

I. PENDAHULUAN

1.1 Latar Belakang

Banyak sekolah yang mengikuti arus perkembangan teknologi informasi, sekolah-sekolah tersebut mengikuti perkembangan teknologi guna meningkatkan kemajuan sistem pendidikan di sekolah. Salah satu sekolah yang ingin memajukan sistem pendidikannya adalah SMK Swadhipa 1 yang berada di kecamatan Natar, Lampung Selatan. SMK tersebut merupakan Sekolah Menengah Kejuruan. Sekolah ini merupakan salah satu sekolah yang bergerak dalam bidang keilmuan teknologi yang menuntut siswanya untuk terbiasa dengan keberadaan teknologi informasi pada masa kini.

Teknologi informasi sangat diperlukan dalam rangka member inovasi dalam sistem pembelajaran disekolah. Sistem pembelajaran yang ada pada saat ini masih dalam bentuk arsip sehingga mengalami kendala dalam penyimpanannya, dan mengalami keterbatasan ruang dan waktu untuk mengaksesnya,serta orang tua siswa masih mengalami kesulitan dalam memonitoring nilai anak di sekolah. Sistem yang berjalan saat ini juga dapat membuat siswa dan guru bergerak statis. Berdasarkan masalah tersebut, diperlukan adanya sebuah sistem pembelajaran yang dapat digunakan untuk memudahkan dan memperkuat model pembelajaran siswa disekolah secara efektif dan efisien. Adapun tujuan dari penulisan ini adalah untuk menghasilkan sebuah aplikasi pembelajaran *online* berbasis *web* yang akan memudahkan siswa dan

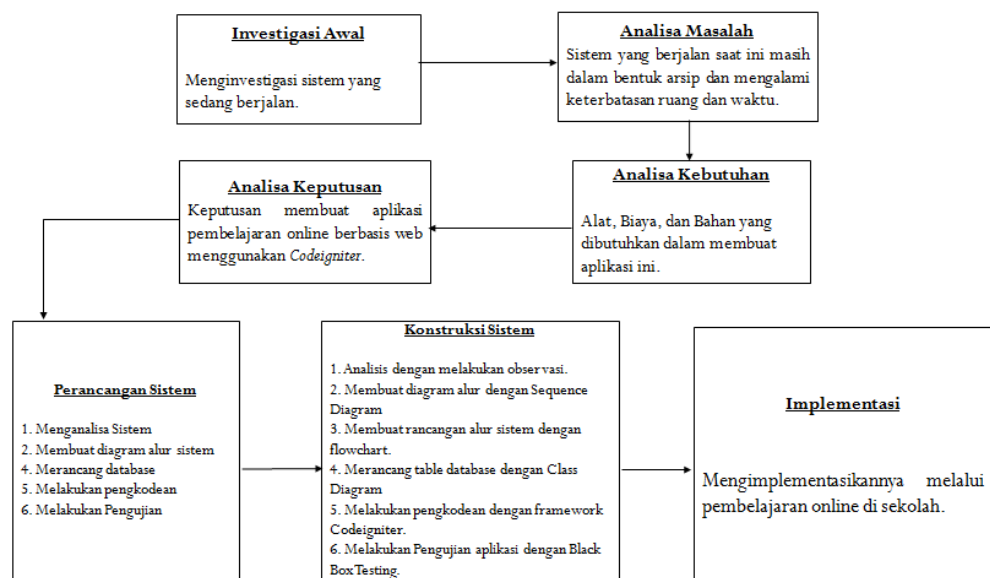
guru dalam kegiatan belajar mengajar dan untuk meningkatkan kualitas sistem pendidikan di sekolah tersebut.

1.2 Tujuan

Tujuan dari penulisan ini adalah untuk menghasilkan sebuah aplikasi Pembelajaran *Online* Berbasis *Web* agar dapat membantu menangani masalah pembelajaran yang ada di SMK Swadhipa 1 Natar, Lampung Selatan.

1.3 Kerangka Pemikiran

Agar dalam pembuatan sistem ini lebih terarah maka penulis menggunakan kerangka pemikiran. Tahapan kerangka pemikiran dalam pembuatan aplikasi ini disajikan pada Gambar 1.



Gambar 1. Gambar Kerangka Pemikiran

1.4 Kontribusi

Kontribusi yang diharapkan dari pembuatan Aplikasi Pembelajaran *Online* Berbasis *Web* ini diantaranya :

- (1) Mempermudah proses belajarmengajar di kelas.
- (2) Memberi akses bagi guru, siswa agar dapat belajar dan memonitoring dimana saja dan kapan saja.
- (3) Memberi akses bagi orang tua siswa agar dapat melihat nilai anak di sekolah.

1.5 Batasan Masalah

Agar penulisan laporan proyek mandiri ini lebih terarah dan permasalahan yang dihadapi tidak lebih luas maka perlu adanya batasan masalahyaitu :

1. Aplikasi Pembelajaran *Online* Berbasis *Web* hanya tentang penyampaian materi dan evaluasi belajar siswa dan ujian pada Jurusan Multimedia.
2. Informasi yang dihasilkan nantinya berupa laporan hasil evaluasi belajar siswa-siswidi SMK Swadhipa 1 Jurusan Multimedia.

II. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi merupakan kumpulan perintah program yang dibuat khusus untuk melakukan pekerjaan atau prosedur tertentu (Hendrayudi, 2009)

2.2 Pembelajaran

Pembelajaran adalah “membelajarkan siswa menggunakan asas pendidikan maupun teori belajar yang merupakan penentu utama keberhasilan pendidikan”. Pembelajaran merupakan proses komunikasi dua arah. Mengajar dilakukan pihak guru sebagai pendidik, sedangkan belajar oleh peserta didik.(Sagala, 2009).

2.3 Internet

Internet adalah suatu komputer yang saling terhubung dengan jaringan lain, dan melakukan aktivitas komunikasi antara yang satu dengan yang lainnya dalam area yang tak terbatas (Setiawan, 2012).

2.4 Website

Website merupakan sekumpulan halaman-halaman web yang diperoleh dari sebuah alamat domain dan berisi informasi. *Hyperlink* adalah hubungan antara satu halaman dengan halaman web lain, sedangkan *hypertext* adalah teks yang menjadi alat penghubung (Yuhefizar, 2013).

