak sama dengan jam pelajaran lain |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                           |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                           |
| Test case 40            | Skenario 5               | Pengujian mengubah jam pelajaran tapi jam pelajaran sama dengan jam pelajaran lain      |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Jam pelajaran telah digunakan</i> "                  |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Jam pelajaran telah digunakan</i> "                  |

**Tabel 7.7** Hasil pengujian manajemen jam pelajaran (lanjutan)

| Manajemen Jam Pelajaran |                          |  |
|-------------------------|--------------------------|--|
| Test case 41            | Skenario 6               | Pengujian mengubah jam pelajaran dengan mengosongkan form                      |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> " |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> " |
| Test case 42            | Skenario 7               | Pengujian menghapus jam pelajaran  |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                 |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                 |
| Test case 43            | Skenario 8               | Pengujian melihat jam pelajaran  |
|                         | Keluaran yang diharapkan | Sistem menampilkan jam pelajaran dalam bentuk tabel                            |
|                         | Keluaran                 | Sistem menampilkan jam pelajaran dalam bentuk tabel                            |

Dari 8 *test case* yang diuji pada Tabel 7.7, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen jam pelajaran.

### 7.2.7 Pengujian Manajemen Data Kelas

Hasil pengujian *use case* manajemen data kelas dapat dilihat pada Tabel 7.8.

**Tabel 7.8** Hasil pengujian data kelas

| Manajemen Data Kelas |                          |   |
|----------------------|--------------------------|---|
| Test case 44         | Skenario 1               | Pengujian menambah data kelas   |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                |
| Test case 45         | Skenario 2               | Pengujian menambah data kelas tapi data kelas sudah ada di <i>database</i> sistem |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data kelas telah digunakan</i> "               |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data kelas telah digunakan</i> "               |

**Tabel 7.8** Hasil pengujian data kelas (lanjutan)

| Manajemen Data Kelas |                          |  |
|----------------------|--------------------------|--|
| Test case 46         | Skenario 3               | Pengujian menambah data kelas dengan mengosongkan form                         |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "              |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "              |
| Test case 47         | Skenario 4               | Pengujian mengubah data kelas dan data kelas tidak sama dengan data kelas lain |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                  |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                  |
| Test case 48         | Skenario 5               | Pengujian mengubah data kelas tapi data kelas sama dengan data kelas lain      |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data kelas telah digunakan</i> "            |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data kelas telah digunakan</i> "            |
| Test case 49         | Skenario 6               | Pengujian mengubah data kelas dengan mengosongkan form                         |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> " |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong</i> " |
| Test case 50         | Skenario 7               | Pengujian menghapus data kelas   |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                 |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                 |
| Test case 51         | Skenario 8               | Pengujian melihat data kelas   |
|                      | Keluaran yang diharapkan | Sistem menampilkan data kelas dalam bentuk tabel                               |
|                      | Keluaran                 | Sistem menampilkan data kelas dalam bentuk tabel                               |

Dari 8 test case yang diuji pada Tabel 7.8, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada use case spesifikasi manajemen data kelas.



### 7.2.8 Pengujian Manajemen Data Pengajar

Hasil pengujian *use case* manajemen data pengajar dapat dilihat pada Tabel 7.9.

**Tabel 7.9** Hasil pengujian manajemen data pengajar

| Manajemen Data Pengajar |                          |   |
|-------------------------|--------------------------|---|
| Test case 52            | Skenario 1               | Pengujian menambah data pengajar  |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
| Test case 53            | Skenario 2               | Pengujian menambah data pengajar dengan data valid tapi data pengajar sudah ada di <i>database</i> sistem   |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pengajar telah digunakan</i> "   |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Pengajar telah digunakan</i> "   |
| Test case 54            | Skenario 3               | Pengujian menambah data pengajar dengan mengosongkan form   |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 55            | Skenario 4               | Pengujian mengubah data pengajar  |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
| Test case 56            | Skenario 5               | Pengujian mengubah data pengajar dengan data yang sudah ada di <i>database</i> sistem   |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pengajar telah digunakan</i> "   |

**Tabel 7.9** Hasil pengujian manajemen data pengajar (lanjutan)

| Manajemen Data Pengajar |                          |   |
|-------------------------|--------------------------|---|
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Pengajar telah digunakan</i> "   |
| Test case 57            | Skenario 6               | Pengujian mengubah data pengajar dengan mengosongkan form   |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Nama kolom&gt; tidak boleh kosong*</i> " atau " <i>Pilih &lt;Nama kolom&gt;**</i> "<br><br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 58            | Skenario 7               | Pengujian menghapus data pengajar   |
|                         | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
|                         | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
| Test case 59            | Skenario 8               | Pengujian melihat data pengajar   |
|                         | Keluaran yang diharapkan | Sistem menampilkan data pengajar dalam bentuk tabel   |
|                         | Keluaran                 | Sistem menampilkan data pengajar dalam bentuk tabel   |

Dari 8 *test case* yang diuji pada Tabel 7.9, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen data pengajar.

