

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

#### **A. Tahap *Inception***

Tahap *inception* memiliki fokus terhadap pemodelan proses bisnis (*business modeling*) dan mendefinisikan kebutuhan sistem (*requirements*) yang akan dibuat. Kegiatan yang dikerjakan pada fase *inception* meliputi: memahami ruang lingkup proyek mulai dari membangun kasus bisnis, analisis kebutuhan, waktu pengerjaan, pembiayaan, dan pengelolaan risiko.

##### **1. *Business Modelling***

###### **a. Studi Literatur**

Digunakan untuk memperoleh pemahaman dasar terkait manajemen pendidikan, sistem informasi manajemen, pembelajaran *online* dan metode pengembangan perangkat lunak. Studi literatur bersumber dari buku teks, jurnal ilmiah, makalah seminar, prosiding, laporan penelitian dan *website* pusat-pusat kajian yang memiliki reputasi.

###### **b. Observasi dan Wawancara**

Kegiatan observasi dilakukan di SMK Negeri 2 Wonosari untuk mengamati proses manajemen sekolah khususnya manajemen pembelajaran yang diterapkan. Wawancara dilakukan bersama Drs. Bekti Yahrini M.Pd selaku Wakil Kepala Sekolah II Bidang Kurikulum, Bapak Wasno S.T selaku Ketua Program Keahlian Teknik Komputer dan Informatika (TIK), beberapa guru normatif dan adaptif, serta sejumlah siswa. Berikut beberapa keterangan yang didapatkan selama proses observasi dan wawancara:

- 1) Pelaksanaan manajemen pembelajaran siswa di SMKN 2 Wonosari terdiri dari beberapa tahap seperti yang tersaji pada Gambar 12.



Gambar 12. Tahapan Manajemen Pembelajaran di SMKN 2 Wonosari

- 2) Hasil wawancara dengan Wakil Kepala Sekolah II Bidang Kurikulum didapatkan beberapa keterangan sebagai berikut:
- Koordinasi dan pengarahan penyusunan administrasi guru masih terbatas, mengakibatkan perangkat pembelajaran guru tidak sesuai dengan program kurikulum. Proses verifikasi dokumen administrasi masih menggunakan *hardcopy* sehingga menyebabkan penumpukan berkas di ruangan, mobilitas terbatas, serta hanya bisa dilakukan pada jam kerja sekolah.
  - Pendistribusian berkas atau *file* sekolah belum terpusat menjadikan penyebaran informasi lamban, tidak seragam, serta terjadinya duplikasi. Dibutuhkan wadah pemasaran dokumen agar memudahkan dalam pengelolaan, pengawasan, dan menghemat waktu proses pencarian.

- c) Membutuhkan media untuk melakukan pemantauan aktivitas belajar kelas dan kinerja pengajaran guru melalui berita acara pengajaran yang tertulis pada buku kemajuan kelas.
  - d) Buku kemajuan kelas yang berisikan catatan pengajaran guru dan presensi kehadiran siswa tidak terdokumentasikan dengan baik, karena buku sering hilang atau rusak sehingga data tidak terbaca. Mengakibatkan proses rekapitulasi dan pelaporan kepada kepala sekolah terhambat.
  - e) Pemantauan Laporan Hasil Belajar (LHB) tiap kelas secara langsung oleh staf kurikulum diperlukan untuk mempercepat proses penyusunan rapor siswa, penyelesaian administrasi, dan tindakan evaluasi. Tidak perlu menggunakan mekanisme lama, dimana guru mata pelajaran membuat laporan nilai akhir secara manual, memberikan kepada wali kelas, kemudian wali kelas menyusunnya ke dalam rapor dan diserahkan pada pihak kurikulum.
- 3) Hasil wawancara dengan beberapa guru mata pelajaran didapatkan keterangan sebagai berikut:
- a) Penyimpanan berkas guru belum terkelola dengan baik, dibutuhkan mekanisme pengelolaan yang bisa menjadikan berkas mudah diakses kembali dan fleksibel untuk digunakan.
  - b) Kesulitan dalam melakukan kontrol dan pengawasan aktivitas belajar siswa diluar jam pembelajaran sekolah.
  - c) Memerlukan strategi dan pendekatan pembelajaran modern untuk keperluan: penyediaan sumber belajar, pelaksanaan aktivitas belajar, penilaian, dan pelaporan.

- d) Perkembangan belajar siswa tidak terekam dengan baik, diperlukan media pencatatan hasil belajar dari waktu ke waktu untuk setiap aktivitas belajar yang telah selesai diikuti. Hasilnya berguna sebagai bahan evaluasi dan perencanaan pembelajaran selanjutnya.
  - e) Administrasi buku kemajuan kelas rawan rusak dan hilang sehingga catatan pengajaran tidak terdokumentasi dengan baik. Proses rekapitulasi serta pelaporan masih dilakukan secara manual.
- 4) Hasil wawancara dengan peserta didik didapatkan beberapa keterangan sebagai berikut:
- a) Sering lupa mengerjakan tugas kelas karena banyaknya kegiatan di sekolah.
  - b) Tugas siswa sering terselip, tertukar dengan kelas lain, atau bahkan hilang ketika pengumpulannya menggunakan kertas. Waktu pengumpulan terbatas pada jam-jam pelajaran sekolah.
  - c) Adanya keterlambatan informasi tugas kelas yang disampaikan oleh guru piket ketika guru pengajaran tidak hadir. Pemberitahuan informasi sekolah terbatas melalui papan pengumuman.
  - d) Penilaian yang dilakukan oleh guru tidak bisa dipantau langsung oleh siswa, mereka menganggap kurangnya transparansi nilai.
  - e) Mengunduh transkrip nilai untuk keperluan mencari beasiswa dan mendaftar pekerjaan sehingga tidak perlu repot mengurus administrasi peminjaman, membongkar arsip dan mengembalikan ke tempatnya.

c. Deskripsi Produk

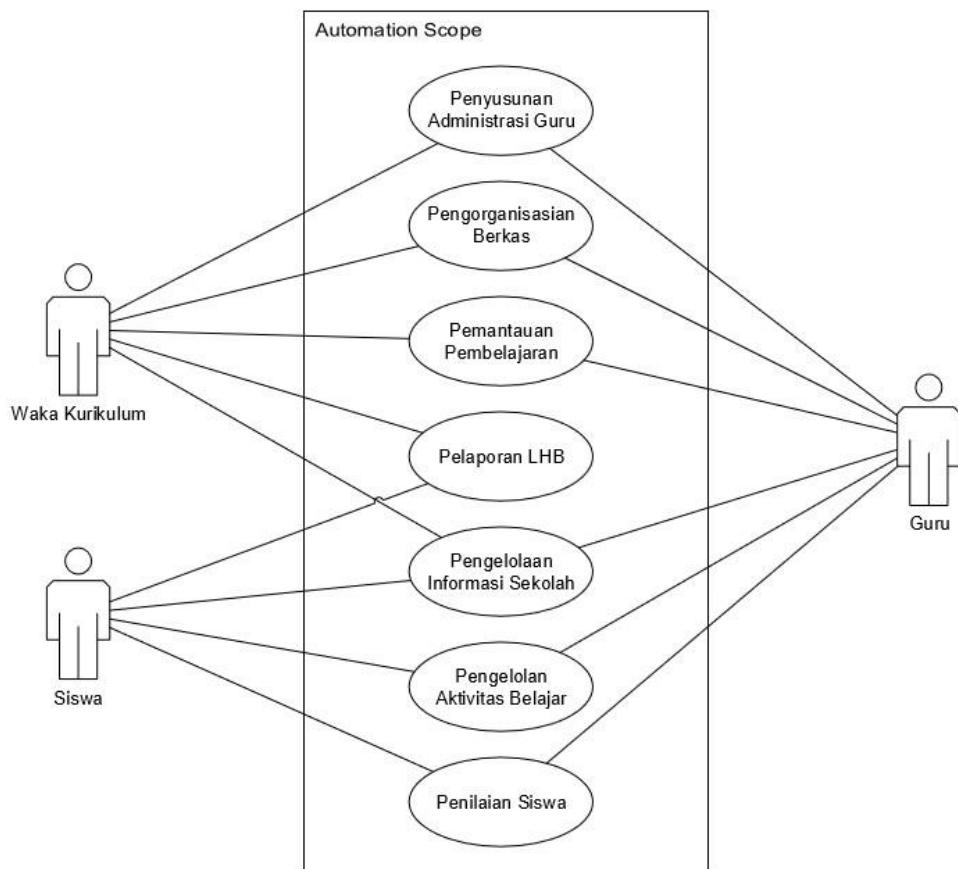
Pengembangan sistem informasi manajemen pembelajaran memiliki tujuan untuk membantu proses manajemen sekolah terutama pengelolaan pembelajaran yang meliputi:

- 1) Tahap perencanaan, staf kurikulum bersama guru bisa saling melakukan koordinasi dalam penyusunan dan verifikasi dokumen administrasi.
- 2) Tahap pengorganisasian, memfasilitasi pengelolaan dokumen sekolah dan guru mulai dari penyimpanan, pengaturan hak akses dan pendistribusian secara terpusat.
- 3) Tahap pelaksanaan dan pengendalian:
  - a) Staf kurikulum memantau proses pelaksanaan pembelajaran kelas melalui Berita Acara Pengajaran (BAP) yang telah dibuat guru.
  - b) Guru bisa membagikan materi pelajaran, membuat aktivitas belajar, mengisi berita acara pengajaran, melakukan penilaian, melaporkan hasil belajar.
  - c) Siswa dapat mengakses informasi pembelajaran kelas, mengumpulkan tugas (*online submission*), melihat catatan belajar individu, cetak transkrip nilai.
- 4) Tahap evaluasi, guru bisa melakukan pemantauan pembelajaran siswa, memberikan catatan belajar dan nilai akhir. Sedangkan staf kurikulum bisa melihat seluruh rekapitulasi buku kemajuan dan LHB kelas.

d. *Business Case*

Setelah mendapatkan proses bisnis manajemen pembelajaran, maka dilanjutkan dengan pemodelan menggunakan *business use case*. *Business use case* adalah kumpulan *workflow* yang terhubung di dalam sebuah organisasi,

menjelaskan apa yang dikerjakan oleh organisasi tersebut. Langkah selanjutnya mengidentifikasi kasus yang dapat diselesaikan dengan sistem dan memasukannya kedalam *automation scope* sebagai acuan pengembangan.



Gambar 13. *Business Case* Sistem Informasi Manajemen Pembelajaran

Gambar 13 memperlihatkan *automation scope* dari *business case*, terdiri dari tujuh *use case* yang akan menjadi acuan dalam pembuatan sistem informasi manajemen pembelajaran, diantaranya:

- 1) Penyusunan administrasi guru
- 2) Pengorganisasian berkas
- 3) Pengelolaan informasi sekolah
- 4) Pengelolaan aktivitas belajar

- 5) Penilaian siswa
- 6) Pemantauan pembelajaran
- 7) Pelaporan LHB (Laporan Hasil Belajar)

## 2. *Requirements*

Peneliti mengidentifikasi kebutuhan fungsional sistem manajemen pembelajaran siswa berdasarkan kasus bisnis yang ada dalam *automation scope*.

Hasil dari proses identifikasi kebutuhan fungsional diantaranya:

- a. Pengguna sistem terbagi menjadi tiga kategori yaitu staf kurikulum, guru mata pelajaran, dan siswa.
- b. Kebutuhan fungsional untuk staf kurikulum meliputi:
  - 1) Memverifikasi dokumen administrasi milik guru.
  - 2) Mengorganisasikan dokumen milik sekolah.
  - 3) Mengelola data jurusan dan kelas.
  - 4) Mengelola data pengguna guru dan siswa.
  - 5) Mengelola data mata pelajaran dan jadwal pelajaran.
  - 6) Mengelola data informasi sekolah.
  - 7) Memantau pembelajaran setiap kelas.
  - 8) Melihat laporan hasil belajar (LHB) setiap kelas.
- c. Kebutuhan fungsional guru mata pelajaran meliputi:
  - 1) Menyusun dokumen administrasi bersama admin.
  - 2) Mengorganisasikan dokumen individu.
  - 3) Membagikan informasi dan materi pembelajaran.
  - 4) Memberikan penugasan dan penilaian kepada siswa.

- 5) Melacak aktivitas belajar siswa.
  - 6) Mengelola buku kemajuan kelas.
  - 7) Mengelola nilai akhir mata pelajaran.
- d. Kebutuhan fungsional untuk siswa meliputi:
- 1) Mendaftar ke dalam mata pelajaran kelas.
  - 2) Melihat informasi pembelajaran kelas dan sekolah.
  - 3) Mengakses materi pembelajaran.
  - 4) Melihat catatan seluruh tugas kelas.
  - 5) Mengumpulkan tugas.
  - 6) Melihat catatan belajar individu setiap mata pelajaran.
  - 7) Melihat laporan hasil belajar.

### **3. *Analysis & Design***

Belum ada aktivitas *analysis* dan *design* karena fase *inception* lebih pada memodelkan proses bisnis, mendefinisikan kebutuhan sistem dan mendeskripsikan ruang lingkup proyek.

### **4. *Implementation***

Belum ada aktivitas *implementation* karena fase *inception* lebih pada memodelkan proses bisnis, mendefinisikan kebutuhan sistem dan mendeskripsikan ruang lingkup proyek.

### **5. *Test***

Belum ada aktivitas *test* karena fase *inception* lebih pada memodelkan proses bisnis, mendefinisikan kebutuhan sistem dan mendeskripsikan ruang lingkup proyek.

## 6. *Deployment*

Belum ada aktivitas *deployment* karena fase *inception* lebih pada memodelkan proses bisnis, mendefinisikan kebutuhan sistem dan mendeskripsikan ruang lingkup proyek.

## 7. *Configuration & Change Management*

Berkaitan dengan aktivitas manajemen dalam sebuah tim pengembang, peneliti tidak melakukan aktivitas proses ini karena proyek dikerjakan sendiri.

## 8. *Project Management*

Mengelola proyek dengan membuat perencanaan waktu dan target agar proses pengembangan berjalan dengan efektif dan efisien. Merinci keseluruhan aktivitas dari setiap fase pengembangan dengan membuat deskripsi kerja, waktu pengerjaan dan keterkaitan antara masing-masing pekerjaan (dependensi). Contoh dependensi adalah untuk mulai bekerja dengan A2 terlebih dahulu harus menyelesaikan pekerjaan A1. Rincian aktivitas pengembangan tersaji di Tabel 12.