2.5 Data

Data adalah hasil dokumentasi atas suatu kejadian atau fakta yang terjadi. Untuk merekam data dalam media digital, maka diperlukan sebuah *database* (Setiawan, 2012).

2.6 Database

Database adalah kumpulan informasi dari suatu subjek, memiliki keterkaitan yang lengkap dan terstruktur. *Database* menyediakan struktur informasi yang digunakan secara bersama-sama oleh aplikasi-aplikasi yang berbeda (Setiawan, 2012).

2.7 Hypertext Markup Language (HTML)

Hyper Text Markup Language atau HTML adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* internet (Prayitno, 2010).

2.8 Hypertext Preprocessor (PHP)

Hypertext Preprocessor atau PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML (Anisya, 2013). PHP banyak dipakai untuk membuat situs *web* dinamis, yaitu kode PHP diselipkan diantara script kode-kode HTML yang merupakan bahasa *markup* standar untuk dunia *web* (Supardi, 2015).

2.9 Cascading Style Sheet (CSS)

Cascading Style Sheet atau CSS adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur *style* suatu dokumen. Pada umumnya CSS dipakai

untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML (Sulistiyawan dkk, 2008).

2.10 Javascript

Javascript adalah bahasa script populer yang dipakai untuk menciptakan halaman *web* yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. *Javascript* merupakan perekat yang menyatukan halaman-halaman *web* (Winarno dkk, 2014).

2.11 MySQL

MySQL adalah suatu perangkat lunak *database* relasi (*Relational Database Management Sistem*), seperti halnya *ORACLE*, *POSTGRESQL*, *MSSQL*, dan sebagainya. *SQL* merupakan singkatan dari *Structure Query Language*, didefinisikan sebagai sebuah sintaks perintah-perintah tertentu yang digunakan untuk mengelola suatu *database* (Anisya, 2013).

2.12 PHPMyAdmin

PHPmyadmin adalah sebuah *software* berbasis pemrograman PHP yang dipergunakan sebagai administrator MySQL melalui *browser (web)* yang digunakan untuk manajemen *database* (Rahman, 2013).

2.13 XAMPP

XAMPP adalah sebuah *software web server* apache yang di dalamnya sudah tersedia *database server* mysql dan *support php programming* (Handayani, 2012).

2.14 *Bootstrap*

Bootstrap adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. *Bootstrap* adalah *template* desain *web* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman (Rozi, 2015).

2.15 *Notepad++*

Notepad++ merupakan sebuah *software* yang digunakan untuk menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan di atas sistem operasi *Microsoft Windows* (Masruri, 2015).

2.16 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah salah satu standar bahasa yang umum digunakan di dunia industri untuk mendefinisikan *requirement*, menganalisis dan membuat desain, serta menggambarkan arsitektur dalam pemrograman berorientasi pada objek (Rosa dan Shalahuddin, 2014).

Beberapa bagian dari UML diagram yang digunakan dalam perancangan aplikasi, adalah sebagai berikut :

1. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah

interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa dan Shalahuddin, 2014).

2. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak (Rosa dan Shalahuddin, 2014).

3. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa dan Shalahuddin, 2014). Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas.

2.17 FAST

FAST (Framework for the Application Of Sistem Thinking) merupakan sebuah metodologi bersifat *agile* yang cukup fleksibel untuk mendukung berbagai macam proyek serta strategi pengembangan sistem. Huruf-huruf pada *FAST* merupakan singkatan dari “*Framework for the Application of Sistem Thinking*”. Fase-fase pada *FAST* antara lain: *scope definition, problem analysis, requirements analysis, logical design, decision analysis, physical design and integration, construction and testing, installation and delivery* (L. Whitten & Bentley, 2004)

1. Scope Definition

Merupakan fase yang umum pada awal proyek, yang akan menjabarkan masalah-masalah apa yang perlu diatasi, disesuaikan dengan ukuran, batasan, serta tujuan proyek. Fase *scope definition* ini dipicu oleh beberapa kombinasi dari *problems, opportunities, dan directives* dimana kerangka kerja *PIECES* memberi penjelasan problem statement yang baik.

2. Problem Analysis

Mempelajari sistem yang sudah ada dan menganalisis penemuan-penemuan masalah untuk mendukung tim proyek dalam memahami masalah-masalah yang menyebabkan proyek itu. Hasil dari analisis masalah adalah memutuskan proyek untuk selanjutnya: (a) dibatalkan karena masalah tidak pantas diselesaikan atau (b) disetujui lalu dilanjutkan ke fase selanjutnya atau (c) ruang lingkup yang dikurangi atau ditambahkan (dengan perubahan biaya dan jadwal), lalu dilanjutkan ke fase selanjutnya.

3. Requirement Analysis

Fase ini mendefinisikan dan memprioritaskan kebutuhan bisnis untuk menentukan kebutuhan secara fungsional dan non-fungsional. Seperti kemampuan apa yang harus disediakan sistem kepada pengguna, seperti ukuran tingkat kinerja yang diperlukan, data apa yang harus diambil dan disimpan.

4. *Logical Design*

Terjemahan dari kebutuhan pengguna dari sisi bisnis kepada model sistem yang hanya menggambarkan kebutuhan sistem dan bukan rancangan bersifat teknis apalagi penerapan dari kebutuhan tersebut.

5. *Decision Analysis*

Berdasarkan kebutuhan bisnis dan desain logika, tentunya sangat banyak alternatif keputusan yang dapat diambil untuk merancang sistem informasi dalam memenuhi kebutuhan-kebutuhan terdefinisi. Beberapa pertimbangan yang biasanya muncul adalah: proses mana saja yang sebaiknya diotomatisasi, apakah sistem dibangun sendiri atau membeli dari *vendor*, apakah mendesain secara internal atau berbasis *web*, dan lain sebagainya.

6. *Physical Design and Integration*

Terjemahan dari kebutuhan pengguna dari sisi bisnis kepada model sistem yang menggambarkan penerapannya secara teknis.

7. *Construction and Testing*

Fase dimana model fisik dari sistem akan dibangun dan diuji untuk mengetahui apakah sistem telah memenuhi kebutuhan dan spesifikasi yang telah

direncanakan. Ada pula implementasi rancangan antarmuka sistem lama dengan sistem baru, serta antarmuka pengguna.