### 7.2.9 Pengujian Manajemen Pembagian Kelas

Hasil pengujian *use case* manajemen pembagian kelas dapat dilihat pada Tabel 7.10.

**Tabel 7.10** Hasil pengujian manajemen pembagian kelas

| Manajemen Pembagian Kelas |                          |  |
|---------------------------|--------------------------|--|
| Test case 60              | Skenario 1               | Pengujian menambah data siswa baru pada kelas-kelas yang telah dibuat  |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Jumlah berhasil&gt; dari &lt;jumlah data terpilih&gt; data terpilih berhasil ditambahkan</i> "    |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Jumlah berhasil&gt; dari &lt;jumlah data terpilih&gt; data terpilih berhasil ditambahkan</i> "    |
| Test case 61              | Skenario 2               | Pengujian dengan mengosongkan form   |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Jumlah gagal&gt; dari &lt;jumlah data terpilih&gt; data terpilih tidak berhasil ditambahkan</i> " |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Jumlah gagal&gt; dari &lt;jumlah data terpilih&gt; data terpilih tidak berhasil ditambahkan</i> " |
| Test case 62              | Skenario 3               | Pengujian dengan data melebihi kuota kelas   |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Jumlah gagal&gt; dari &lt;jumlah data terpilih&gt; data terpilih tidak berhasil ditambahkan</i> " |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Jumlah gagal&gt; dari &lt;jumlah data terpilih&gt; data terpilih tidak berhasil ditambahkan</i> " |
| Test case 63              | Skenario 4               | Pengujian mengubah data siswa pada kelas yang telah diinputkan   |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "  |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "  |
| Test case 64              | Skenario 5               | Pengujian dengan mengosongkan form   |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "  |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "  |
| Test case 65              | Skenario 6               | Pengujian dengan data melebihi kuota kelas   |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Kuota kelas telah penuh</i> "   |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>Kuota kelas telah penuh</i> "   |

**Tabel 7.10** Hasil pengujian manajemen pembagian kelas (lanjutan)

| Manajemen Pembagian Kelas |                          |  |
|---------------------------|--------------------------|--|
| Test case 66              | Skenario 7               | Pengujian menghapus data siswa dari kelas yang ada             |
|                           | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> " |
|                           | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> " |
| Test case 67              | Skenario 8               | Pengujian melihat pembagian kelas                              |
|                           | Keluaran yang diharapkan | Sistem menampilkan pembagian kelas dalam bentuk tabel          |
|                           | Keluaran                 | Sistem menampilkan pembagian kelas dalam bentuk tabel          |

Dari 8 *test case* yang diuji pada Tabel 7.10, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen pembagian kelas.

### 7.2.10 Pengujian Manajemen Kalender Akademik

Hasil pengujian *use case* manajemen kalender akademik dapat dilihat pada Tabel 7.11.

**Tabel 7.11** Hasil pengujian manajemen kalender akademik

| Manajemen Kalender Akademik |                          |   |
|-----------------------------|--------------------------|---|
| Test case 68                | Skenario 1               | Pengujian menambah data kalender akademik                                 |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "        |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "        |
| Test case 69                | Skenario 2               | Pengujian menambah data dengan data yang sudah ada di database sistem     |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Tanggal kegiatan telah digunakan</i> " |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Tanggal kegiatan telah digunakan</i> " |
| Test case 70                | Skenario 3               | Pengujian menambah data kalender akademik dengan mengosongkan form        |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "         |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "         |