Tabel 12. Rincian Aktivitas Pengembangan

No	ID	Nama Aktivitas	Durasi	Predecessor
1	A1	<b><i>Business Modelling</i></b> Menetapkan <i>business case</i> sistem dan memahami ruang lingkup proyek.	7 Hari	
2	A2	<b><i>Requirements</i></b> Mendefinisikan kebutuhan sistem untuk memodelkan persyaratan sistem.	7 Hari	A1
3	A3	<b><i>Analysis &amp; Design</i></b> Membangun kerangka kerja arsitektur sistem, mengembangkan rencana proyek.		

No	ID	Nama Aktivitas	Durasi	Predecessor
	A3.1	<b>Pemodelan UML</b> Memodelkan sistem menggunakan UML: <i>use case diagram, activity diagram, sequence diagram, class diagram,</i>	14 Hari	A2
	A3.2	<b>Merancang Basis Data</b> Pembuatan <i>ERD</i> , implementasi basis data	7 Hari	A2
	A3.3	<b>Merancang Antarmuka</b> Pembuatan desain <i>user interface</i> halaman <i>web</i> .	14 Hari	A2
4	A4	<b>Implementation</b> Pengembangan komponen dan fitur-fitur sistem berdasarkan desain arsitektur sistem.		
	A4.1	Komponen Admin (Staf kurikulum)	21 Hari	A3
	A4.2	Komponen Guru Mata Pelajaran	21 Hari	A3
	A4.3	Komponen Siswa	14 Hari	A3
5	A5	<b>Test</b> Pengujian kualitas perangkat lunak ISO 25010.	14 Hari	A4
6	A6	<b>Deployment</b> Rilis produk, pembuatan dokumentasi penggunaan dan instalasi sistem.	7 Hari	A5

Tabel 13. Penjadwalan Penggerjaan Sistem Informasi Manajemen

No	Nama Aktivitas	Bulan											
		1				2				3			
		1	2	3	4	1	2	3	4	1	2	3	4
1	<i>Business Modelling</i>												
2	<i>Requirements</i>												
3	<i>Analysis &amp; Design</i>												
	Pemodelan UML												
	Desain Basis Data												

	Desain Antarmuka												
4	<i>Implementation</i>												
	Komponen Waka II Kurikulum												
	Komponen Guru Mapel												
	Komponen Siswa												
5	<i>Test</i>												
6	<i>Deployment</i>												

Jadwal tersebut menjelaskan *core workflow* atau kegiatan inti dari proses pengembangan sistem informasi manajemen pembelajaran siswa. Pelaksanaan aktivitas proses dikerjakan secara iteratif/berulang mengikuti kebutuhan di lapangan. Sebagai contohnya, aktivitas perancangan antarmuka sistem dapat dilakukan ulang saat proses membuat kode program.

## 9. *Environment*

Menyediakan keperluan pengembangan termasuk *hardware* dan *software* sesuai dengan proyek yang akan dikerjakan. Penggunaan *hardware* meliputi seperangkat laptop dengan spesifikasi tertera pada Tabel 14, sedangkan detail *software* yang digunakan tersaji pada Tabel 15.

Tabel 14. Spesifikasi *Hardware* Laptop

No	Spesifikasi
1	<i>Processor</i> : AMD FX-7600P Radeon R7, 4C+8G (4CPUs),~ 2.7GHz
2	RAM: 4096 MB
3	VGA: AMD Radeon (TM) R7 Graphics, 2267 MB
4	<i>Hard Disk</i> : 1000 GB
5	<i>Operating System</i> : Windows 10 Pro - 64 Bit

Tabel 15. *Daftar Software*

No	Nama	Deskripsi
1	Apache/2.4.37	Bertanggungjawab menangani <i>request response</i> HTTP dan logging informasi sebagai <i>web server</i> .
2	MySQL	Perangkat lunak sistem manajemen basis data SQL (DBMS).
3	Visual Studio Code	<i>Text editor</i> untuk menulis kode program.
4	Laravel 5.7	<i>Framework</i> aplikasi <i>web</i> berbasis PHP.
5	Modelio	Memodelkan <i>requirements systems</i> dengan UML.
6	Pencil v.3.0.4	Merancang desain berupa <i>wireframe</i> sistem.
7	Bitbucket	Aplikasi VCS yang berguna mengelola file projek.
8	Web Browser	Menjalankan aplikasi <i>web</i> yang dikembangkan.

## B. Tahap *Elaboration*

Tahap *elaboration* memiliki fokus utama terhadap perancangan/memodelkan arsitektur sistem. Menganalisis dan mendesain sistem, kemudian menerapkannya menjadi sebuah purwarupa sistem. Kegiatan yang dikerjakan pada fase *elaboration* meliputi: pembuatan *use case* berdasarkan kasus dan entitas eksternal yang telah teridentifikasi pada *business case*, membuat *requirement model* untuk sistem dengan *Unified Modeling Language(UML)*, *Entity Relationship Diagram (ERD)* dan desain antarmuka.

## 1. ***Business Modeling***

### a. Deskripsi produk

Mendeskripsikan kembali sistem dengan mengidentifikasi pengguna (aktor), aktivitas yang terjadi (kata kerja), objek yang terlibat (kata benda) termasuk permasalahan yang timbul. Secara detail diuraikan sebagai berikut:

Sistem informasi manajemen pembelajaran berbasis *web* merupakan sistem yang dikembangkan untuk membantu pengelolaan pembelajaran sehingga meningkatkan efektivitas dan efisiensi proses pembelajaran. Melalui sistem ini, **admin** bisa **mengelola jurusan, kelas, mata pelajaran, jadwal, guru, siswa, dan informasi sekolah**. Pada tahap perencanaan **admin** bisa melakukan koordinasi dalam **menyusun** dan **memverifikasi** **dokumen administrasi guru**. Dalam tahap pengorganisasian **admin** dan **guru** bisa **mengelola** **dokumen** mulai dari **menyimpan, membagikan, dan mengatur** hak akses. Selanjutnya pada tahap pelaksanaan dan pengawasan, **admin** bisa **memantau** pelaksanaan **pembelajaran** melalui **catatan pengajaran guru** dan **presensi siswa**. **Guru** bisa **menambahkan** **informasi, materi ajar dan tugas** kelas, **mengisi** **catatan pengajaran** kelas dan **presensi siswa**, serta melakukan **penilaian**. Sedangkan **siswa** bisa **mengakses** **informasi pembelajaran** kelas dan sekolah, **mendaftar** ke dalam **mata pelajaran**, **mengunduh** **materi** dan **mengumpulkan** **tugas**, **melihat** **catatan belajar** individu dan **mencetak** **laporan hasil belajar** semester. Pada tahap akhir yaitu evaluasi, **admin** bisa **melihat** dan **mencetak** **laporan hasil rekapitulasi pembelajaran** setiap kelas mulai dari **catatan pengajaran guru, daftar hadir siswa, sampai laporan hasil belajar kelas**. Sedangkan **guru** bisa **melihat** **catatan belajar** setiap siswa dan **membuat** **laporan nilai akhir** mata pelajaran.

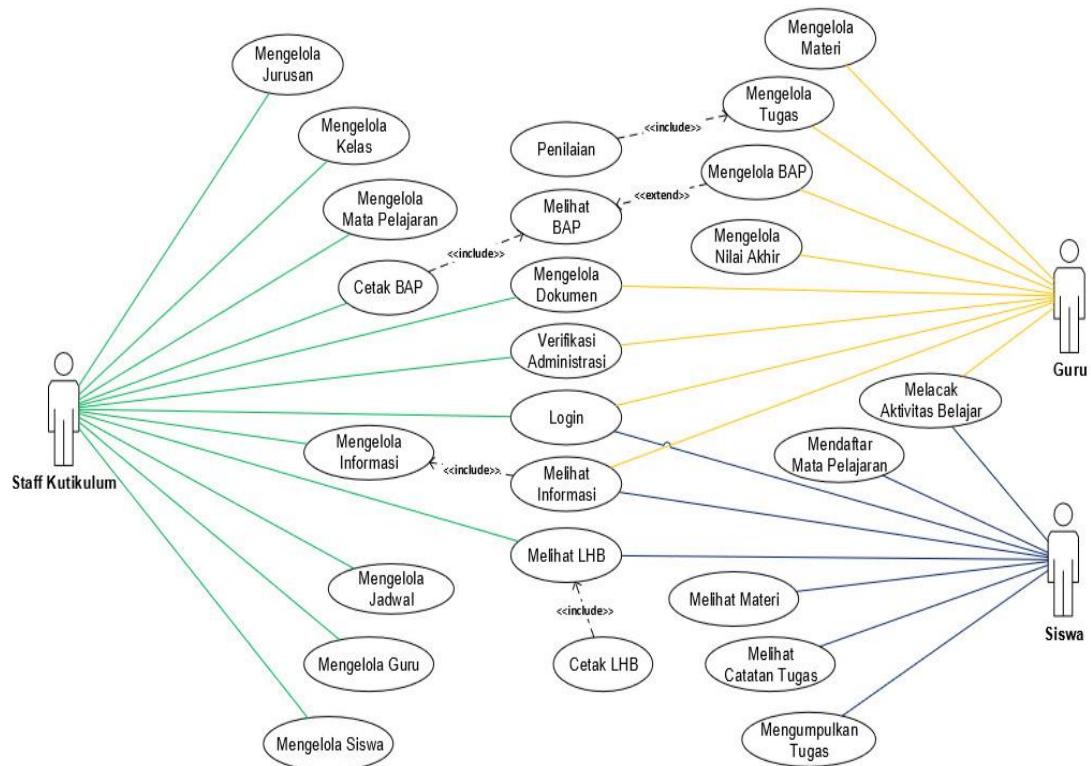
Fungsi tambahan dari sistem ini siswa bisa memberikan **polling** penilaian terhadap materi pengajaran guru.

Keterangan:

Biru = aktor, Kuning = kata kerja, Hijau = kata benda, Merah = masalah

b. *Use Case Diagram*

Pemodelan *use case diagram* mengacu pada kasus bisnis yang berada di dalam *automation scope* dan identifikasi aktor, aktivitas, objek yang telah dilakukan dalam pendeskripsian produk.



Gambar 14. *System Use Case Diagram*

Penjelasan dari aktor yang terlibat didalam sistem manajemen pembelajaran siswa bisa di lihat pada Tabel 16, sedangkan deskripsi *use case* dapat dilihat pada Tabel 17.

Tabel 16. Deskripsi Aktor *Use Cases*

No	Aktor	Deskripsi
1	Admin	Staf kurikulum dalam sistem manajemen pembelajaran siswa berperan sebagai admin. Admin mengendalikan mayoritas jalanya sistem melalui kewenangan/tugas meliputi: mengelola jurusan, kelas, mata pelajaran, guru, siswa, jadwal pengajaran, informasi, dokumen sekolah, memverifikasi dokumen administrasi guru, melihat dan mencetak laporan rekapitulasi kemajuan kelas, melihat dan mencetak laporan hasil belajar tiap kelas.
2	Guru	Guru dapat melihat informasi sekolah, mengelola dokumen pribadi, bimbingan menyusun administrasi pengajaran, mengelola aktivitas kelas meliputi: membuat topik pelajaran, menyediakan materi ajar ( <i>file</i> atau tautan), mengelola tugas, memberikan penilaian, melihat catatan belajar siswa, mengelola dan membuat laporan kemajuan kelas serta nilai akhir mata pelajaran.
3	Siswa	Siswa dapat melihat informasi sekolah, mendaftar mata pelajaran, melihat aktivitas kelas, mengunduh dan memberikan polling materi ajar, mengumpulkan tugas, melihat catatan tugas dan catatan belajar individu, serta melihat dan mencetak laporan hasil belajar semester.

Tabel 17. Deskripsi *Use Cases*

No	Use Case	Deskripsi
<b>Admin</b>		
1	Mengelola Jurusan	Mengelola data jurusan meliputi: membuat, melihat, mengubah dan menghapus data.
2	Mengelola Kelas	Mengelola data kelas meliputi: membuat, melihat, mengubah dan menghapus data.

No	Use Case	Deskripsi
3	Mengelola Mata Pelajaran	Mengelola data mata pelajaran meliputi: membuat, melihat, mengubah dan menghapus data.
4	Mengelola Guru	Mengelola data guru meliputi: membuat, melihat, mengubah dan menghapus data.
5	Mengelola Siswa	Mengelola data siswa meliputi: membuat, melihat, mengubah dan menghapus data.
6	Mengelola Jadwal Pelajaran	Mengelola data jadwal meliputi: membuat, melihat, mengubah dan menghapus data.
7	Mengelola Informasi Sekolah	Mengelola data informasi meliputi: membuat, melihat, mengubah dan menghapus data.
8	Verifikasi Administrasi Guru	Melihat berkas administrasi guru, memberikan penilaian, dan mengubah status berkas.
9	Mengelola Dokumen (Mengorganisasikan)	Mengelola direktori (menambah, melihat, mengubah, menghapus); Mengelola dokumen dalam direktori (menambah, melihat, mengubah, menghapus); Mengatur status dokumen (publik atau privat).
10	Melihat Berita Acara Pengajaran (BAP)	Melihat BAP setiap kelas, terdiri dari catatan pengajaran guru dan presensi kehadiran siswa.
11	Cetak BAP	Mencetak data BAP setiap kelas (.pdf)
12	Melihat Laporan Hasil Belajar (LHB)	Melihat LHB setiap kelas, berisikan nilai akhir dari mata pelajaran yang diambil oleh siswa.
13	Cetak LHB	Mencetak data LHB setiap kelas (.pdf)
<b>Guru Mata Pelajaran</b>		
14	Melihat informasi sekolah	Melihat informasi sekolah meliputi : kegiatan, agenda, dan pengumuman sekolah.