8. *Installation and Delivery*

Pada fase ini terjadi transisi sistem lama ke sistem baru dan membantu pengguna dalam pelatihan dalam penggunaan sistem baru. Untuk mendukung transisi yang mulus, harus ada perencanaan yang matang supaya tidak terjadi kerugian saat ditinggalkannya sistem lama.

2.18 Framework

Framework adalah kerangka kerja sebuah program. *Framework* juga dapat diartikan sebagai kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer/programmer* dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke *database*, pemanggilan *variable*, dan *file*. Sehingga *developer* lebih fokus dan lebih cepat membangun aplikasi (Rosmala dkk, 2011).

2.19 Codeigniter

Menurut Blanco dan Upton (2009:7) Codeigniter adalah *Powerful Open Source PHP framework* yang mudah dikuasai, dibangun untuk *PHP programmers* yang membutuhkan *toolkit* sederhana dan baik untuk membuat *full-featured web applications*. *CodeIgniter* adalah *MVC framework* yang di desain untuk mempermudah *programmer*.

2.20 Model-View-Controller (MVC)

Model-View-Controller atau MVC adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (*Trygve Reenskaug*) untuk enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk dipresentasikan pada sebuah *user interface* (Hidayat dkk, 2012).

Definisi teknis dari arsitektur *Model View Controller* (MVC) dibagi menjadi tiga lapisan (Hidayat dkk, 2012) :

1. *Model*

Model digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Hanya model yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data.

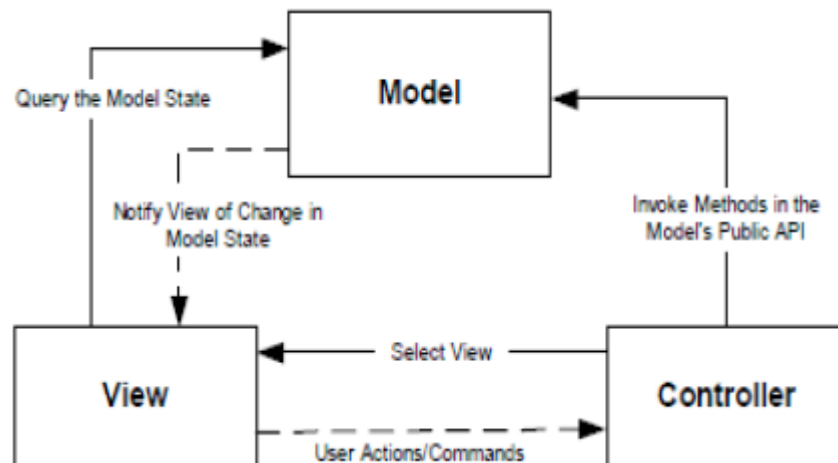
2. *View*

View bertanggung jawab untuk pemetaan grafis ke sebuah perangkat. *View* biasanya memiliki hubungan dengan sebuah permukaan layar dan tahu bagaimana untuk membuatnya. *View* melekat pada model dan *me-render* isinya ke permukaan layar.

3. *Controller*

Controller menerima input dari pengguna dan menginstruksikan *model* dan *view* untuk melakukan aksi berdasarkan masukan tersebut. *Controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi.

Model, *view* dan *controller* saling berhubungan. Oleh karena itu, untuk mengilustrasikan hubungan dasar *Model-View-Controller* dapat dilihat pada Gambar 2.



Gambar 2. Hubungan *model*, *view* dan *controller*

Sumber : Hidayat dkk (2012)

2.21 Object Oriented Programming (OOP)

Pemrograman berorientasi objek atau *object-oriented programming* merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class* (Wibowo, 2015). Konsep dasar pemrograman berorientasi objek atau *object-oriented programming* (OOP) yang dapat berjalan di PHP adalah :

1. *Class* didefinisikan dengan menampung nilai properti dan *method*, properti adalah sebuah data yang menjelaskan tentang *class* dan metode adalah tingkah laku yang dapat dilakukan oleh *object*.
2. *Object* adalah hasil instansiasi dari *class* dan mengandung seluruh *resource* yang telah didefinisikan pada *class*.
3. *Encapsulation* adalah mekanisme “membungkus” sebuah data pada sebuah *object*. Tiga *modifier* yang dapat diimplementasikan untuk melakukan “pembungkusan” data yaitu *private*, *protected*, dan *public*.
4. *Polymorphisme* membuat objek-objek yang berasal dari *subclass* yang berbeda, diperlakukan sebagai objek-objek dari satu *superclass*.

5. *Constructor*, *class* yang memiliki metode konstruktor memanggil metode ini pada setiap objek yang baru dibentuk.
6. *Destructor*, *destructor* akan segera dipanggil setelah tidak ada referensi lain.
7. *Inheritance* (pewarisan) adalah cara untuk menggunakan kembali kode objek yang ada, atau untuk mendirikan *subtype* dari objek yang sudah ada, atau keduanya, tergantung pada dukungan bahasa pemrograman.
8. *Final Keyword* adalah untuk mencegah proses *overriding method* pada *class* anak (sub-*class*) hal ini dapat diterapkan pada metode dan *class*.
9. *Class Abstraction*, *class* yang mendefinisikan sebagai *abstract* tidak bisa diintiasiasi, dan *class* yang terdiri paling tidak satu *method abstract* harus didefinisikan sebagai *abstract class*.
10. *Object Interfaces* memungkinkan untuk membuat kode yang menentukan *method* mana yang akan diimplementasi tanpa harus mendefinisikan bagaimana *method* tersebut akan bekerja (hanya nama *method* saja).

2.22 Black Box Testing

Black box testing atau pengujian kotak hitam, disebut juga pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Teknik pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program (Pressman, 2010).

Menurut Simarmata (2010), klasifikasi *Black Box Testing* mencakup beberapa pengujian yaitu :

1. Pengujian fungsional (*functional testing*)

Perangkat lunak diuji untuk persyaratan fungsional. Pengujian dilakukan dalam bentuk tertulis untuk memeriksa perintah-perintah pengguna, manipulasi data, serta operasi *back-end*.

2. Pengujian tegangan (*stress testing*)

Pengujian tegangan berkaitan dengan kualitas aplikasi di dalam lingkungan.