**Tabel 7.11** Hasil pengujian manajemen kalender akademik (lanjutan)

| Manajemen Kalender Akademik |                          |   |
|-----------------------------|--------------------------|---|
| Test case 71                | Skenario 4               | Pengujian mengubah data kalender akademik   |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
| Test case 72                | Skenario 5               | Pengujian mengubah data kalender akademik tapi data kalender akademik sama dengan data kalender akademik lain |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Tanggal kegiatan telah digunakan</i> "                                     |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Tanggal kegiatan telah digunakan</i> "                                     |
| Test case 73                | Skenario 6               | Pengujian mengubah data kalender akademik dengan mengosongkan form  |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "   |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt;</i> "   |
| Test case 74                | Skenario 7               | Pengujian menghapus data kalender akademik  |
|                             | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
|                             | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
| Test case 75                | Skenario 8               | Pengujian melihat data kalender akademik  |
|                             | Keluaran yang diharapkan | Sistem menampilkan data kalender akademik dalam bentuk tabel  |
|                             | Keluaran                 | Sistem menampilkan data kalender akademik dalam bentuk tabel  |

Dari 8 test case yang diuji pada Tabel 7.11, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada use case spesifikasi manajemen kalender akademik.

### 7.2.11 Pengujian Manajemen Jadwal Mata Pelajaran

Hasil pengujian use case manajemen jadwal mata pelajaran dapat dilihat pada Tabel 7.12.

**Tabel 7.12** Hasil pengujian manajemen jadwal mata pelajaran

| Manajemen Jadwal Mata Pelajaran |                          |   |
|---------------------------------|--------------------------|---|
| Test case 76                    | Skenario 1               | Pengujian menambah jadwal mata pelajaran  |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "  |
| Test case 77                    | Skenario 2               | Pengujian menambah jadwal mata pelajaran dengan satu guru mengajar pada jam pelajaran yang sama dalam satu hari   |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data tidak berhasil ditambahkan karena ada bentrokan jadwal guru</i> "         |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data tidak berhasil ditambahkan karena ada bentrokan jadwal guru</i> "         |
| Test case 78                    | Skenario 3               | Pengujian menambah jadwal mata pelajaran dengan mata pelajaran yang melebihi alokasi jam mengajar                 |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data tidak berhasil ditambahkan karena alokasi jam pelajaran sudah penuh</i> " |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data tidak berhasil ditambahkan karena alokasi jam pelajaran sudah penuh</i> " |
| Test case 79                    | Skenario 4               | Pengujian mengubah jadwal mata pelajaran  |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "   |
| Test case 80                    | Skenario 5               | Pengujian mengubah jadwal mata pelajaran dengan satu guru mengajar pada jam pelajaran yang sama dalam satu hari   |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data tidak berhasil diubah karena ada bentrokan jadwal guru</i> "              |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data tidak berhasil diubah karena ada bentrokan jadwal guru</i> "              |

**Tabel 7.12** Hasil pengujian manajemen jadwal mata pelajaran (lanjutan)

| Manajemen Jadwal Mata Pelajaran |                          |  |
|---------------------------------|--------------------------|--|
| Test case 81                    | Skenario 6               | Pengujian mengubah jadwal mata pelajaran dengan mata pelajaran yang melebihi alokasi jam mengajar            |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data tidak berhasil diubah karena alokasi jam pelajaran sudah penuh</i> " |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data tidak berhasil diubah karena alokasi jam pelajaran sudah penuh</i> " |
| Test case 82                    | Skenario 7               | Pengujian menghapus jadwal mata pelajaran  |
|                                 | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "   |
|                                 | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "   |
| Test case 83                    | Skenario 8               | Pengujian melihat jadwal mata pelajaran  |
|                                 | Keluaran yang diharapkan | Sistem menampilkan jadwal mata pelajaran dalam bentuk tabel  |
|                                 | Keluaran                 | Sistem menampilkan jadwal mata pelajaran dalam bentuk tabel  |

Dari 8 *test case* yang diuji pada Tabel 7.12, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen jadwal mata pelajaran.

### 7.2.12 Pengujian Manajemen Kegiatan

Hasil pengujian *use case* manajemen kegiatan dapat dilihat pada Tabel 7.13.