No	Use Case	Deskripsi
15	Verifikasi Administrasi	Menambah, melihat status dan penilaian, mengubah, dan menghapus data dokumen administrasi.
16	Mengelola Dokumen (Mengorganisasikan)	Mengelola direktori (menambah, melihat, mengubah, menghapus); Mengelola dokumen dalam direktori (menambah, melihat, mengubah, menghapus); Mengatur status dokumen.
17	Mengelola Materi Pelajaran	Mengelola data materi ajar meliputi: membuat, melihat, mengubah dan menghapus data.
18	Mengelola Tugas	Mengelola data tugas meliputi: membuat, melihat, mengubah dan menghapus data.
19	Penilaian	Melihat detail tugas siswa, menginputkan nilai, mengubah nilai, memberikan catatan belajar.
20	Melacak Aktivitas Belajar (Catatan belajar siswa)	Melihat riwayat aktivitas belajar siswa meliputi: catatan materi, tugas, kehadiran.
21	Mengelola Berita Acara Pengajaran	Mengelola data BAP meliputi: membuat, melihat, mengubah dan menghapus data.
22	Cetak BAP	Mencetak data BAP setiap kelas kedalam file .pdf
23	Mengelola Nilai Akhir	Mengelola data nilai akhir siswa meliputi: membuat, melihat dan mengubah data.
<b>Siswa</b>		
24	Melihat informasi sekolah	Melihat informasi sekolah meliputi : kegiatan, agenda, dan pengumuman sekolah.
25	Mendaftar Mata Pelajaran	Melakukan pendaftaran mata pelajaran ( <i>enroll</i> ) dengan memasukan kode.
26	Melihat Materi	Melihat materi ajar, mendownload file lampiran dan memberikan polling penilaian.

No	Use Case	Deskripsi
27	Melihat Catatan Tugas	Melihat daftar tugas kelas dari seluruh mata pelajaran yang ditempuh.
28	Mengumpul Tugas	Melakukan unggah file tugas ke dalam sistem
29	Melacak Aktivitas Belajar (Catatan hasil belajar)	Melihat catatan aktivitas belajar untuk setiap mata pelajaran yang diambil meliputi: catatan materi, tugas, kehadiran.
30	Melihat Laporan Hasil Belajar (LHB)	Melihat LHB setiap semester, berisikan nilai akhir dari mata pelajaran yang diambil oleh siswa.
31	Cetak LHB	Mencetak data LHB kedalam file .pdf

Aktivitas berikutnya setelah *use case* selesai dibuat adalah menjabarkanya dalam format skenario *use case*. *Use case* dapat dipahami sebagai deskripsi atau spesifikasi umum yang mendukung skenario yang berbeda. Setiap skenario adalah realisasi tertentu atau struktur proses dari *use case*. *Use case* memiliki skenario eksekusi *main flow* (aliran utama) dan *alternate* (alternatif). Skenario alternatif adalah eksekusi dari aliran utama yang melewati satu atau lebih aliran alternatif.

*Use case* pengelolaan data menggunakan pola standar CRUD (*Create, Read, Update, Delete*) sehingga memiliki skenario yang sama. *User case* tersebut meliputi mengelola jurusan, mengelola kelas, mengelola mata pelajaran, mengelola guru, mengelola siswa, mengelola jadwal, mengelola informasi, mengelola dokumen, mengelola materi dan mengelola tugas. Dengan alasan adanya kemiripan alur maka peneliti mengambil contoh skenario *use case* mengelola data siswa. Untuk lebih lengkapnya skenario *use case* dapat dilihat pada Lampiran 4.

## Skenario *use case* mengelola siswa

<b><i>Use case</i> mengelola siswa</b>		<b>Mengelola Siswa</b>
<b>Nama</b>	:	Mengelola Siswa
<b>Aktor</b>	:	Admin
<b>Deskripsi</b>	:	Mendeskripsikan proses yang digunakan untuk mengelola data siswa, meliputi standar CRUD: <i>Create, Retrieve, Update, Delete</i> data.
<b><i>Successful completion</i></b>		
1.	Admin memilih :	<ul style="list-style-type: none"><li>• <i>Create</i> (Menambah) : <i>Variant 1a, 1b</i></li><li>• <i>Retrieve</i> (Melihat) : <i>Variant 1c</i></li><li>• <i>Update</i> (Mengubah) : <i>Variant 1d</i></li><li>• <i>Delete</i> (Menghapus) : <i>Variant 1e</i></li></ul>
<b><i>Alternative</i></b>		
<i>Variant 1a : Create</i> siswa :		
1a.1	Admin memasukan data :	nis, nama, kelas id, jenis kelamin, telephone, tahun masuk, <i>username, password, email, user type</i> , foto, status.
<i>Variant 1b : Import</i> siswa dari file .xlsx/.csv :		
1b.1	Admin <i>upload</i> file data siswa dalam format .xlsx/.csv	
<i>Variant 1c : Retrieve</i> siswa :		
1c.1	Admin mengidentifikasi data siswa yang akan ditampilkan	
1c.2	Sistem menampilkan data siswa :	nis, nama, kelas, jenis kelamin, telephone, tahun masuk, <i>username, password, email, user type</i> , foto, status.
<i>Variant 1d : Update</i> siswa		
1d.1	Menggunakan <i>variant 1b: Retrieve</i> siswa	
1d.2	Admin memperbarui data siswa dengan data yang baru	

---

*Variant 1e: Delete siswa*

1e.1 Admin mengidentifikasi data siswa yang akan dihapus.

---

*Exception (1a.1, 1d.2)a:* Data NIS sudah terdaftar

(1a.1, 1d.2)a.1: Sistem memberitahu user bahwa data NIS yang dimasukan sudah terdaftar

(1a.1,1d.2)a.2: Kembali kelangkah yang menyebabkan *exception* terjadi.

---

*Exception (1a.1, 1d.2)b:* Masukan data tidak sesuai dengan ketentuan

(1a.1, 1d.2)b.1: Sistem memberitahu admin bahwa data yang dimasukan tidak sesuai dengan ketentuan yang diminta.

(1a.1, 1d.2)b.2: Admin mengubah masukan data yang salah sesuai dengan permintaan.

---

*Exception (1b.1)a* File *import* tidak valid

(1b.1)a.1: Sistem akan memberitahu bahwa struktur tabel tidak sesuai dengan ketentuan.

(1b.1)a.2: Kembali kelangkah yang menyebabkan *exception*.

---

**Precondition**

1. User admin teridentifikasi oleh sistem.
2. Admin memiliki *list/daftar* data siswa yang akan dikelola melalui pola CRUD.

---

**Postcondition**

1. Admin berhasil mengelola data siswa:
  - *Variant 1a:* Berhasil melakukan penambahan data siswa dan membuat *user account*
  - *Variant 1b:* Berhasil melakukan import data siswa melalui file .xlsx/.csv.
  - *Variant 1c:* Berhasil menampilkan data siswa
  - *Variant 1d:* Berhasil melakukan perubahan data siswa
  - *Variant 1e:* Berhasil menghapus data siswa dan *user account*

---

**Assumptions** : Tidak ada

## 2. ***Requirements***

Memperbaiki analisis kebutuhan yang telah didefinisikan sebelumnya dan menjelaskan lebih detail kebutuhan fungsionalitas sistem manajemen pembelajaran berdasarkan deskripsi produk dan *use case diagram*. Aktivitas ini dikerjakan supaya aktivitas pemodelan sistem dan pengkodean program terfokus pada domain permasalahan yang akan diselesaikan. Hasil dari proses identifikasi kebutuhan fungsional diantaranya:

- a. Kategori pengguna sistem terbagi menjadi tiga level yaitu staf kurikulum sebagai admin, guru dan siswa. Masing-masing level mempunyai hak akses dan wewenang (fungsi) yang berbeda.
- b. Menggunakan sistem *login* untuk melakukan autentikasi user sehingga setiap fungsi pada sistem dapat diakses sesuai dengan level user.
- c. Fungsi-fungsi untuk admin meliputi: mengelola data guru, mengelola data siswa, mengelola data mata pelajaran, mengelola data jadwal, mengelola data kelas, mengelola data jurusan, mengelola informasi sekolah, verifikasi administrasi guru, mengelola dokumen sekolah, melihat berita acara pengajaran (BAP), mencetak BAP, melihat laporan hasil belajar (LHB) siswa, mencetak LHB.
- d. Fungsi-fungsi untuk guru diantaranya: melihat informasi sekolah, verifikasi dokumen administrasi guru, mengorganisasikan dokumen, mengelola materi pelajaran, mengelola tugas kelas, menilai tugas, melacak aktivitas belajar

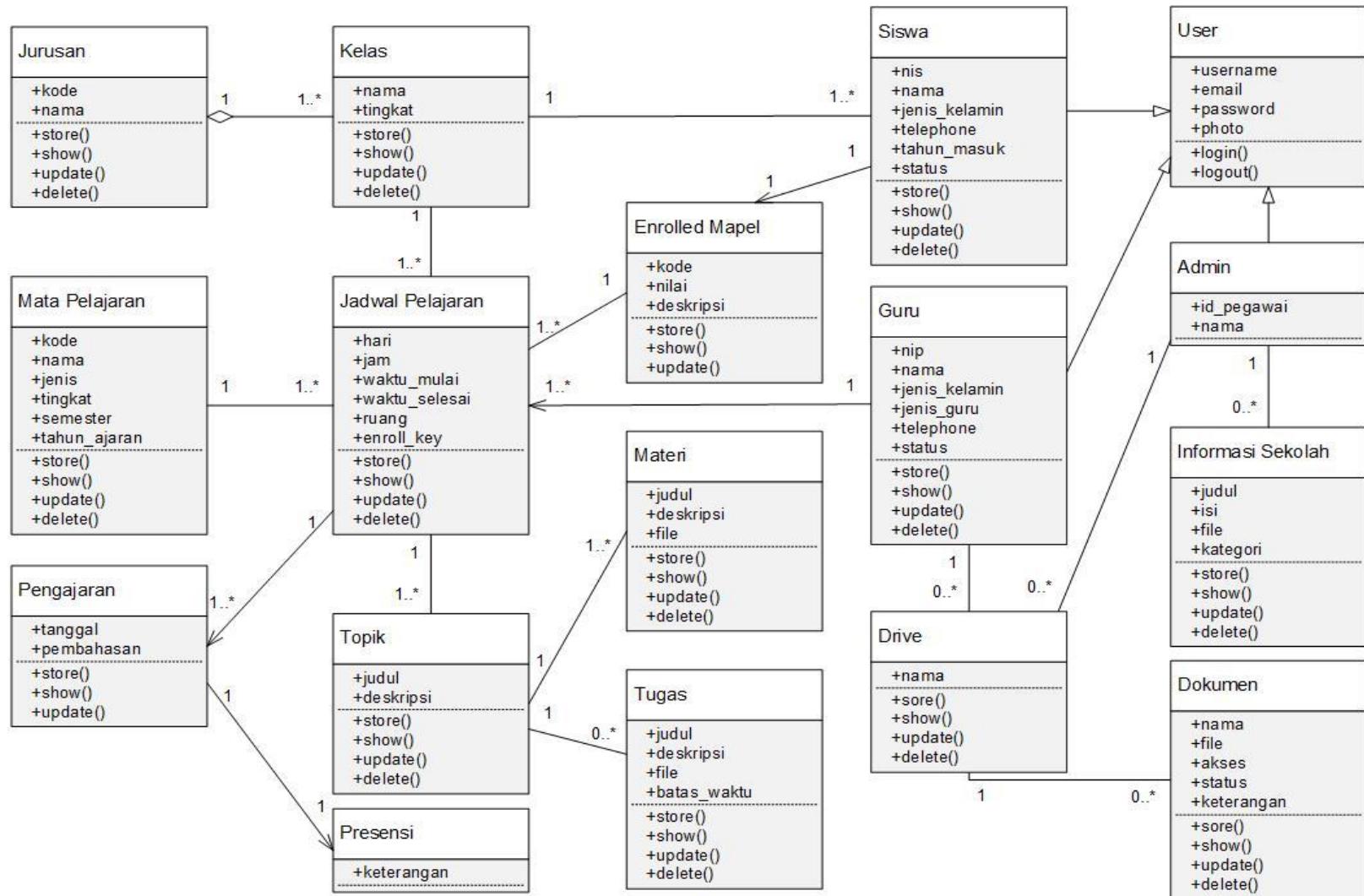
siswa, mengelola berita acara pengajaran (BAP), mencetak BAP, mengelola nilai akhir.

- e. Fungsi-fungsi untuk siswa diantaranya: melihat informasi sekolah, mendaftar mata pelajaran, melihat materi, melihat catatan tugas, mengumpulkan tugas, melihat catatan belajar individu, melihat laporan hasil belajar (LHB) semester, mencetak LHB.

### 3. *Analysis & Design*

#### a. *Class Diagram*

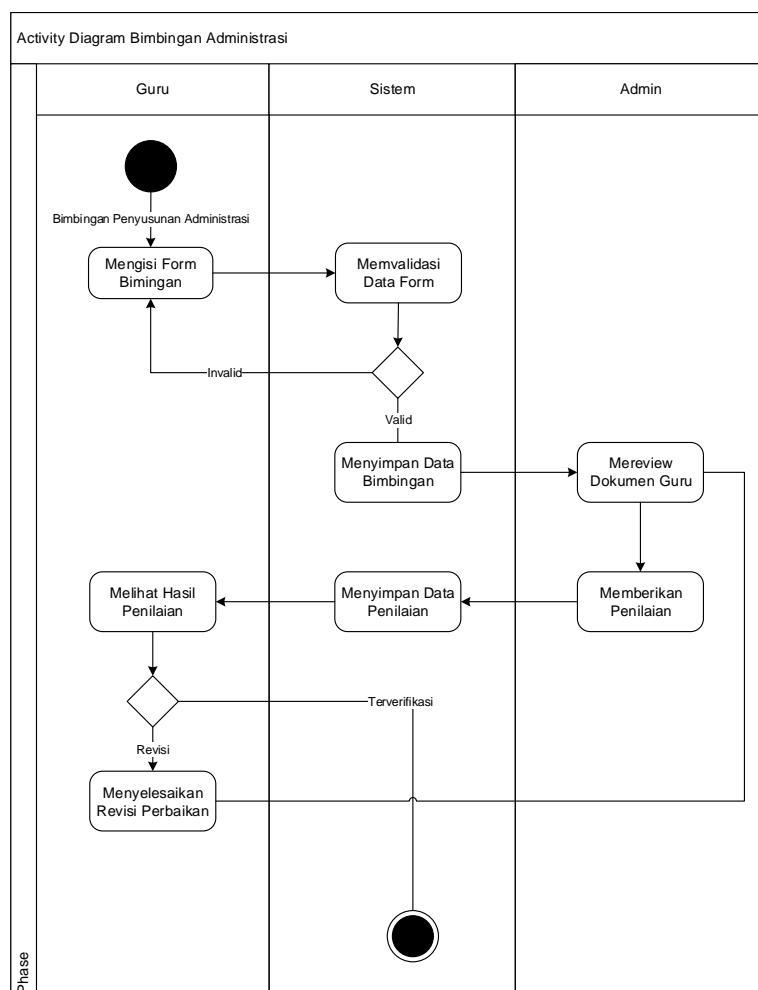
Pembuatan *class diagram* merujuk pada *use case* dan konsep pengkodean *framework* Laravel yang menggunakan standar operasi *Create, Read, Update, Delete* (CRUD). Langkah pembentukan dimulai dari melakukan pengidentifikasi objek-objek (*domain modelling*) pada kata benda yang terdapat pada daftar *requirement* yang diklasifikasikan pada area (domain) permasalahan yang sama untuk dijadikan *candidate class* pada *class diagram*. Selanjutnya menentukan *attribute* dan *method* yang merepresentasikan *class* tersebut. *Attribute* adalah karakteristik yang digunakan untuk mendeskripsikan sebuah entitas, sedangkan *method* mendefinisikan tugas spesifik yang dapat dikerjakannya. Setelah semua kelas berhasil didefinisikan, dilanjutkan dengan mendeskripsikan bagaimana interaksi hubungan antar class menggunakan konsep *cardinality*. Contoh hasil pemodelan dengan *class diagram* tersaji pada Gambar 15.