3. Pengujian beban (*load testing*)

Pada pengujian beban aplikasi akan diuji dengan beban berat atau masukan, seperti yang terjadi pada pengujian situs *web*.

4. Pengujian khusus (*ad-hoc testing*)

Jenis pengujian *ad-hoc* dilakukan tanpa penciptaan rencana pengujian (*test plan*) atau kasus pengujian (*test case*).

5. Pengujian penyelidikan (*exploratory testing*)

Pengujian penyelidikan sama dengan pengujian khusus dan dilakukan untuk mempelajari atau mencari aplikasi.

6. Pengujian usabilitas (*usability testing*)

Pengujian ini disebut juga pengujian untuk keakraban pengguna (*testing for user-friendliness*). Pengujian ini dilakukan jika *interface* pengguna dari aplikasi penting dan harus spesifik untuk jenis pengguna tertentu.

7. Pengujian asap (*smoke testing*)

Pengujian ini disebut pengujian kenormalan (*sanity testing*). Pengujian ini dilakukan untuk memeriksa apakah aplikasi tersebut sudah siap untuk pengujian yang lebih besar dan bekerja dengan baik tanpa celah sampai tingkat yang paling diharapkan.

8. Pengujian pemulihan (*recovery testing*)

Pengujian pemulihan (*recovery testing*) pada dasarnya dilakukan untuk memeriksa seberapa cepat dan baiknya aplikasi dapat pulih terhadap semua jenis *crash* atau kegagalan hardware.

9. Pengujian volume

Pengujian volume dilakukan terhadap efisiensi dari aplikasi.

10. Pengujian domain (*domain testing*)

Pengujian domain merupakan penjelasan yang sangat sering menjelaskan teknik pengujinya.

11. Pengujian *scenario*

Pengujian skenario adalah pengujian yang realistis, kredibel dan memotivasi *stakeholders*, tantangan untuk program dan mempermudah penguji untuk melakukan evaluasi.

12. Pengujian regresi (*regression testing*)

Pengujian regresi adalah gaya pengujian yang berfokus pada pengujian ulang (*retesting*) setelah ada perubahan.

13. Penerimaan pengguna (*user acceptance*)

Pada jenis pengujian ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan.

14. Pengujian alfa (*alpha testing*)

Pengujian ini, pengguna akan diundang ke pusat pengembangan.

15. Pengujian beta (*beta testing*)

Pada jenis ini, perangkat lunak didistribusikan sebagai sebuah versi beta dengan pengguna yang menguji aplikasi di situs mereka.

III. METODE PELAKSANAAN

3.1 Tempat dan Waktu

Proyek Usaha Mandiri berjudul “Aplikasi Pembelajaran *Online* berbasis *Website*” ini akan dilaksanakan pada bulan September – Desember 2017 di Politeknik Negeri Lampung.

3.2 Alat dan Bahan

Alat dan bahan yang dibutuhkan dalam penulisan Proyek Mandiri dengan judul “Aplikasi Pembelajaran *Online* berbasis *Website*” dalam pembuatan program ini ada 2 jenis, yaitu perangkat keras dan perangkat lunak seperti :

3.2.1 Perangkat Keras

Perangkat keras yang digunakan adalah salahsatu unit laptop denganspesifikasisebagaiberikut:

1. Laptop
2. Modem

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

1. *Sistem Operasi Microsoft Windows 8.1*
2. *XAMPP*
3. *Web Browser*

4. Aplikasi *Enterprise Architect*
5. Aplikasi *Easy Case*
6. Aplikasi *Microsoft Visio*

3.3 Metode Pengembangan sSistem

Metodologi pengembangan yang digunakan dalam perancangan Aplikasi Pembelajaran *Online* Berbasis *Web* menggunakan metode *FAST*. Tahapan-Tahapannya adalah sebagai berikut :

3.3.1 Scope Definition

Tahapan ini merupakan tahapan yang umum, berupa penjelasan masalah-masalah yang terdapat pada sistem yang sedang berjalan. Dengan mengamati permasalahan yang ada saat ini pengamat memperoleh peluang untuk membangun aplikasi pembelajaran yang efektif dan efisien.

3.3.2 Problem Analysis

Tahapan menganalisa masalah yaitu menganalisis permasalahan yang ditemukan dalam sistem yang sedang berjalan saat ini, permasalahan yang ada mendukung proyek penulis untuk membuat sistem dan dari masalah yang ada dapat menghasilkan keputusan pembangunan aplikasi ke tahap yang selanjutnya.

3.3.3 Requirement Analysis

Tahapan ini menjelaskan apa saja kebutuhan-kebutuhan utama. Aplikasi yang dibuat akan menyediakan sistem seperti apa kepada guru dan siswa. Kemudian kinerja aplikasi dan data yang disimpan dan dikelola.

3.3.4 Logical Design

Tahapan *Logical Design* menggambarkan kebutuhan sistem, kebutuhan siswa dan guru kepada model sistem yang diberikan.

3.3.5 Decision Analysis

Tahapan ini merupakan analisa keputusan, dalam tahap ini penulis memutuskan sistem yang dibuat dibangun sendiri tidak melalui pihak-pihak lain, mendesain sistem secara interna dan aplikasi yang dibuat berbasis website.

3.3.6 Physical Design and Integration

Tahapan ini menggambarkan kebutuhan siswa dan guru kepada model yang digambarkan secara teknikal.

3.3.7 Construction and Testing

Tahapan ini merupakan tahapan pembangunan sistem dan memuat rancangan antarmuka sistem lama dengan sistem yang baru, kemudian melakukan pengujian menggunakan *black box testing* untuk mengetahui kelemahan sistem.

3.3.8 Installation and Delivery

Tahapan ini adalah transisi sistem yang belum terkomputerisasi ke sistem yang berbasis web, kemudian membantu siswa dan guru dalam pelatihan penggunaan sistem yang baru.