**Tabel 7.13** Hasil pengujian manajemen kegiatan

| Manajemen Kegiatan |                          |   |
|--------------------|--------------------------|---|
| Test case 84       | Skenario 1               | Pengujian menambah data kegiatan  |
|                    | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                    |
|                    | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                    |
| Test case 85       | Skenario 2               | Pengujian menambah data kegiatan dengan data yang sudah ada di <i>database</i> sistem |
|                    | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Kegiatan telah digunakan</i> "                     |
|                    | Keluaran                 | Sistem menampilkan notifikasi " <i>Kegiatan telah digunakan</i> "                     |

**Tabel 7.13** Hasil pengujian manajemen kegiatan (lanjutan)

| Manajemen Kegiatan |                          |   |
|--------------------|--------------------------|---|
| Test case 86       | Skenario 3               | Pengujian menambah data kegiatan dengan mengosongkan form   |
|                    | Keluaran yang diharapkan | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong*" atau "Pilih <Nama kolom>**"<br>* Form berupa masukan<br>** Form berupa pilihan |
|                    | Keluaran                 | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong*" atau "Pilih <Nama kolom>**"<br>* Form berupa masukan<br>** Form berupa pilihan |
| Test case 86       | Skenario 4               | Pengujian mengubah data kegiatan  |
|                    | Keluaran yang diharapkan | Sistem menampilkan notifikasi "Data berhasil diubah"  |
|                    | Keluaran                 | Sistem menampilkan notifikasi "Data berhasil diubah"  |
| Test case 88       | Skenario 5               | Pengujian mengubah data kegiatan dengan mengosongkan form   |
|                    | Keluaran yang diharapkan | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong"   |
|                    | Keluaran                 | Sistem menampilkan notifikasi "<Nama kolom> tidak boleh kosong"   |
| Test case 89       | Skenario 6               | Pengujian menghapus data kegiatan   |
|                    | Keluaran yang diharapkan | Sistem menampilkan notifikasi "Data berhasil dihapus"   |
|                    | Keluaran                 | Sistem menampilkan notifikasi "Data berhasil dihapus"   |
| Test case 90       | Skenario 7               | Pengujian melihat kegiatan  |
|                    | Keluaran yang diharapkan | Sistem menampilkan kegiatan dalam bentuk tabel  |
|                    | Keluaran                 | Sistem menampilkan kegiatan dalam bentuk tabel  |

Dari 7 test case yang diuji pada Tabel 7.13, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada use case spesifikasi manajemen kegiatan.



### 7.2.13 Pengujian Manajemen Presensi Siswa

Hasil pengujian *use case* manajemen presensi siswa dapat dilihat pada Tabel 7.14.

**Tabel 7.14** Hasil pengujian manajemen presensi siswa

| Manajemen Presensi Siswa |                          |   |
|--------------------------|--------------------------|---|
| Test case 91             | Skenario 1               | Pengujian menambah data presensi siswa  |
|                          | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Jumlah berhasil&gt; dari &lt;jumlah data terpilih&gt; data terpilih berhasil ditambahkan</i> " |
|                          | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Jumlah berhasil&gt; dari &lt;jumlah data terpilih&gt; data terpilih berhasil ditambahkan</i> " |
| Test case 92             | Skenario 2               | Pengujian mengubah data presensi siswa  |
|                          | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>&lt;Jumlah berhasil&gt; dari &lt;jumlah data terpilih&gt; data terpilih berhasil diubah</i> "      |
|                          | Keluaran                 | Sistem menampilkan notifikasi " <i>&lt;Jumlah berhasil&gt; dari &lt;jumlah data terpilih&gt; data terpilih berhasil diubah</i> "      |
| Test case 93             | Skenario 3               | Pengujian menghapus data presensi siswa   |
|                          | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
|                          | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "  |
| Test case 94             | Skenario 4               | Pengujian menampilkan semua presensi siswa.   |
|                          | Keluaran yang diharapkan | Sistem menampilkan data presensi siswa dalam bentuk tabel   |
|                          | Keluaran                 | Sistem menampilkan data presensi siswa dalam bentuk tabel   |

Dari 4 *test case* yang diuji pada Tabel 7.14, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen presensi siswa.

### 7.2.14 Pengujian Manajemen Wali Kelas

Hasil pengujian *use case* manajemen wali kelas dapat dilihat pada Tabel 7.15.