Gambar 15. *Class Diagram* Sistem Informasi Manajemen Pembelajaran

b. *Activity Diagram*

*Activity diagram* menunjukkan aktivitas yang terlibat dalam suatu proses atau dalam pemrosesan data. Proses memperlihatkan urutan tindakan yang dilakukan dan mengidentifikasi hasilnya. Memberikan representasi visual dari suatu alur interaksi dalam skenario tertentu sebagai pelengkap *use case*. Contoh hasil pemodelan dengan *activity diagram* tersaji pada Gambar 16.



Gambar 16. *Activity Diagram* Bimbingan Administrasi Guru

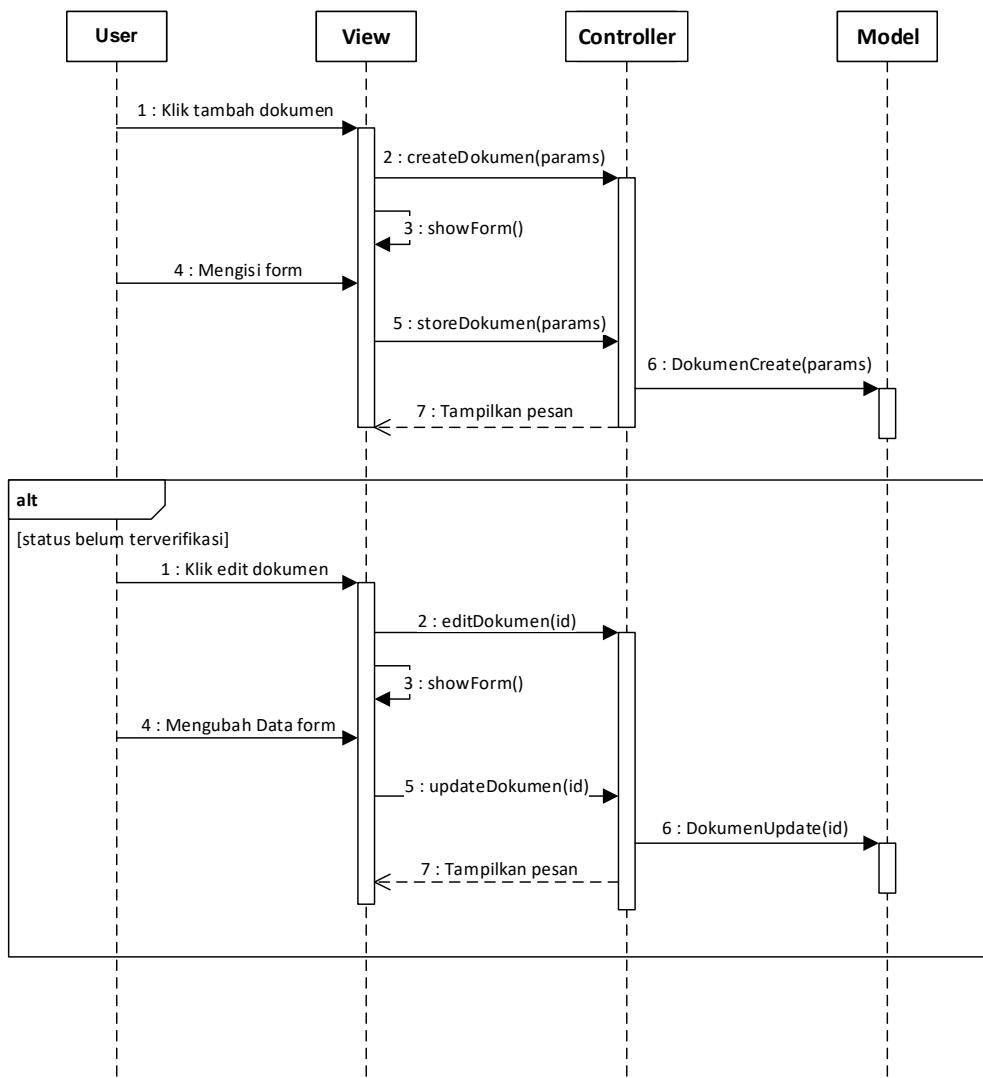
Desain *activity diagram* lebih lengkapnya dapat dilihat pada Lampiran 5. Pemodelan *activity diagram* sistem informasi manajemen pembelajaran menghasilkan 16 desain, dengan rincian seperti pada pada Tabel 18.

Tabel 18. Daftar *Activity Diagram*

<i>Activity Diagram</i>			
1	<i>Login dan logout</i>	9	Melacak aktivitas belajar
2	Standar CRUD	10	Membuat BAP
3	Verifikasi administrasi guru	11	Memasukan nilai akhir
4	Mengorganisasikan dokumen	12	Mendaftar mata pelajaran
5	Melihat rekapitulasi BAP	13	Mengerjakan aktivitas kelas
6	Melihat rekapitulasi LHB	14	Melihat catatan tugas
7	Membuat aktivitas kelas	15	Melihat catatan belajar
8	Penilaian tugas	16	Melihat LHB

c. *Sequence Diagram*

*Sequence diagram* menunjukan waktu transaksi antar objek saat terjadi selama eksekusi sistem. Interaksi berlangsung dari atas ke bawah di sepanjang garis waktu vertikal, sementara panah horizontal mewakili pesan dari satu objek ke objek lainnya. Contoh hasil pemodelan dengan *sequence diagram* verifikasi dokumen administrasi guru tersaji pada Gambar 17.



Gambar 17. *Sequence Diagram* Verifikasi Dokumen Administrasi Guru

Desain *sequence diagram* lebih lengkapnya dapat dilihat pada Lampiran 6.

Pemodelan *sequence diagram* sistem informasi manajemen pembelajaran menghasilkan 16 desain, dengan rincian seperti pada pada Tabel 19.

Tabel 19. Daftar *Sequence Diagram*

Sequence Diagram			
1	<i>Login</i> dan <i>logout</i>	9	Melacak aktivitas belajar
2	Standar CRUD	10	Membuat BAP
3	Verifikasi administrasi guru	11	Memasukan nilai akhir

4	Mengorganisasikan dokumen	12	Mendaftar mata pelajaran
5	Melihat rekapitulasi BAP	13	Mengerjakan aktivitas kelas
6	Melihat rekapitulasi LHB	14	Melihat catatan tugas
7	Membuat aktivitas kelas	15	Melihat catatan belajar
8	Penilaian tugas	16	Melihat LHB

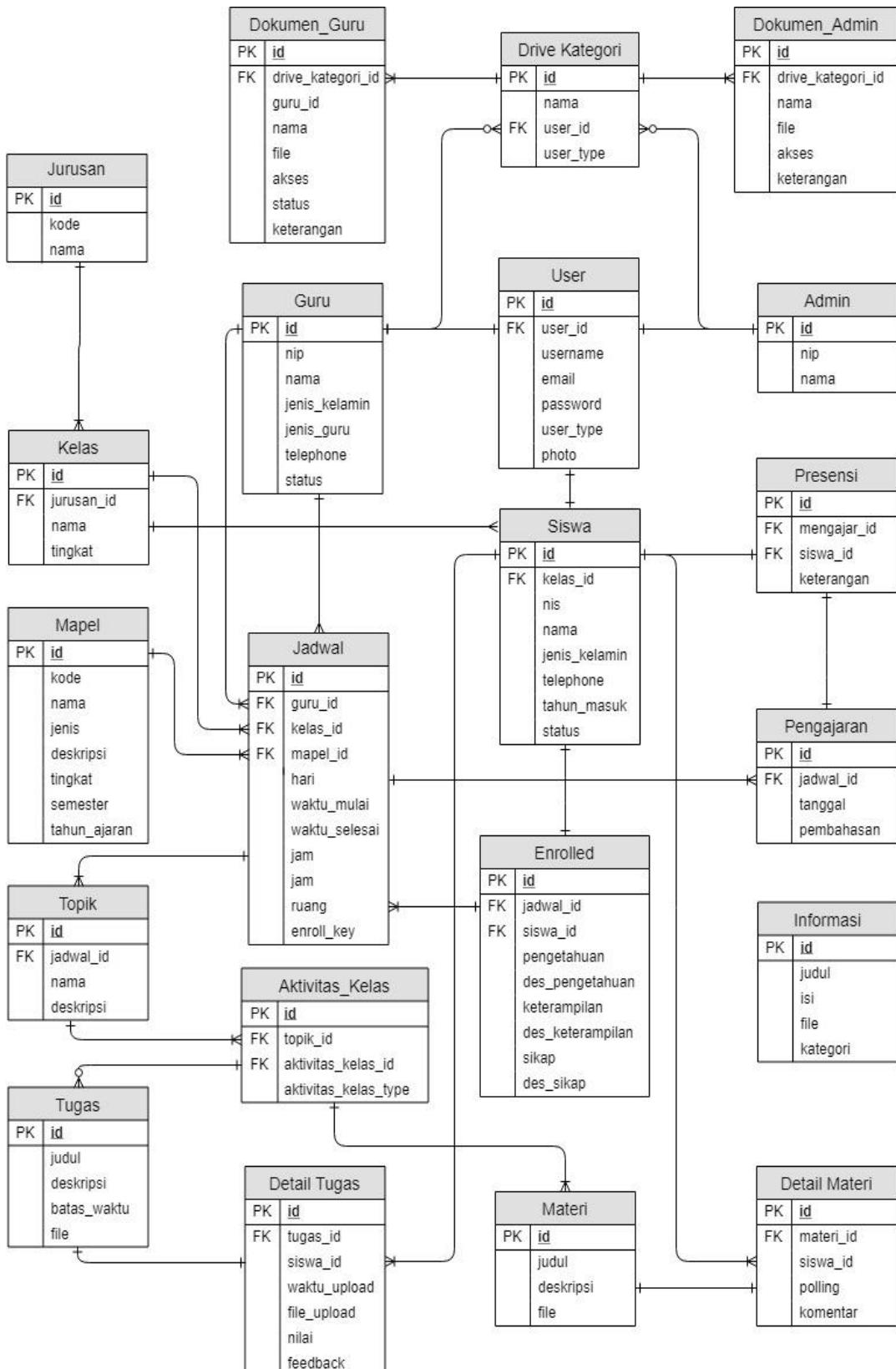
d. Desain *Database*

Pembuatan rancangan *database* bertujuan untuk menyusun struktur tabel dan menentukan hubungan/relasi antar tabel yang akan diimplementasikan dalam DBMS (*Database Management System*) menggunakan MySQL. Desain *database* merujuk pada desain *class diagram*. Pembuatan desain *database* menggunakan model ERD (*Entity Relationship Diagram*) yang menggambarkan hubungan antar entitas dalam basis data. Dalam ERD sistem informasi manajemen pembelajaran yang ditunjukkan Gambar 18 terdapat beberapa penambahan *attribute* dan tabel karena menyesuaikan dengan aturan implementasi relasi dalam *database*. Dalam setiap tabel terdapat *attribute* “id” digunakan sebagai *primary key*, “objek\_id” sebagai *foreign key*. Kamus data lebih lengkapnya bisa dilihat pada Lampiran 7. Sedangkan daftar tabel *database* dapat dilihat pada Tabel 20.