3.4 Analisa Biaya

Tabel 1. Anggaran Biaya

No	Penggunaan	Jumlah	Biaya
1.	Transportasi		Rp. 75.000,-
2.	Jasa Print		Rp. 75.000,-
3.	Biaya Pembuatan Program		
	a. Analisis & Desain Sistem		Rp. 2000.000,-
	b. Pengkodean		Rp. 3.500.000,-
4.	Alat Tulis		
	a. Pena	3 Buah	Rp. 3.000,-
	b. Pensil	1 Buah	Rp. 5.000,-
	c. Penghapus	1 Buah	Rp. 5.000,-
	d. Type-X	1 Buah	Rp. 5.000,-
	f. Map	3 Buah	Rp. 6.000,-
5.	Buku Tinjauan Pustaka	1 Buah	Rp. 200.000,-
6.	Hosting Web	1 Tahun	Rp. 700.000,-
Total			Rp. 6.574.000,-

IV. HASIL DAN PEMBAHASAN

4.1 Gambaran Umum

Gambaran umum sekolah merupakan penggambaran berupa penjelasan tentang sejarah umum, visi, misi dan struktur organisasi SMK Swadhipa 1 Kecamatan Natar, kabupaten Lampung Selatan.

4.1.1 Sejarah Umum

SMK 1 Swadhipa Natar berdiri pada 17 Maret 1987 kemudian dibukalah bidang keahlian Bisnis dan Manajemen dengan Program Keahlian Akuntansi. Pada tahun 2005 SMK Swadhipa 1 Natar ditetapkan memperoleh akreditasi dengan peringkat B, pada tahun 2012 SMK Swadhipa 1 Natar mendapatkan sertifikat ISO.

SMK Swadhipa 1 Natar beralamat di jl. Raya bumi sari no.217 Natar, lampung selatan. SMK Swadhipa 1 Natar sudah lebih dari 2 Dekade memberikan bekal pengetahuan, pemahaman serta wawasan tentang bidang keahlian kepada seluruh siswanya sebagai mata rantai dan mata kehidupan dalam mencerdaskan anak bangsa.

4.1.2 Visi

Mewujudkan Sekolah Berkualitas, Berdaya Saing Melalui Pendidikan Berdasarkan IPTEK Dan IMTAK. Dengan Indikator :

a. Mampu Bersaing Dalam Bidang Bisnis Dan Manajemen Agar Dapat Diterima Di Dunia Usaha

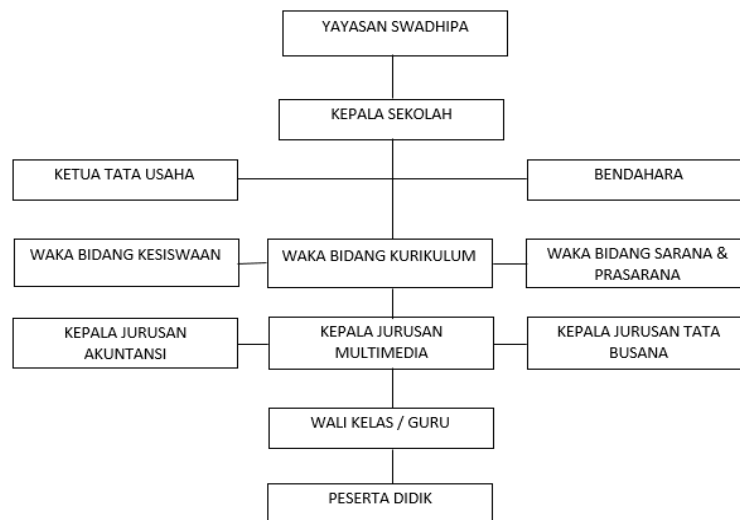
b. Mampu Mewujudkan Kegiatan Ekstrakurikuler Yang Sesuai Dengan Bakat Dan Minat Peserta Didik.

4.1.3 Misi

- a. Meningkatkan Proses Belajar Mengajar Efektif Dan Efisien
- b. Meningkatkan Kualitas Pendidikan Dan Tenaga Kependidikan Melalui Diklat Dan MGMP
- c. Meningkatkan Aktifitas Dalam Bidang Akademik Dan Non Akademik
- d. Meningkatkan Kedisiplinan Seluruh Warga Sekolah Menjaga Dan Melestarikan Kebersihan, Kenyamanan, Dan Kekeluargaan Di Lingkungan Sekolah.

4.1.4 Struktur organisasi

Struktur organisasi sebuah sekolah memiliki peran yang sangat penting dalam menjalankan kegiatan di suatu sekolah. Struktur organisasi sekolah harus dapat meningkatkan mutu suatu sekolah ke arah yang lebih baik dan dapat menciptakan koordinasi dan kerjasama diantara semua bagian yang terdapat dalam sekolah dalam rangka meningkatkan kualitas pendidikan dan tanggung jawab. Adapun struktur organisasi SMK Swadhipa 1 Natar Lampung Selatan dapat dilihat pada Gambar



Gambar 3. Struktur Organisasi SMK Swadhipa 1.

4.2 Hasil dan Pembahasan

Hasil dari proyek mandiri yang berjudul “Aplikasi pembelajaran *online* berbasis website” melalui beberapa tahapan yaitu :

4.2.1 Scope Definition

Tahap ini bertujuan untuk mengetahui sistem yang sedang berjalan saat ini, kemudian untuk mengetahui kekurangan dan masalah-masalah yang akan diatasi dengan konsep *problems*, *opportunities*, dan *directives*.

4.2.2 Problem Analysis

Narasumber yang memberikan informasi adalah ketua jurusan multimedia SMK Swadhipa 1 yaitu Saudara Heri Kurniawan. Kemudian kami melakukan observasi permasalahan dan melihat peluang yang ada pada sistem pendidikan di SMK tersebut.

Hasil observasi terhadap sistem pembelajaran yang sedang berjalan di SMK Swadhipa 1 adalah :

1. Siswa menerima materi di kelas, materi yang diterima berupa penyampaian secara lisan oleh guru, dan materi dalam bentuk *hardcopy*.
2. Guru memberikan materi secara lisan saat di kelas, dan member materi kepada siswa dalam bentuk *hardcopy*.

Dari observasi yang telah dilakukan didapatkan beberapa informasi yaitu sejarah sekolah, visi misi, struktur organisasi SMK Swadhipa 1, dan sistem yang sedang berjalan saat ini.