Tabel 7.15 Hasil pengujian manajemen wali kelas

| Manajemen Wali Kelas |                          |  |
|----------------------|--------------------------|--|
| Test case 95         | Skenario 1               | Pengujian menambah data wali kelas   |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                 |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil ditambahkan</i> "                 |
| Test case 96         | Skenario 2               | Pengujian menambah data wali kelas dengan data sudah ada di <i>database</i> sistem |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Wali kelas telah digunakan</i> "                |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Wali kelas telah digunakan</i> "                |
| Test case 97         | Skenario 3               | Pengujian menambah data wali kelas dengan mengosongkan form                        |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt; "</i>                  |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt; "</i>                  |
| Test case 98         | Skenario 4               | Pengujian mengubah data wali kelas   |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                      |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil diubah</i> "                      |
| Test case 99         | Skenario 5               | Pengujian mengubah data wali kelas dengan data sudah ada di <i>database</i> sistem |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Wali kelas telah digunakan</i> "                |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Wali kelas telah digunakan</i> "                |
| Test case 100        | Skenario 6               | Pengujian mengubah data wali kelas dengan mengosongkan form                        |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt; "</i>                  |
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Pilih &lt;Nama kolom&gt; "</i>                  |
| Test case 101        | Skenario 7               | Pengujian menghapus data wali kelas  |
|                      | Keluaran yang diharapkan | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> "                     |

**Tabel 7.15** Hasil pengujian manajemen wali kelas (lanjutan)

| Manajemen Wali Kelas |                          |  |
|----------------------|--------------------------|--|
|                      | Keluaran                 | Sistem menampilkan notifikasi " <i>Data berhasil dihapus</i> " |
| Test case 102        | Skenario 8               | Pengujian melihat data wali kelas                              |
|                      | Keluaran yang diharapkan | Sistem menampilkan data wali kelas dalam bentuk tabel          |
|                      | Keluaran                 | Sistem menampilkan data wali kelas dalam bentuk tabel          |

Dari 8 *test case* yang diuji pada Tabel 7.15, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen wali kelas.

### 7.2.15 Pengujian Manajemen Rekap

Hasil pengujian *use case* manajemen rekap dapat dilihat pada Tabel 7.16.

**Tabel 7.16** Hasil pengujian manajemen rekap

| Manajemen Rekap |                          |  |
|-----------------|--------------------------|--|
| Test case 103   | Skenario 1               | Pengujian membuat data rekap                     |
|                 | Keluaran yang diharapkan | Sistem menampilkan data rekap dalam bentuk tabel |
|                 | Keluaran                 | Sistem menampilkan data rekap dalam bentuk tabel |

Dari 1 *test case* yang diuji pada Tabel 7.16, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi manajemen rekap.

### 7.2.16 Pengujian Mencetak Laporan

Hasil pengujian *use case* mencetak laporan dapat dilihat pada Tabel 7.17.

**Tabel 7.17** Hasil pengujian mencetak laporan

| Mencetak Laporan |                          |                                  |
|------------------|--------------------------|----------------------------------|
| Test case 104    | Skenario 1               | Pengujian membuat data rekap     |
|                  | Keluaran yang diharapkan | Sistem menampilkan halaman cetak |
|                  | Keluaran                 | Sistem menampilkan halaman cetak |

Dari 1 *test case* yang diuji pada Tabel 7.17, didapatkan hasil keluaran sesuai dengan hasil keluaran yang diharapkan seperti yang sudah didefinisikan pada *use case* spesifikasi mencetak laporan.

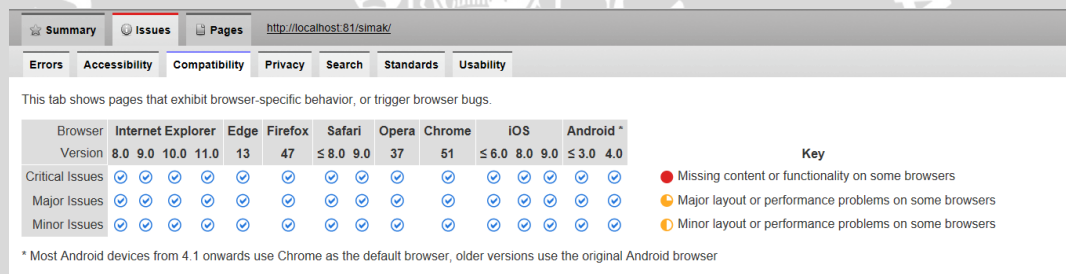
### 7.3 Pengujian *Compatibility*

*Compatibility* merupakan salah satu kebutuhan non fungsional dari sistem informasi manajemen akademik maka dari itu diperlukan pengujian *compatibility* untuk memastikan sistem yang dibangun dapat berjalan di browser seperti *Google Chrome*, *Mozilla Firefox*, *Internet Explorer*, *Opera*.