Tabel 20. Daftar Tabel *Database*

No	Nama Tabel	Deskripsi
1	User	Menyimpan data user/pengguna yang terdiri dari admin, guru dan siswa.
2	Admin	Menyimpan data deskripsi admin
3	Guru	Menyimpan data biodata guru
4	Siswa	Menyimpan data biodata siswa

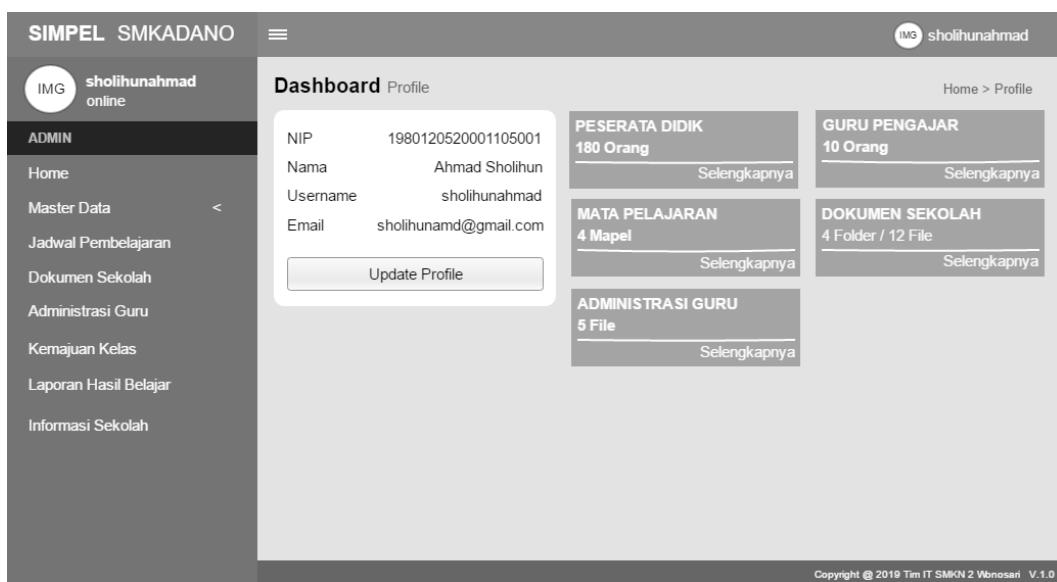
No	Nama Tabel	Deskripsi
5	Jurusan	Menyimpan data jurusan
6	Kelas	Menyimpan data kelas
7	Mata Pelajaran	Menyimpan data mata pelajaran
8	Jadwal Pelajaran	Menyimpan data jadwal pelajaran kelas
9	Info Sekolah	Menyimpan data informasi sekolah
10	Drive	Menyimpan data direktori
11	Dokumen Sekolah	Menyimpan data dokumen/file sekolah
12	Dokumen Guru	Menyimpan data dokumen/file guru
13	Enrolled Mapel	Menyimpan data mata pelajaran yang ditempuh setiap siswa
14	Topik	Menyimpan data topik pembelajaran tiap mapel
15	Aktivitas Kelas	Menyimpan data aktivitas kelas, menghubungkan tabel topik, tabel materi dan tabel tugas
16	Materi	Menyimpan data materi pelajaran
17	Detail Materi	Menyimpan <i>polling</i> penilaian dan tanggapan materi dari siswa
18	Tugas	Menyimpan data tugas pelajaran
19	Detail Tugas	Menyimpan data penilaian tugas siswa
20	Catatan Pengajaran	Menyimpan data catatan pengajaran dari setiap guru mata pelajaran
21	Presensi Siswa	Menyimpan data presensi siswa berdasarkan jam pembelajaran kelas



Gambar 18. ERD Sistem Informasi Manajemen Pembelajaran Siswa

#### 4. *Implementation*

Aktivitas yang dikerjakan pada tahap *implementation* adalah membuat *wireframe*/desain tampilan pengguna sistem dan pengimplementasian *database* yang telah dimodelkan menggunakan *Entity Relationship Diagram* (ERD). Kegiatan mendesain tampilan pengguna menggunakan *software* Pencil v.3.0.4, sedangkan implementasi ERD dilakukan dengan pembuatan tabel dan relasi antar tabel di MySQL. Hasil aktivitas desain tampilan digunakan untuk memberikan gambaran awal, jika terdapat ketidaksesuaian setelah dikomunikasikan ke pihak pengguna maka akan dilakukan perbaikan sesuai dengan permintaan. Contoh *wireframe* dari halaman *dashboard* admin tersaji pada Gambar 19.



Gambar 19. Desain Halaman *Dashboard* Admin

Desain tampilan pengguna lebih lengkapnya bisa dilihat pada Lampiran 8. Sistem informasi manajemen pembelajaran memiliki 27 halaman dengan rincian seperti pada Tabel 21.

Tabel 21. Daftar Desain Tampilan Pengguna

<b>Tampilan Pengguna</b>			
<b>No</b>	<b>Admin</b>	<b>Guru</b>	<b>Siswa</b>
1	Halaman Beranda	Halaman Beranda	Halaman Beranda
2	Halaman Jurusan	Halaman Drive	Halaman Mapel
3	Halaman Kelas	Halaman Dokumen	Halaman Stream Mapel
4	Halaman Mata Pelajaran	Halaman Mata Pelajaran	Halaman Catatan Tugas
5	Halaman Informasi Sekolah	Halaman Stream Mata Pelajaran	Halaman Catatan Belajar
6	Halaman Jadwal Pengajaran	Halaman Kelola Mata Pelajaran	Halaman Laporan Hasil Belajar (LHB)
7	Halaman Drive	Halaman Penilaian Tugas	
8	Halaman Dokumen Sekolah	Halaman Catatan Pengajaran (BAP)	
9	Halaman Verifikasi Dokumen	Halaman Catatan Belajar Siswa	
10	Halaman Catatan Pengajaran (BAP)	Halaman Nilai Akhir	
11	Halaman Laporan Hasil Belajar (LHB)		

Setelah aktivitas perancangan desain tampilan sistem selesai dilanjutkan dengan mengimplementasikan ERD. Contoh hasil implementasi ERD berupa tabel jadwal pelajaran tersaji pada Gambar 20.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
	1	id 	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
	2	mapel_id	int(10)		UNSIGNED	No	None		
	3	guru_id	int(10)		UNSIGNED	No	None		
	4	kelas_id	int(10)		UNSIGNED	No	None		
	5	hari	varchar(15)	utf8mb4_unicode_ci		No	None		
	6	jam	tinyint(4)			No	None		
	7	waktu_awal	time			No	None		
	8	waktu_akhir	time			No	None		
	9	ruang	varchar(30)	utf8mb4_unicode_ci		No	None		
	10	enroll_key	varchar(15)	utf8mb4_unicode_ci		No	None		
	11	created_at	timestamp			Yes	NULL		
	12	updated_at	timestamp			Yes	NULL		

Gambar 20. Implementasi Tabel Jadwal

## 5. *Test*

Aktivitas *test* belum dikerjakan karena fase *elaboration* fokus pada analisis dan desain arsitektur sistem dengan melakukan pembuatan *system use case*, *requirement model* menggunakan UML *activity diagram*, *sequence diagram*, *class diagram*, desain ERD, implementasi ERD, dan desain antarmuka.

## 6. *Deployment*

Aktivitas *deployment* belum dikerjakan karena fase elaboration fokus pada analisis dan desain arsitektur sistem dengan melakukan pembuatan *system use case*, *requirement model* menggunakan UML *activity diagram*, *sequence diagram*, *class diagram*, desain ERD, implementasi ERD, dan desain antarmuka.

## 7. *Configuration & Change Management*

Berkaitan dengan aktivitas manajemen dalam sebuah tim pengembang, peneliti tidak melakukan aktivitas proses ini karena proyek dikerjakan sendiri.

## 8. *Project Management*

Aktivitas pengembangan masih mengikuti *timeline* yang dibuat pada tahap *inception*.

## **9. *Environment***

Tidak perlu dikerjakan lagi, karena penyediaan keperluan pengembangan secara lengkap sudah dilakukan pada tahap *inception*.

### **C. *Tahap Construction***

Tahap *construction* memiliki fokus utama terhadap implementasi dan pengujian sistem perangkat lunak pada kode program. Perbaikan rancangan UML, ERD, pembuatan database dan desain tampilan pengguna masih dimungkinkan jika memang diperlukan. Aktivitas yang dikerjakan pada fase *construction* meliputi: menulis kode program dan menguji kode program untuk memastikan tidak ada kesalahan (*bug* atau *error*) ketika dijalankan.

#### **1. *Business Modeling***

Tidak mengerjakan aktivitas *business modeling* karena proses bisnis sudah teridentifikasi dengan baik, kebutuhan sistem dan ruang lingkup proyek yang akan dikembangkan sudah sesuai dengan permintaan.

#### **2. *Requirements***

Peninjauan ulang bersama beberapa guru IT terhadap *prototype* arsitektur sistem yang telah dibuat pada tahap *elaboration*, menghasilkan tambahan fitur sebagai berikut:

- a. Metode *import* data guru dan siswa menggunakan file format .xls/.xlsx.
- b. Laporan Hasil Belajar (LHB) untuk *user* admin/staf kurikulum dibuat dua jenis: LHB siswa kelas dan LHB mata pelajaran kelas.
- c. Pelaporan nilai akhir oleh guru dalam dua format .pdf dan .xls/.xlsx.

Ketiga fitur tersebut tetap harus diimplementasikan karena masih masuk kedalam lingkup fungsionalitas sistem yang dibutuhkan pengguna.

### 3. *Analysis & Design*

Adanya penambahan *requirements* sistem tidak mempengaruhi desain sistem secara keseluruhan sehingga tidak perlu dilakukan *modelling* ulang karena ketiga fitur tambahan masih masuk kedalam lingkup fungsionalitas sistem yang sudah didefinisikan. Peneliti hanya perlu menambahkan deskripsi ke dalam *system use case* seperti pada Tabel 22.

Tabel 22. Tambahan *Requirements* Sistem

No	Use Case	Deskripsi
<b>Admin</b>		
4	Mengelola Guru	Mengelola data guru: membuat, <b>meng-import</b> , melihat, mengubah dan menghapus data.
5	Mengelola Siswa	Mengelola data siswa: membuat, <b>meng-import</b> , melihat, mengubah dan menghapus data.
12	Melihat Laporan Hasil Belajar (LHB)	Melihat <b>LHB siswa</b> setiap kelas dan <b>LHB mata pelajaran</b> setiap kelas.
<b>Guru</b>		
22	Mencetak Nilai Akhir	Mencetak data nilai akhir mata pelajaran kedalam format <b>file .pdf dan .xls/.xlsx</b> .

### 4. *Implementation*

Implementasi kode program pada sistem informasi manajemen pembelajaran menggunakan *framework* Laravel 5.7 dengan *text editor* visual code. Penulisan kode program mengacu pada desain arsitektur sistem yang telah dibuat pada tahap analisis dan desain. *Framework* Laravel menggunakan

arsitektur MVC dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Berikut ini adalah potongan *source code* program yang terdiri dari *model*, *view*, *controller*, dan *routes*.

a. *Model*

*Model* adalah objek yang membentuk domain masalah, merepresentasikan struktur data dan fungsi-fungsi digunakan oleh sistem dalam pengelolaan basis data seperti *create*, *read*, *update*, *delete*.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Jadwal extends Model
{
    protected $table = 'jadwal';

    protected $fillable = [
        'mapel_id', 'guru_id', 'kelas_id', 'hari', 'jam',
        'waktu_awal', 'waktu_akhir', 'ruang', 'enroll_key'
    ];

    public function mapel(){
        return $this->belongsTo('App\Models\Mapel');
    }

    public function guru(){
        return $this->belongsTo('App\Models\Guru');
    }

    public function kelas(){
        return $this->belongsTo('App\Models\Kelas');
    }

    public function enrolled_mapel(){
        return $this->hasMany('App\Models\Enrolled_Mapel');
    }
    ....
```

Gambar 21. Penggalan Kode Program Model Jadwal Pelajaran

b. *View*

*View* adalah komponen grafis yang menyediakan kepada pengguna informasi yang terkandung dalam *model*. Ketika *model* mengalami perubahan, *view* harus diperbarui. Bagian yang mengatur tampilan ke pengguna.

```
@extends('master')

@section('content')
<section class="content-header">
  <h1>
    Dashboard
    <small>Jurusan</small>
  </h1>
  <ol class="breadcrumb">
    <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
    <li class="active">Jurusan</li>
  </ol>
</section>
<section class="content">
  <div class="row">
    <div class="col-xs-12">
      <div class="box">
        <div class="box-header">
          .....
        </div>
        <div class="box-body">
          .....
        </div>
        <div class="box-footer">
          .....
        </div>
      </div>
    </div>
  </section>
  @endsection
```

Gambar 22. Potongan Kode Program *View* Halaman Jurusan

c. *Controller*

*Controller* menerjemahkan tindakan pengguna (yaitu interaksi dengan tampilan) ke dalam operasi pada *model*. Berisi perintah-perintah yang berfungsi

untuk memproses suatu data dan mengirimkannya ke halaman *web*. Bagian yang menjembatani *model* dan *view*.

```
<?php
namespace App\Http\Controllers\Admin;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Kelas;
use App\Models\Guru;
use App\Models\Mapel;
use App\Models\Jadwal;
use DataTables;
use DB;

class JadwalController extends Controller {
    public function index() { return view("pages.jadwal");}
    public function dataTableJadwal(){
        $model = Jadwal::with('mapel','guru','kelas')->get();
        return DataTables::of($model)
            ->addColumn('action',function($model) {
                return view('partials._action',[

                    'model' => $model,
                    'url_show' => route('jadwal.show', $model->id),
                    'url_edit' => route('jadwal.edit', $model->id),
                    'url_destroy' => route('jadwal.destroy', $model->id)
                ]);
            })
            ->addIndexColumn()
            ->rawColumns(['action'])
            ->toJson();
    }
}
```

Gambar 23. Potongan Kode Program *Controller* Jadwal

d. *Routes*

*Routes* berfungsi menangani *request* kemudian mengarahkan aplikasi untuk memanggil halaman/resource tertentu. Semua permintaan dipetakan dengan bantuan *routes* ke *controller* terkait.

```

...
//ROUTE ADMIN
Route::group(['namespace' => "Admin", "prefix" => "admin",
'middleware' => ['auth', 'admin']], function(){
    Route::get('jadwal/table', 'JadwalController@dataTableJadwal')
->name('jadwal.table');

    Route::post('jadwal/{jadwal}', 'JadwalController@update')
->name('jadwal.update');

    Route::resource('jadwal', 'JadwalController', ['except'=>['update']]);
};

Route::get('drive', 'UserAdminController@driveKategori')
->name('drive.kategori');

Route::get('drive_kategori', 'UserAdminController@driveKategoriC
reate')->name('drive.kategori_create_admin');

Route::post('drive_kategori', 'UserAdminController@driveKategori
Store')->name('drive.kategori_store');

Route::get('drive_kategori/{id}/edit', 'UserAdminController@driv
eKategoriEdit')->name('drive.kategori_edit');

Route::post('drive_kategori/{id}', 'UserAdminController@driveKat
egoriUpdate')->name('drive.kategori_update');

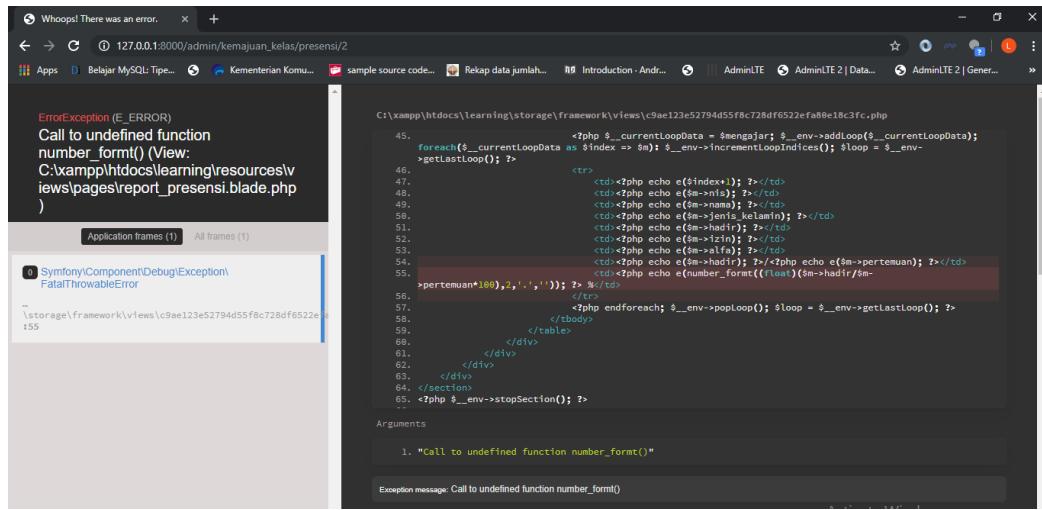
Route::delete('drive_kategori/{id}', 'UserAdminController@driveK
ategoriDestroy')->name('drive.kategori_destroy');
...

```

Gambar 24. Potongan Kode Program *Routes*

## 5. *Test*

Kegiatan *test* pada tahap ini dilakukan untuk mengecek penulisan kode program, fungsi program, dan tampilan yang sudah diimplementasikan. Peneliti memanfaatkan fitur *built-in* Laravel untuk mengetahui adanya kesalahan dalam kode program. Fitur *debug* Laravel menampilkan pesan kesalahan dan menunjuk baris kode yang bermasalah seperti yang terlihat pada Gambar 25.