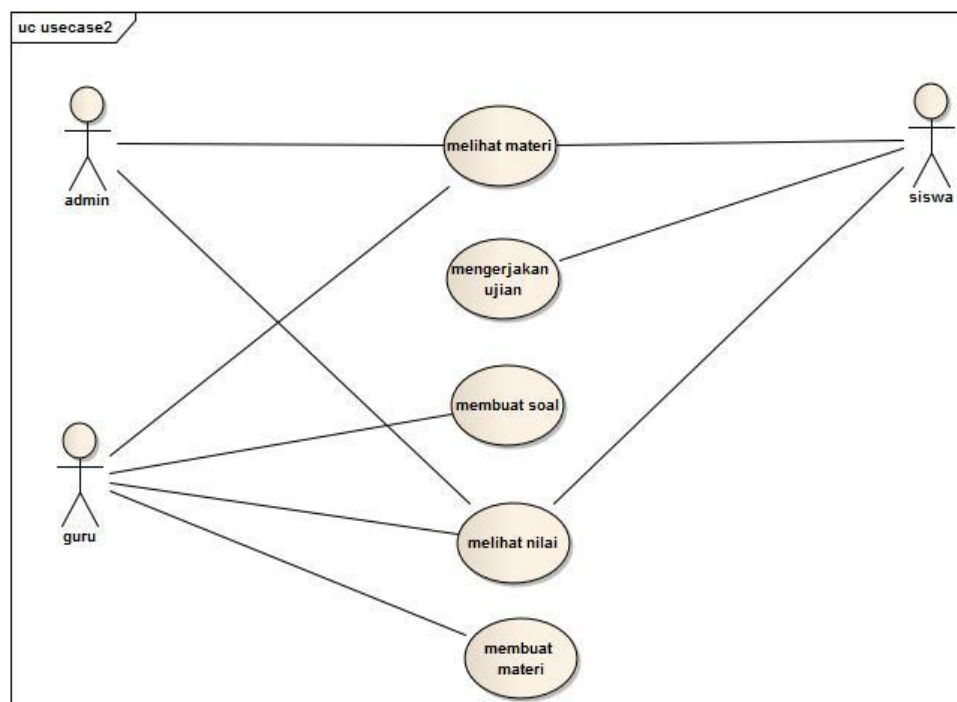
4.2.3 Requirement Analysis

Kebutuhan fungsional masing-masing user yaitu admin, siswa, dan guru dijelaskan sebagai berikut ini :

1. Level Admin
 - a. Mengelola data siswa dan guru
 - b. Menambahkan, mengubah serta menghapus data siswa & guru
 - c. Menambahkan dan mengubah data admin
2. Level Guru
 - a. Mengelola data materi dan soal ujian
 - b. Mengelola data kelas
3. Level Siswa
 - a. Mengakses materi
 - b. Mengikuti ujian

4.2.4 Logical Desain

Pada tahapan *logical desain* membutuhkan *Unified Modelling Language* (UML) untuk mendeskripsikan seluruh proses yang terjadi pada sistem yang akan diusulkan. UML merupakan suatu alat untuk menggambarkan kemudian membuat hasil analisa menjadi bentuk visual yang berisi sintak. Berikut ini *use case diagram* dalam UML yang digunakan dalam perancangan sistem yang diusulkan yaitu aplikasi pembelajaran *online* berbasis web disajikan pada Gambar 4.



Gambar 4. *Use Case Diagram*

Pada aplikasi pembelajaran *online* berbasis web ini terdapat 5 (lima) usecase seperti yang disajikan pada Gambar 4.

a. *Use case diagram* level admin

Use case diagram level admin menggambarkan perilaku yang dilakukan oleh admin dalam system aplikasi. Tingkah laku yang dilakukan oleh admin adalah login akun, melihat data, mengelola data siswa dan guru.

b. *Use case diagram* level siswa

Use case diagram level siswa menggambarkan perilaku yang dilakukan oleh siswa dalam aplikasi pembelajaran *online* berbasis web. Tingkah laku yang dapat dilakukan oleh siswa adalah login, melihat materi, mengikuti ujian, dan melihat nilai.

c. *Use case diagram* level guru

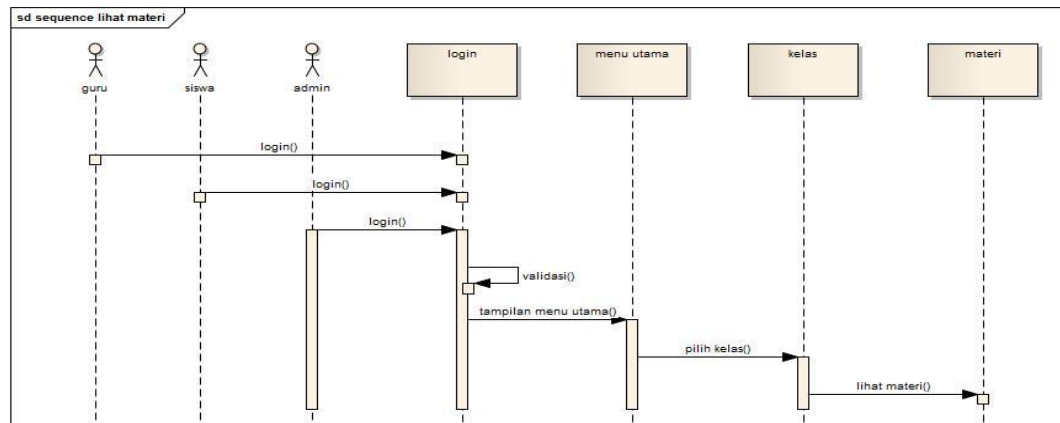
Use case diagram level guru menggambarkan perilaku yang dilakukan oleh guru dalam aplikasi pembelajaran *online* berbasis web. Tingkah laku yang dapat dilakukan oleh guru adalah login, membuat kelas, dan membuat materi.

4.2.5 Decision Analysis

Pada tahapan ini merupakan kebutuhan bisnis dan analisis logika, tentunya sangat banyak alternatif keputusan yang dapat diambil untuk merancang sistem guna memenuhi kebutuhan dari permasalahan yang ada pada sistem pembelajaran di sekolah tersebut. Maka pada tahapan ini menggunakan *sequence diagram* untuk melihat proses apa saja, dan pesan-pesan kolaborasi antar object yang dibangun pada sistem ini. Berikut ini adalah *sequence diagram* yang ada di Aplikasi pembelajaran *Online* berbasis web.