*Tools* yang digunakan untuk melakukan pengujian *compatibility* adalah *SortSite 5 Trial*. *SortSite* dipilih karena mampu menguji perangkat lunak berbasis *web* secara keseluruhan pada setiap halaman yang dimiliki. *Sortsite* menguji *compability* berdasarkan 3 hal utama yaitu :

1. *Critical Issues*: Apakah ada kesalahan dengan hilangnya konten atau fungsional pada browser.
2. *Major Issues*: berdasarkan pada *layout* utama seperti bentuk table , posisi menu , ukuran gambar dan lainnya.
3. *Minor Issues*: berdasarkan pada *layout* kecil seperti perbedaan jarak tulisan, pewarnaan tombol dan lainnya.

Berikut hasil pengujiannya dapat dilihat pada Gambar 7.1.



| Browser         | Internet Explorer | Edge | Firefox | Safari    | Opera | Chrome | iOS           | Android * |
|-----------------|-------------------|------|---------|-----------|-------|--------|---------------|-----------|
| Version         | 8.0 9.0 10.0 11.0 | 13   | 47      | ≤ 8.0 9.0 | 37    | 51     | ≤ 6.0 8.0 9.0 | ≤ 3.0 4.0 |
| Critical Issues | ✓ ✓ ✓ ✓           | ✓    | ✓       | ✓ ✓       | ✓     | ✓      | ✓ ✓ ✓         | ✓ ✓       |
| Major Issues    | ✓ ✓ ✓ ✓           | ✓    | ✓       | ✓ ✓       | ✓     | ✓      | ✓ ✓ ✓         | ✓ ✓       |
| Minor Issues    | ✓ ✓ ✓ ✓           | ✓    | ✓       | ✓ ✓       | ✓     | ✓      | ✓ ✓ ✓         | ✓ ✓       |

\* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

**Gambar 7.1** Hasil pengujian *Compatibility* menggunakan *SortSite 5 Trial*

Pada Gambar 7.1 menunjukkan bahwa hasil dari pengujian *compatibility* merupakan sistem dapat berjalan dengan baik pada *Google Chrome* (versi 51), *Mozilla Firefox* (versi 47), *Internet Explorer* (versi 8.0, 9.0, 10.0, 11.0), *Safari* (versi ≤ 8.0, 9.0), *Edge* (versi 13), *Opera* (versi 37), *browser Android* (versi ≤ 3.0, 4.0) dan *browser iOS* (versi ≤ 6.0, 8.0, 9.0).

### 7.4 Analisis Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan didapatkan hasil analisis sebagai berikut :

1. Pengujian fungsionalitas yang dilakukan pada 104 *test case* menghasilkan nilai 100% valid atau sesuai dengan keluaran yang diharapkan pada sistem.
2. Pengujian non-fungsional *compatibility* yang dilakukan dengan menggunakan *tools SortSite 5 Trial* menunjukkan bahwa perangkat lunak dapat berjalan dengan baik pada semua target *browser*.

## BAB 8 PENUTUP

### 8.1 Kesimpulan

Berdasarkan dari penelitian konstruksi sistem informasi manajemen akademik menghasilkan kesimpulan sebagai berikut:

1. Penelitian Analisa dan Perancangan Sistem Informasi Manajemen Akademik dievaluasi dengan menggunakan WEBUSE. Evaluasi dilakukan dengan cara memberikan kuesioner dan prototipe sistem informasi manajemen akademik kepada calon pengguna. Adapun hasil evaluasi sebagai berikut:
  - a. Alur pengguna dengan *basic flow* dan *alternative flow* sudah sesuai.
  - b. Respon sistem sudah sesuai dengan harapan pengguna karena hasil yang ditampilkan oleh sistem sudah sesuai dengan keinginan pengguna .
  - c. Navigasi sudah sesuai dengan harapan pengguna karena posisi dan isi navigasi sudah jelas. *Link* mudah dikenali dengan ukuran huruf dan tombol yang jelas dan dapat digunakan.
  - d. Antarmuka pengguna mudah dipahami sehingga mudah dioperasikan dan sudah konsisten dalam desainnya.
2. Analisis dan perancangan sistem informasi manajemen akademik dapat diimplementasikan menjadi sistem informasi dengan melakukan penyesuaian terhadap *framework CodeIgniter*. Penyesuaian dilakukan dengan melakukan perubahan pada *class-class controller* menjadi kelas turunan dari *class CI\_Controller* dan *class-class model* menjadi kelas turunan dari *class CI\_Model* dimana *class CI\_Controller* dan *CI\_Model* merupakan *class* yang terdapat dalam *framework*.
3. Pengujian sistem informasi manajemen akademik dilakukan dengan metode *functional black-box testing* dan *compatibility testing* dengan penjelasan sebagai berikut:
  - a. Pengujian fungsional dilakukan dengan cara melakukan observasi keluaran berdasarkan masukan tertentu pada *test case* yang telah dikembangkan dari *use scenario* yang berjumlah 104 butir.
  - b. Pengujian *compatibility* dilakukan dengan menggunakan *tools SortSite* versi 5 *trial*. Adapun cara mengujinya adalah (i) membuka *tools* tersebut, (ii) pilih menu *compatibility*, (iii) masukkan *link* sistem informasi, (iv) tekan tombol untuk memulai pengujian, tunggu hingga proses pengujian selesai dan hasil pengujian keluar.