```

Whoops! There was an error.

ErrorException (E_ERROR)
Call to undefined function
number_formt() (View: C:\xampp\htdocs\learning\resources\views\pages\report_presensi.blade.php)

```

```

C:\xampp\htdocs\learning\storage\framework\views\c9ae123e52794d55f8c728df6522ef8a88e18c3fc.php
45.     <?php $_currentLoopData = $mengajar; $_env->addLoop($_currentLoopData);
46.     foreach($_currentLoopData as $index => $m): $_env->incrementLoopIndices(); $loop = $_env-
47.     >getLastLoop(); ?> <tr>
48.         <td><?php echo e($m->nis); ?></td>
49.         <td><?php echo e($m->nama); ?></td>
50.         <td><?php echo e($m->jenis_kelamin); ?></td>
51.         <td><?php echo e($m->hadir); ?></td>
52.         <td><?php echo e($m->absen); ?></td>
53.         <td><?php echo e($m->absen_late); ?></td>
54.         <td><?php echo e($m->hadir); ?></td><?php echo e(number_formt((float)$m->hadir/$m-
55. >pertemuan*100),2,'.','.'); ?> </td>
56.     </tr>
57.     </tbody>
58.   </table>
59. </div>
60. </div>
61. </div>
62. </div>
63. </section>
64. <?php $_env->stopSection(); ?>
65.

```

Arguments