1. *Sequence Diagram* Lihat Materi

Sequence Diagram untuk melihat materi yang diberikan oleh guru melalui aplikasi menggambarkan alur proses dalam menampilkan materi untuk siswa. Rancangan *Sequence Diagram* untuk melihat materi disajikan pada Gambar 5.

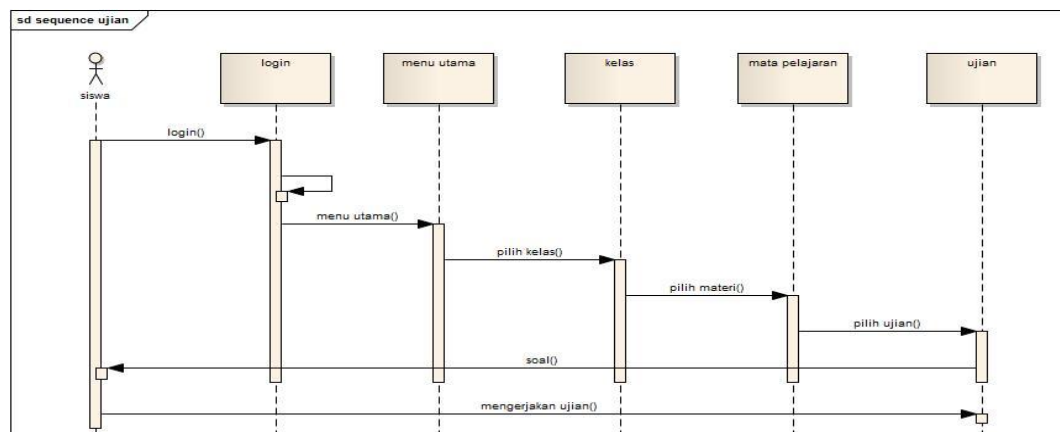


Gambar 5. *Sequence Diagram* Lihat Materi

2. *Sequence Diagram* Ujian

Sequence Diagram Ujian menggambarkan alur proses siswa untuk mengikuti ujian *online* yang telah diaktivasi oleh guru, setelah mengikuti ujian siswa akan memperoleh nilai dari ujian yang telah dikerjakan.

Rancangan *sequence diagram* ujian disajikan pada Gambar 6

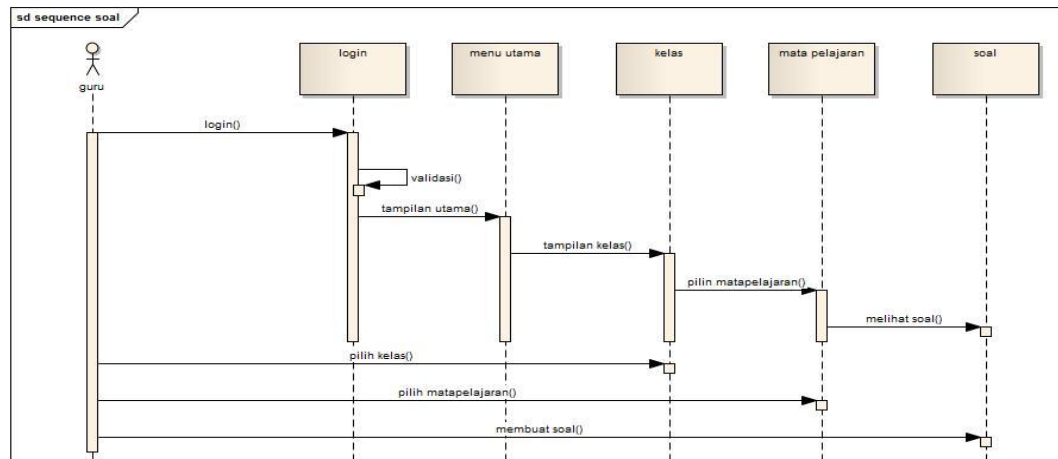


Gambar 6. *Sequence Diagram* Mengerjakan Ujian

3. *Sequence Diagram* Membuat Soal

Sequence Diagram untuk membuat soal oleh guru menggambarkan alur proses guru dalam membuat soal untuk siswa dengan cara menginputkan

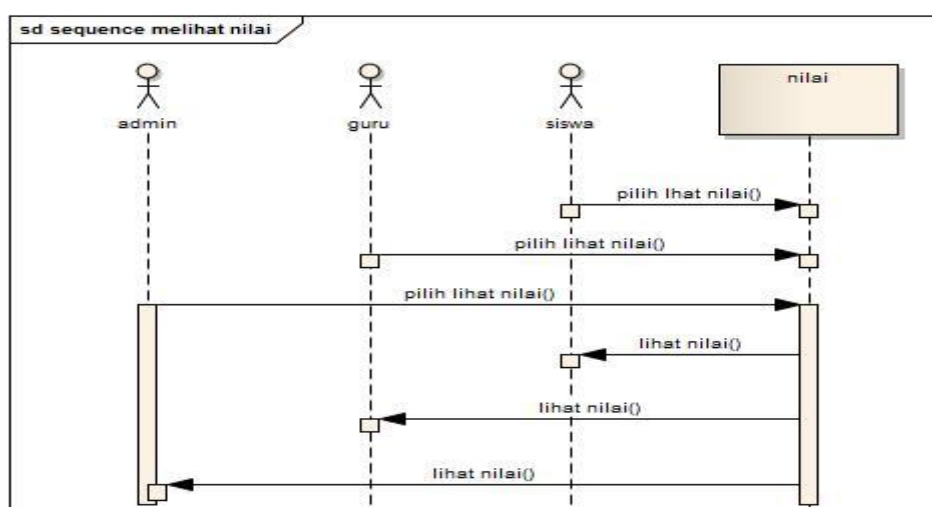
dan mengunggah soal ujian pada menu soal. Rancangan *sequence diagram* soal disajikan pada Gambar 7.