4. Berdasarkan hasil pengujian fungsional dan *compatibility* yang telah dilakukan, diperoleh hasil:
  - a. 100% valid pada 104 *test case* yang diuji.
  - b. Sistem bisa berjalan pada *browser Google Chrome* (versi 51), *Mozilla Firefox* (versi 47), *Internet Explorer* (versi 8.0, 9.0, 10.0, 11.0), *Safari* (versi  $\leq 8.0, 9.0$ ), *Edge* (versi 13), *Opera* (versi 37), *browser Android* (versi  $\leq 3.0, 4.0$ ) dan *browser iOS* (versi  $\leq 6.0, 8.0, 9.0$ ).

## 8.2 Saran

Diharapkan pada penelitian selanjutnya sistem informasi manajemen akademik agar dapat dikembangkan sebagai berikut:

1. Melakukan pengembangan agar sistem mampu membuat jadwal mata pelajaran secara otomatis yang mana ketika data guru, kelas dan jam pelajaran sudah tersedia maka proses pembuatan jadwal bisa dilakukan secara otomatis.
2. Mengembangkan sistem informasi manajemen akademik berbasis *mobile* ataupun *desktop*.



## DAFTAR PUSTAKA

- A.S, R. & Shalahuddin, M., 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informastika.
- Agarwal, B. B., Tayal, S. P. & Gupta, M., 2010. *Software Engineering & Testing*.
- Basuki, A. P., 2014. *Proyek Membangun Website Berbasis PHP dengan Codeigniter*. Yogyakarta: Lokomedia.
- Budiharto, D. W., 2014. *Membangun web handal dengan menggunakan J2EE dan MVC*.
- Candrawati, W. M., 2014. *Rancang Bangun Sistem Informasi Praktikum pada Program Teknologi Informasi dan Ilmu Komputer*. Malang: Universitas Brawijaya.
- Chiew, T. K. & Salim, S. S., 2003. *WEBUSE: WEBSITE USABILITY EVALUATION TOOL*. s.l.:Malaysian Journal of Computer Science.
- EllisLab, 2016. [Online] Tersedia di : <http://www.codeigniter.com/> [Diakses: 21 November 2016].
- Fatha, I., 2017. *Analisis dan Perancangan Sistem Informasi Manajemen Akademik*. Malang: Universitas Brawijaya.
- Gilmore, W. J., 2010. *Beginning PHP and MySQL*. 4th penyunt. Apress.
- Guru99, 2016. [Online] Tersedia di : <http://www.guru99.com/compatibility-testing.html> [Diakses: 28 November 2016].
- Norad, 2017. *About Evaluation*. [Online] Tersedia di : <https://www.norad.no/en/front/evaluation/what-is-evaluation/> [Diakses: 18 Januari 2017].
- PowerMapper, 2016. [Online] Tersedia di : <http://www.powermapper.com/products/sortsite/checks/browser-compatibility/> [Diakses: 28 November 2016].
- Pressman, R. S., 2012. *Rekayasa Perangkat Lunak - Buku Satu, Pendekatan Praktisi*. 7 penyunt. Yogyakarta: Andi Yogyakarta.
- Sidik, B., 2012. *Framework CodeIgniter*. Bandung: Informatika.
- Sommerville, I., 2009. *Software Engineering*. 9th penyunt. China Machire Press.
- Sunyoto, A., 2007. *AJAX Membangun Web dengan Teknologi Asynchronous Javascript & XML*. Yogyakarta: Andi Offset.