```

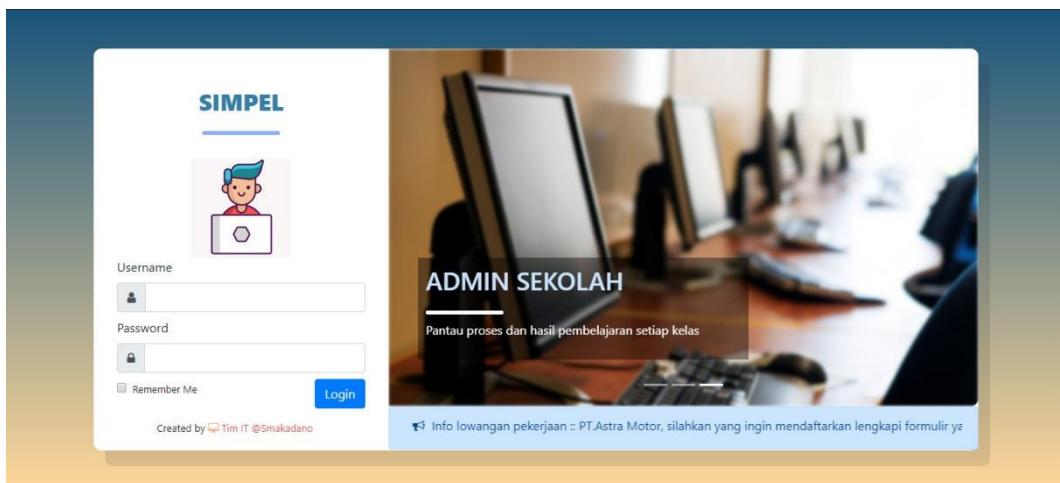
1. "Call to undefined function number_formt()"

```

Exception message: Call to undefined function number\_formt()

Gambar 25. Fitur *Debug Built-in* Laravel

*Web browser* digunakan sebagai alat bantu menjalankan program untuk mengetahui adakah kegagalan yang terjadi pada fungsi program (Gambar 26) dan tampilan halaman *web* (Gambar 27). Hasil pembuatan sistem lebih lengkapnya bisa dilihat pada Lampiran 9.



Gambar 26. Pengecekan Fungsi *Login*

No	NIS	Nama	Pengetahuan KI-3			Keterampilan KI-4			Sikap Spiritual dan Sosial KI-1 dan KI-2		
			Nilai	Angka	Predikat	Nilai	Angka	Predikat	Nilai	Angka	Predikat
1	1139	Almira Shifa A	84	3,66	A-	89	4	A	80	4	B
2	15551	Adelina Listiana Ma'rifah	0	1	D	0	1	D	0	1	K
3	15552	Aditya Wisnu Ralhan	0	1	D	0	1	D	0	1	K
4	15553	Adnan Fathoni	0	1	D	0	1	D	0	1	K
5	15554	Afrizal Febrianto	0	1	D	0	1	D	0	1	K
6	15555	Ahmad Firmansyah	n	1	n	n	1	n	n	1	K

Gambar 27. Pengecekan Tampilan Rekapitulasi Nilai Akhir

## 6. Deployment

Aktivitas *deployment* belum dikerjakan karena fase *construction* lebih fokus pada analisis pada menulis dan menguji kode program agar tidak terjadi kesalahan *bug* atau *error* ketika dijalankan.

## 7. Configuration & Change Management

Berkaitan dengan aktivitas manajemen dalam sebuah tim pengembang, peneliti tidak melakukan aktivitas proses ini karena proyek dikerjakan sendiri.

## 8. Project Management

Aktivitas pengembangan masih mengikuti *timeline* yang dibuat pada tahap *inception*.

## 9. Environment

Tidak perlu dikerjakan lagi, karena penyediaan keperluan pengembangan secara lengkap sudah dilakukan pada tahap *inception*.

## **D. Tahap *Transition***

Tahap *transition* memiliki fokus utama melakukan *deployment* atau pemindahan sistem dari lingkungan pengembang ke lingkungan operasional. Aktivitas yang dikerjakan diantaranya: melakukan instalasi sistem, sosialisasi sistem pada pengguna, pelatihan pengguna, pemeliharaan dan pengujian sistem agar harapan *user* terpenuhi.

### **1. *Business Modeling***

Tidak mengerjakan aktivitas *business modeling* karena proses bisnis sudah teridentifikasi dengan baik, kebutuhan sistem dan ruang lingkup proyek yang akan dikembangkan sudah sesuai dengan permintaan.

### **2. *Requirements***

Tidak mengerjakan aktivitas *requirements* karena analisis kebutuhan fungsional sistem informasi manajemen pembelajaran siswa sudah sesuai dengan kebutuhan pengguna.

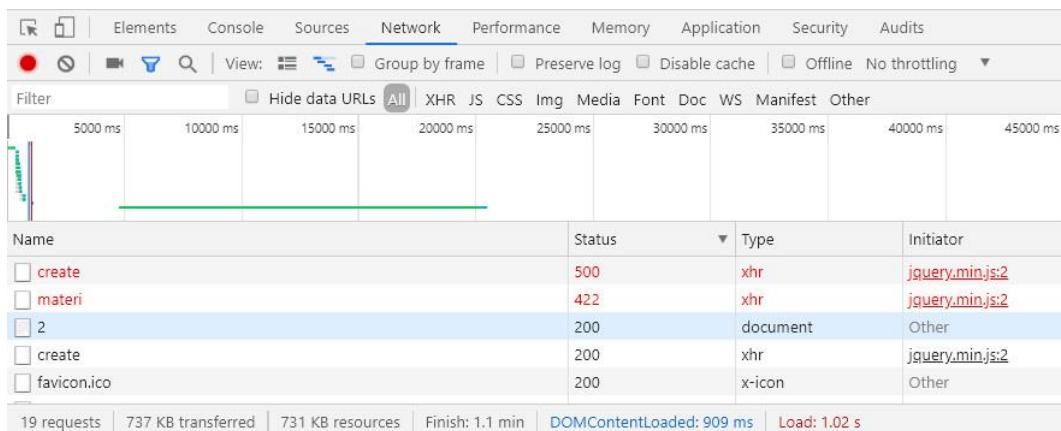
### **3. *Analysis & Design***

Tidak mengerjakan aktivitas *analysis & design* karena pemodelan *use case diagram, activity diagram, sequence diagram, class diagram, Entity Relationship Diagram* (ERD) dan implementasi ERD, desain antarmuka sudah sesuai dengan kebutuhan.

### **4. *Implementation***

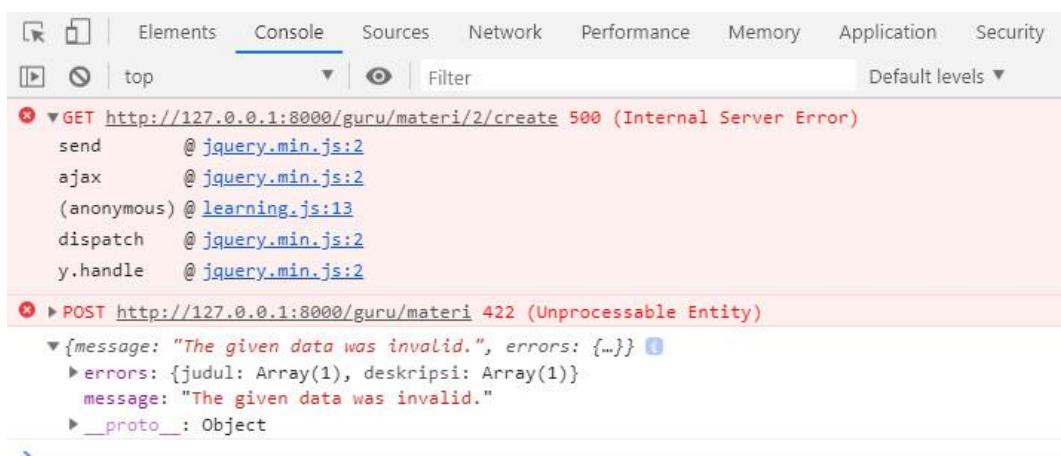
Aktivitas implementasi pada fase *transition* meliputi optimasi kode program yang telah dibuat, melakukan serangkaian uji coba fitur dengan beberapa kemungkinan kondisi, menambahkan beberapa *exception* untuk menangani *request*

pengguna yang tidak sesuai dengan ketentuan sistem. Optimasi bisa dilakukan menggunakan beberapa cara, berikut ini salah satu contoh hasil analitik yang bisa digunakan sebagai acuan dalam mengoptimalkan sistem.



Gambar 28. Optimasi Menggunakan *Network Activity* DevTools

Gambar 28 menunjukkan *network response* dari *request* pengguna terhadap sistem yang menampilkan waktu respon, assets, dan berbagai komponen yang ada pada halaman *web* tersebut. Memudahkan pengembang dalam meningkatkan *load time* sistem ketika diakses.



Gambar 29. Optimasi Menggunakan *Console* DevTools

Gambar 29 menunjukkan *log* kesalahan sistem ketika pengguna sedang melakukan suatu *request*. Tampilnya status dan pesan kesalahan memudahkan pengembangan

melakukan *troubleshooting* ketika sedang melakukan pengujian fitur dengan berbagai alternatif kemungkinan.

## 5. *Test*

Sistem diuji menggunakan *software quality model* ISO/IEC 25010 meliputi karakteristik *functional suitability, usability, reliability, performance efficiency*, dan *maintainability*. Pengujian perangkat lunak bertujuan untuk mengetahui dan memastikan kualitas sistem informasi manajemen pembelajaran telah memenuhi standar perangkat lunak.

### a. *Functional Suitability*

Karakteristik *functional suitability* diuji oleh tiga ahli dengan latar belakang keahlian yang berbeda. Ketiga ahli adalah *web developer* dari berbagai perusahaan pengembang perangkat lunak yang ada di Yogyakarta. Tabel 23 menunjukkan daftar penguji sistem informasi manajemen pembelajaran siswa. Sedangkan salah satu skenario pengujian menggunakan metode *test case* bisa dilihat pada Gambar 30.

Tabel 23. Daftar Ahli Pengujian *Functional Suitability*

No	Nama	Profesi	Instansi
1	Anjasmoro Adi Nugroho	<i>Backend Developer</i>	PT. Privatku Global Indo
2	Bintang Muhammad	<i>Frontend Developer</i>	PT. Yeloo Integra Datanet
3	Deni Kurnianto Nugroho	<i>Fullstack Developer</i>	PT. Privatku Global Indo

Test Case ID	GM_007	Test Case Description	Uji fungsi mengisi Berita Acara Pengajaran (BAP) kelas																														
Created By	Luthfi	Reviewed By	Anjas																														
<b>QA Tester's Log</b> Fungsi bekerja dengan baik sesuai scenario, hasil sesuai harapan																																	
<table border="1"> <tr> <td>Tester's Name</td> <td>Anjas</td> <td>Date Tested</td> <td>April 18, 2019</td> </tr> <tr> <td>S #</td> <td>Prerequisites:</td> <td>S #</td> <td>Test Data Requirement</td> </tr> <tr> <td>1</td> <td>Login sebagai guru</td> <td>1</td> <td>Tanggal: 18 April 2019</td> </tr> <tr> <td>2</td> <td></td> <td>2</td> <td>Materi Pelajaran: Pengantar Algoritma</td> </tr> <tr> <td>3</td> <td></td> <td>3</td> <td>Kehadiran: Hadir (25 siswa), Izin (3 siswa), Tidak</td> </tr> <tr> <td>4</td> <td></td> <td>4</td> <td></td> </tr> </table>				Tester's Name	Anjas	Date Tested	April 18, 2019	S #	Prerequisites:	S #	Test Data Requirement	1	Login sebagai guru	1	Tanggal: 18 April 2019	2		2	Materi Pelajaran: Pengantar Algoritma	3		3	Kehadiran: Hadir (25 siswa), Izin (3 siswa), Tidak	4		4							
Tester's Name	Anjas	Date Tested	April 18, 2019																														
S #	Prerequisites:	S #	Test Data Requirement																														
1	Login sebagai guru	1	Tanggal: 18 April 2019																														
2		2	Materi Pelajaran: Pengantar Algoritma																														
3		3	Kehadiran: Hadir (25 siswa), Izin (3 siswa), Tidak																														
4		4																															
<b>Test Scenario</b> Memasukan data BAP kelas, aktivitas pengajaran guru tercatat																																	
<table border="1"> <thead> <tr> <th>Step #</th> <th>Step Details</th> <th>Expected Results</th> <th>Actual Results</th> <th>Pass / Fail / Not executed / Suspended</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pilih menu BAP</td> <td>Tampil daftar mata pelajaran yang diampu</td> <td>As Expected</td> <td>Pass</td> </tr> <tr> <td>2</td> <td>Pilih masukan BAP</td> <td>Tampil daftar siswa mata pelajaran terkait</td> <td>As Expected</td> <td>Pass</td> </tr> <tr> <td>3</td> <td>Masukan test data requirement</td> <td>Test data requirement bisa dimasukan</td> <td>As Expected</td> <td>Pass</td> </tr> <tr> <td>4</td> <td>Klik tombol simpan</td> <td>Berhasil menambah data, redirect ke halaman catatan</td> <td>As Expected</td> <td>Pass</td> </tr> <tr> <td>5</td> <td>Klik tombol detail (sesuai tanggal)</td> <td>Tampil data BAP sesuai tanggal pengajaran guru</td> <td>As Expected</td> <td>Pass</td> </tr> </tbody> </table>				Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	1	Pilih menu BAP	Tampil daftar mata pelajaran yang diampu	As Expected	Pass	2	Pilih masukan BAP	Tampil daftar siswa mata pelajaran terkait	As Expected	Pass	3	Masukan test data requirement	Test data requirement bisa dimasukan	As Expected	Pass	4	Klik tombol simpan	Berhasil menambah data, redirect ke halaman catatan	As Expected	Pass	5	Klik tombol detail (sesuai tanggal)	Tampil data BAP sesuai tanggal pengajaran guru	As Expected	Pass
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended																													
1	Pilih menu BAP	Tampil daftar mata pelajaran yang diampu	As Expected	Pass																													
2	Pilih masukan BAP	Tampil daftar siswa mata pelajaran terkait	As Expected	Pass																													
3	Masukan test data requirement	Test data requirement bisa dimasukan	As Expected	Pass																													
4	Klik tombol simpan	Berhasil menambah data, redirect ke halaman catatan	As Expected	Pass																													
5	Klik tombol detail (sesuai tanggal)	Tampil data BAP sesuai tanggal pengajaran guru	As Expected	Pass																													

Gambar 30. *Test Case* Mengisi BAP Kelas

Terdapat tiga subkarakteristik *functional suitability* yang diuji dengan instrumen pengujian yaitu *functional completeness*, *functional correctness*, dan *functional appropriateness*. Rumus untuk menghitung nilai pengujian adalah:

$$X = \frac{I}{P}$$

Dengan:

I = Jumlah fitur yang berhasil diterapkan

P = Jumlah fitur yang dirancang

Perhitungan menggunakan data dari instrumen pengujian yang telah diisi oleh responden. Hasil pengambilan data *functional suitability* dapat dilihat pada Lampiran 10. Kemudian pengujian ketiga subkarakteristik *functional suitability* dijelaskan sebagai berikut:

### 1) *Functional Completeness*

Peneliti merangkum hasil pengujian *functional completeness* kedalam Tabel 24. Pada tabel tersebut tanggapan dari ketiga penguji diakumulasikan. Sebagai contoh pada nomor pernyataan 1-15 yang merujuk pada user admin memperoleh jawaban “Ya atau 1” sebanyak  $3 \times 15 = 45$ . Angka 3 menunjukkan jumlah penguji, sedangkan angka 15 menunjukkan jumlah butir pernyataan yang mendapat nilai 1.

Tabel 24. Hasil Pengujian *Functional Completeness*

No Pertanyaan	Ya	Tidak
User Admin 1 – 15	$3 \times 15 = 45$	0
User Guru 16 – 29	$3 \times 14 = 42$	0
User Siswa 30 – 38	$3 \times 9 = 27$	0
<b>Total Nilai (I)</b>	<b><math>45 + 42 + 27 = 144</math></b>	<b>0</b>

Perhitungan hasil pengujian subkarakteristik *functional completeness*:

$$\begin{aligned} P &= \text{jumlah pernyataan} \times \text{jumlah penguji} \\ &= 38 \times 3 \\ &= \mathbf{144} \end{aligned}$$

Sehingga:

$$X = \frac{144}{144} = 1$$

### 2) *Functional Correctness*

Peneliti merangkum hasil pengujian *functional correctness* kedalam Tabel 25. Cara mengolah data hasil pengujian dari ahli sama seperti yang dijelaskan dalam *functional completeness*.

Tabel 25. Hasil Pengujian *Functional Correctness*

No Pertanyaan	Ya	Tidak
Umum 1	$3 \times 1 = 3$	<b>0</b>
User Admin 2 – 16	$3 \times 15 = 45$	0
User Guru 17-27	$3 \times 11 = 33$	0
User Siswa 28 – 32	$3 \times 5 = 15$	0
<b>Total Nilai (I)</b>	<b><math>3 + 45 + 33 + 15 = 96</math></b>	<b>0</b>

Perhitungan hasil pengujian subkarakteristik *functional correctness*:

$$\begin{aligned}
 P &= \text{jumlah peryataan} \times \text{jumlah penguji} \\
 &= 32 \times 3 \\
 &= \mathbf{96}
 \end{aligned}$$

Sehingga:

$$X = \frac{96}{96} = 1$$

### 3) *Functional Appropriateness*

Peneliti merangkum hasil pengujian *functional appropriateness* kedalam Tabel 26. Cara mengolah data hasil pengujian dari ahli sama seperti yang dijelaskan dalam *functional completeness*.

Tabel 26. Hasil Pengujian *Functional Appropriateness*

No Pertanyaan	Ya	Tidak
User Admin 1 – 6	$3 \times 6 = 18$	0
User Guru 7 – 12	$3 \times 6 = 18$	0
User Siswa 13 – 16	$3 \times 4 = 12$	0
<b>Total Nilai (I)</b>	<b><math>18 + 18 + 12 = 48</math></b>	<b>0</b>

Perhitungan hasil pengujian subkarakteristik *functional appropriateness*:

$$\begin{aligned} P &= \text{jumlah peryataan} \times \text{jumlah penguji} \\ &= 16 \times 3 \\ &= \mathbf{48} \end{aligned}$$

Sehingga:

$$X = \frac{48}{48} = 1$$

Hasil pengujian menunjukan bahwa semua fitur berhasil diterapkan, berjalan dengan benar sesuai dengan rancangan awal sehingga sistem memenuhi syarat *functional suitability*.

#### b. *Usability*

Karakteristik *usability* pada sistem manajemen pembelajaran siswa diuji dengan USE *Questionnaire*. Kuesioner menggunakan skala likert 5 poin dengan jumlah pertanyaan sebanyak 30 butir. Kuesioner tersebut diisi oleh 30 responden terdiri dari staf kurikulum, guru dan siswa. Hasil pengujian *usability* ditunjukan pada Lampiran 11.

Mengacu pada data hasil pengujian, sebaran jawaban responden bisa dikelompokan menjadi total jawaban sangat tidak setuju (STS) berjumlah **0**, tidak setuju (TS) berjumlah **2**, ragu-ragu (RG) berjumlah **87**, setuju (S) berjumlah **397**, dan sangat setuju (SS) berjumlah **414**. Hasil terebut kemudian dihitung menggunakan rumus:

$$\text{Skor total} = (414 \times 5) + (397 \times 4) + (87 \times 3) + (2 \times 2) + (0 \times 1)$$

$$P \text{ skor} = \frac{3923}{4500} \times 100\% = 87,2\%$$

Persentase pengujian *usability* dari hasil perhitungan sebesar **87,2%**. Kemudian skor tersebut diinterpretasikan berdasarkan ketentuan yang terdapat pada Tabel 8, sehingga bisa ditarik kesimpulan jika sistem yang dikembangkan **Sangat Layak** dari sisi *usability* dan memenuhi standar kriteria. Untuk mengukur nilai konsistensi dari hasil pengujian *usability*, peneliti melakukan perhitungan *alpha cronbach* dengan bantuan *software* SPSS. Gambar 31 menunjukkan hasil perhitungan nilai konsistensi.

### Scale: ALL VARIABLES

#### Case Processing Summary

		N	%
Cases	Valid	30	100,0
	Excluded <sup>a</sup>	0	,0
	Total	30	100,0

a. Listwise deletion based on all variables in the procedure.

#### Reliability Statistics

Cronbach's Alpha	N of Items
,955	30

Gambar 31. Perhitungan Nilai *Alpha CronBach*

Gambar 31 menunjukkan bahwa jumlah item yang diuji sebanyak 30 (jumlah butir pernyataan dalam instrumen) dengan nilai Cronbach's Alpha **0,955**. Jika diterjemahkan berdasarkan Tabel 9 maka instrumen penelitian termasuk kategori **excellent**. Karena nilai Cronbach's Alpha 0,955 melebihi batas standar yaitu 0,60 dan melebihi nilai r tabel 0,361 (dengan signifikansi 5%) maka bisa disimpulkan bahwa keseluruhan item pernyataan dalam instrumen *usability* adalah **reliabel** atau **konsisten**.

### c. *Reliability*

Pengujian *reliability* sistem manajemen pembelajaran siswa dilakukan dengan memakai *software* WAPT Pro v.5.0. Skenario pengujian sistem menggunakan metode *stress testing* dengan hasil pengujian seperti yang ditunjukan Gambar 32.

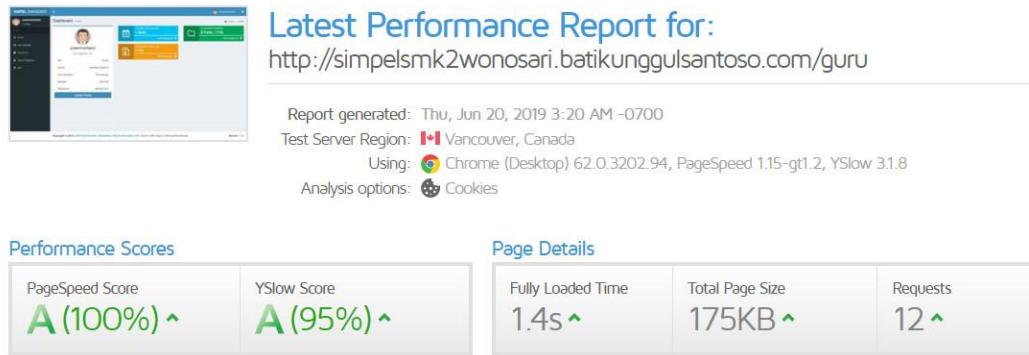
<b>Test execution parameters:</b>									
Test status:	finished								
Test started at:	02/06/2019 19:50:23								
Scenario name:									
Test run comment:									
Test executed by:	X550Z FX (ASUS)								
Test executed on:	ASUS								
Test duration:	0:10:00								
<b>Test result: <span style="color: green;">SUCCESS</span></b>									
<b>Pass/Fail Criteria</b>									
Name	Result								
Session error rate for each profile	SUCCESS								
<b>Summary</b>									
Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes sent	Total KBytes received
Profile1	6292	0	18876	0	122694	0	0	48673	1743529

Gambar 32. Hasil Pengujian *Reliability*

Berdasarkan *report* dari *software* WAPT Pro v5.0, diperoleh *summary* hasil pengujian berupa *successful session* 6292, *failed session* 0, *successful pages* 18876, *failed pages* 0 , *successful hits* 122694, dan *failed hits* 0. Data tersebut menunjukan bahwa selama pengujian *stress testing* tidak ada kesalahan/kegagalan yang terjadi.

### d. *Performance Efficiency*

Karakteristik *performance efficiency* sistem informasi manajemen pembelajaran siswa diuji dengan *software* GTMetrix. Gambar 33 menunjukan salah satu contoh hasil pengujian GTMetrix pada halaman *dashboard* guru.



Gambar 33. Hasil Pengujian *Performance Efficiency* Halaman *Dashboard Guru*  
 Keseluruhan hasil pengujian *performance efficiency* dari setiap halaman *web* terdapat di Lampiran 12. Sedangkan ringkasan hasil pengujian *performance efficiency* disajikan pada Tabel 27.

Tabel 27. Hasil Pengujian *Performance Efficiency*

No	Halaman	Hasil Pengujian		
		Page Speed	Yslow	Waktu
1	<i>Login</i>	99	95	1,9
<b>Admin</b>				
2	<i>Dashboard Admin</i>	100	95	1.4
3	Kelola Jurusan	100	93	1.6
4	Kelola Kelas	100	94	1.6
5	Kelola Mata Pelajaran	99	93	1.6
6	Kelola Guru	99	92	1.5
7	Kelola Siswa	99	91	1.6
8	Kelola <i>User Admin</i>	100	92	1.6
9	Jadwal Pembelajaran	99	93	1.5
10	Direktori Sekolah	100	93	1.6
11	Daftar File Direktori	100	92	1.6
12	Administrasi Guru	99	92	1.5
13	Kemajuan Kelas	100	94	1.5

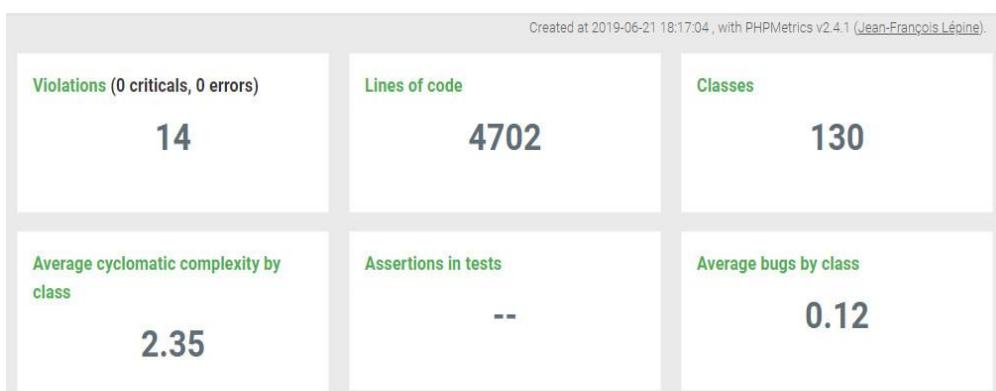
No	Halaman	Hasil Pengujian		
		Page Speed	Yslow	Waktu
14	Mata Pelajaran Kelas	100	95	1.4
15	Rekapitulasi Pengajaran	100	95	1.4
16	Rekapitulasi Daftar Hadir	100	95	1.4
17	Laporan Hasil Belajar Kelas	100	94	1.4
18	<i>Create/Edit</i> Informasi	100	95	1.5
19	<i>List</i> Informasi	100	94	1.4
20	Detail Informasi	100	95	1.3
	<b>Guru</b>			
21	<i>Dashboard</i> Guru	100	95	1.4
22	<i>List</i> Informasi	100	94	1.4
23	Detail Informasi	100	95	1.4
24	Direktori Guru	99	93	1.6
25	Daftar File Direktori	99	93	1.6
26	Daftar Pengajaran	100	94	1.5
27	Aktivitas Belajar Kelas	100	93	1.6
28	Kelola Aktivitas Belajar	100	94	1.7
29	Materi Pelajaran	100	95	1.4
30	Tugas Kelas	99	94	1.5
31	Catatan Belajar Siswa	100	95	1.4
32	Berita Acara Pengajaran	100	95	1.4
33	Presensi Siswa	100	95	1.4
34	Laporan Pengajaran	99	94	1.5
35	Laporan Daftar Hadir Siswa	100	95	1.4
36	Memasukan Nilai Akhir	100	95	1.4
37	Laporan Nilai Akhir	100	95	1.4
	<b>Siswa</b>			
38	<i>Dashboard</i> Siswa	100	95	1.4
39	<i>List</i> Informasi Sekolah	100	94	1.4

No	Halaman	Hasil Pengujian		
		Page Speed	Yslow	Waktu
40	Detail Informasi Sekolah	100	95	1.4
41	Catatan Tugas Kelas	100	95	1.4
42	Daftar Pelajaran	100	93	1.6
43	Aktivitas Belajar Kelas	99	92	1.6
44	Materi Pelajaran	99	93	1.5
45	Tugas Mata Pelajaran	99	93	1.6
46	Catatan Belajar	100	95	1.5
47	Laporan Hasil Belajar	100	95	1.4
<b>Rata-rata</b>		<b>99,72%</b>	<b>93,96%</b>	<b>1,49</b>

Pengujian *performance efficiency* menghasilkan nilai rata-rata *page speed* sebesar **99,72% (Grade A)**, *Yslow* sebesar **93,96% (Grade A)** , dan *load time* sebesar **1,49 detik**. *Load time* kurang dari 10 detik, menandakan bahwa sistem memenuhi standar kriteria.

#### e. *Maintainability*

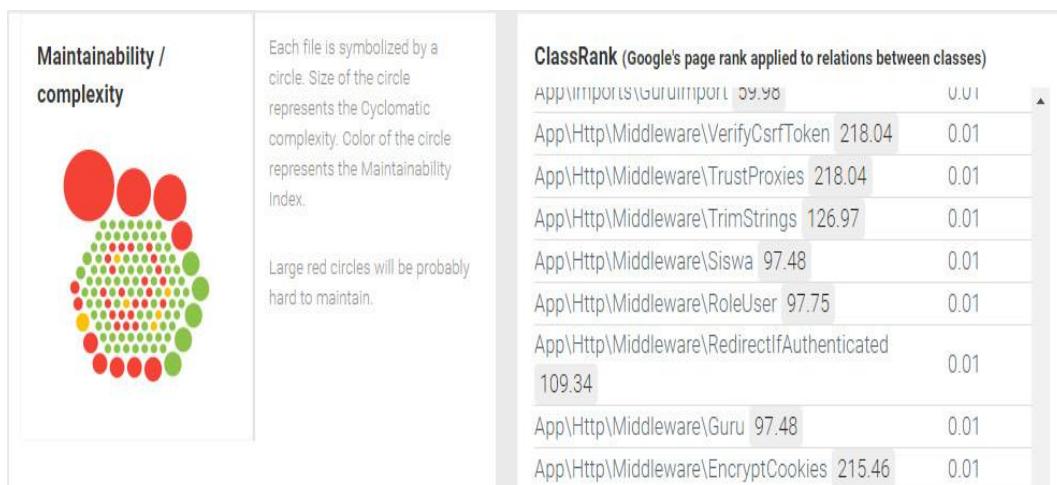
Karakteristik *maintainability* sistem informasi manajemen pembelajaran diuji dengan bantuan *software* PHPMetrics. Hasil pengujian menggunakan PHPmetrics terbagi kedalam beberapa subkategori seperti yang ditunjukkan Gambar 34.



Gambar 34. Hasil Pengujian *Maintainability*

Gambar 34 menunjukan bahwa sistem yang dikembangkan ditulis menggunakan 4702 baris kode program, jumlah *class* sebanyak 130. Rata-rata *bug* tiap *class* bernilai 0,12 artinya kemungkinan program *error* atau tidak berfungsi sebagaimana mestinya sangatlah kecil. Hal ini didukung pula dari hasil analisis *violations* tipe *criticals* dan *error* memiliki nilai 0. Kompleksitas program tiap kelas bernilai 2,35 jika diinterpretasikan berdasarkan Tabel 11 maka masuk kedalam level *low risk program*.

Hasil perhitungan *Maintainability Index* pada PHPmetrics disajikan dalam bentuk tiap *class* seperti diperlihatkan pada Gambar 35. Untuk memperoleh nilai komulatif maka data perlu diolah dengan mencari nilai rata-rata dari keseluruhan nilai MI tiap *class*. Keseluruhan data MI tiap kelas dapat dilihat pada Lampiran 13.

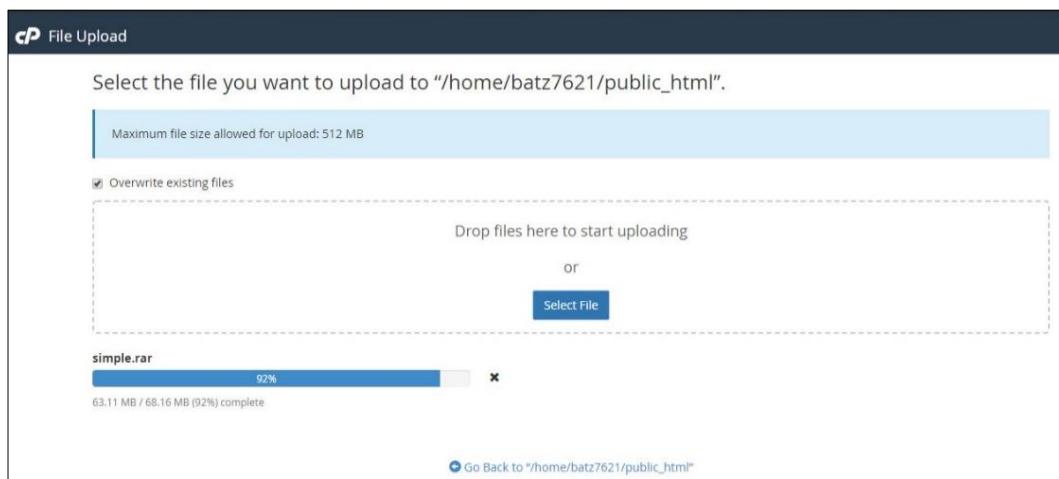


Gambar 35. *Maintainability Index* Setiap *Class*

Dari hasil perhitungan diperoleh *Maintainability Index* (MI) dengan skor 120. Jika diubah kedalam skala 100 maka skor MI menjadi 100. Berdasarkan skor MI bisa ditarik kesimpulan bahwa sistem berhasil memenuhi standar kriteria.

## 6. Deployment

Kegiatan *deployment* diantaranya melakukan *upload* kode program ke *shared/cloud hosting*, melakukan konfigurasi agar sistem dapat diakses melalui jaringan dan membuat panduan penggunaan sistem. Gambar 36 menunjukan proses *upload* ke *hosting* menggunakan cpanel.



Gambar 36. Publish Sistem ke *Hosting*

## 7. Configuration & Change Management

Berkaitan dengan aktivitas manajemen dalam sebuah tim pengembang, peneliti tidak melakukan aktivitas proses ini karena proyek dikerjakan sendiri.

## 8. Project Management

Aktivitas pengembangan masih mengikuti *timeline* yang dibuat pada tahap *inception*.

## 9. Environment

Tidak perlu dikerjakan lagi, karena penyediaan keperluan pengembangan secara lengkap sudah dilakukan pada tahap *inception*.

## **E. Pembahasan Hasil Penelitian**

### **1. Pembahasan Produk**

Beberapa fitur yang utama yang berhasil dikembangkan untuk menyelesaikan permasalahan dalam manajemen pembelajaran meliputi: (1) Fitur konsultasi untuk bimbingan dan verifikasi dokumen administrasi guru sehingga menjamin perangkat pembelajaran guru sejalan dengan program kurikulum. (2) Fitur *drive* untuk mengelola berkas atau *file* mulai dari penyimpanan, berbagi, pengaturan hak akses sehingga semua dokumen sekolah dan guru bisa tertata dengan baik di dalam *database* sistem. (3) Fitur info memudahkan siswa dan guru melakukan akses informasi akademik ataupun non-akademik sekolah sesuai dengan status pengguna. (4) Fitur *classroom* berguna dalam melaksanakan aktivitas pembelajaran virtual mulai dari mendesain pembelajaran, mengorganisasikan materi ajar, pengelolaan tugas, membuat catatan pengajaran, melihat catatan tugas dan catatan belajar siswa sampai pada melakukan penilaian. Membantu guru dalam berinteraksi dan mengontrol aktivitas belajar siswa diluar jam pembelajaran kelas. (5) Fitur pelacakan belajar mampu merekam aktivitas belajar siswa dan tingkat penguasaan materi sehingga membantu guru dalam proses evaluasi, mendiagnistik kendala belajar dan menentukan keputusan guna perbaikan kegiatan belajar mengajar berikutnya. (6) Fitur *report* membantu menyelesaikan administrasi pembelajaran lebih cepat karena pembuatan laporan daftar hadir, pengajaran guru, penilaian hasil belajar dapat di *generate* sistem secara otomatis.

## 2. Pembahasan Hasil Pengujian

Berikut pembahasan terkait hasil pengujian sistem informasi manajemen pembelajaran siswa pada karakteristik *functional suitability, usability, reliability, performance efficiency, dan maintainability*.

### a. *Functional Suitability*

Terdapat tiga subkarakteristik *functional suitability* yang diuji yaitu *functional completeness, functional correctness, dan functional appropriateness*. Hasil pengujian menunjukan bahwa nilai X dari ketiga subkarakteristik tersebut adalah 1, artinya semua fitur berhasil diterapkan, berjalan dengan benar 100% sesuai dengan rancangan awal. Sehingga dapat ditarik kesimpulan bahwa sistem informasi manajemen pembelajaran siswa berhasil memenuhi standar karakteristik *functional suitability*.

### b. *Usability*

Berdasarkan persentase pengujian *usability* didapatkan skor sebesar 87,2%. Jika diinterpretasikan maka sistem yang dikembangkan masuk kedalam kategori Sangat Layak dan memenuhi standar kriteria. Dari segi instrumen pengujian yang digunakan mendapatkan nilai Cronbach's Alpha 0,955. Jika diterjemahkan masuk kategori *excellent*. Karena nilai Cronbach's Alpha 0,955 melebihi batas standar yaitu 0,60 dan melebihi nilai r tabel 0,361 (dengan signifikansi 5%) maka dapat instrumen *usability* dikatakan reliabel atau konsisten. Sehingga dapat ditarik kesimpulan bahwa sistem informasi manajemen pembelajaran siswa berhasil memenuhi standar karakteristik *usability*.

c. *Reliability*

Pengujian sistem informasi manajemen pembelajaran siswa pada karakteristik *reliability* menggunakan WAPT Pro v.5.0. dengan hasil seperti yang ditunjukan pada Tabel 28.

Tabel 28. Hasil Pengujian *Reliability*

<b>Metrik</b>	<b>Sukses</b>	<b>Gagal</b>
<i>Session</i>	6292	0
<i>Pages</i>	18876	0
<i>Hits</i>	122694	0

Berdasarkan *report* dari *software* WAPT Pro v5.0, selama pengujian tidak ditemukan adanya kesalahan/kegagalan yang terjadi, karena memiliki persentase keberhasilan *session*, *pages* dan *hits* sebesar 100 %. Sehingga dapat ditarik kesimpulan bahwa sistem informasi manajemen pembelajaran siswa berhasil memenuhi standar karakteristik *reliability*.

d. *Performance Efficiency*

Pengujian *performance efficiency* dengan *software* GTMetrix menghasilkan nilai rata-rata *page speed* sebesar 99,72% (Grade A), Yslow sebesar 93,96% (Grade A) , dan *load time* sebesar 1,49 detik. *Load time* tidak lebih dari 10 detik, menandakan bahwa sistem memenuhi standar kriteria. Sehingga dapat ditarik kesimpulan bahwa sistem informasi manajemen pembelajaran siswa berhasil memenuhi standar karakteristik *performance efficiency*.

e. *Maintainability*

Dari hasil perhitungan diperoleh *Maintainability Index* (MI) dengan skor 120. Jika diubah kedalam skala 100 maka skor MI menjadi 100. Berdasarkan skor MI

sistem masuk kedalam kategori sangat mudah dirawat. Sedangkan kompleksitas program berdasarkan *class* memiliki skor 2,35 masuk kedalam kategori *low risk program*. Sehingga dapat ditarik kesimpulan bahwa sistem informasi manajemen pembelajaran siswa berhasil memenuhi standar karakteristik *maintainability*.