Gambar 7. *Sequence Diagram* Membuat Soal

4. Sequence Diagram Melihat Nilai

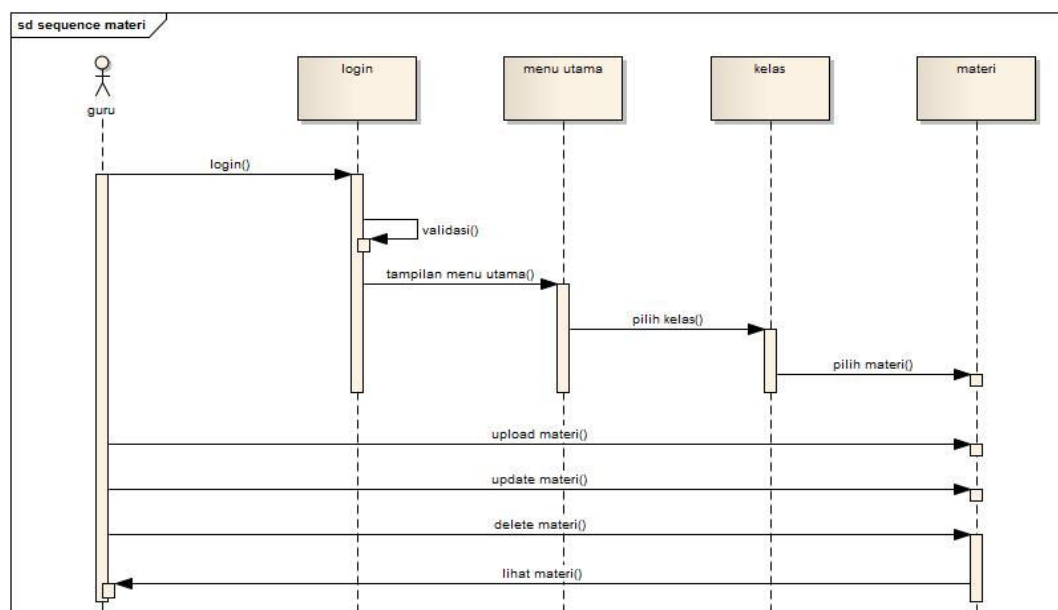
Sequence Diagram untuk melihat nilai menggambarkan alur proses admin, guru, dan siswa dalam melihat nilai tanpa harus login terlebih dahulu ke dalam sistem. Rancangan *sequence diagram* melihat nilai disajikan pada Gambar 8.



Gambar 8. *Sequence Diagram* Melihat Nilai

5. Sequence Diagram Materi

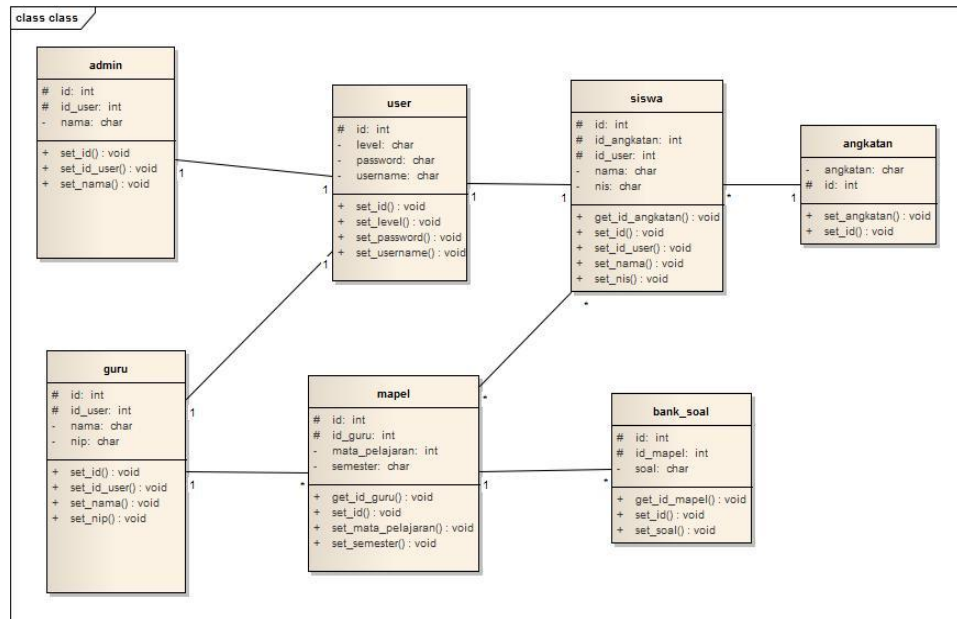
Sequence Diagram untuk materi yang diakses oleh guru menggambarkan alur proses guru dalam mengelola materi mulai dari tambah, ubah, hapus materi yang ada. Rancangan *sequence diagram* materi disajikan pada Gambar 9.



Gambar 9. *Sequence Diagram* Membuat Materi

4.2.6 Physical Design and Integration

Tahapan ini menggambarkan desain fisik kebutuhan siswa dan guru kepada model yang digambarkan secara teknis. Dengan menggunakan class diagram dapat menggambarkan struktur dan deskripsi serta hubungannya antar class. Berikut ini adalah *class diagram* yang ada di Aplikasi pembelajaran *Online* berbasis web disajikan pada Gambar 10.



Gambar 10. Class Diagram

Database yang dibuat menggunakan *software* MySQL . Pada database yang dirancang terdapat 11 tabel yaitu :

1. Database : pum_earning.sql
Tabel : User
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_user	int(11)	id user
nama	varchar(255)	nama admin

2. Database : pum_earning.sql
Tabel : Angkatan
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
angkatan	varchar(4)	angkatan berdasarkan tahun

3. *Database* : pum_elearning.sql
Tabel : Banksoal
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_guru	int(11)	id guru
id_mapel	int(11)	id mata pelajaran
soal	varchar(255)	soal-soal

4. *Database* : pum_elearning.sql
Tabel : Detil_Kelas
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_kelas	int(11)	id kelas
id_siswa	int(11)	id siswa

5. *Database* : pum_elearning.sql
Tabel : Guru
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_user	int(11)	id user
nama	varchar(255)	nama guru
nip	varchar(255)	NIP guru

6. *Database* : pum_elearning.sql
Tabel : Jawaban_banksoal
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_banksoal	int(11)	id bank soal
jawaban	varchar(255)	jawaban soal
status	int(1)	status soal

7. *Database* : pum_elearning.sql

Tabel : Kelas

Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_angkatan	int(11)	id angkatan
id_guru	int(11)	id guru
id_mapel	int(11)	id mata pelajaran
kelas	varchar(255)	kelas siswa
selesai	tinyint(1)	selesai

8. Database : pum_elearning.sql

Tabel : Mapel

Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_guru	int(11)	id guru
semester	varchar(255)	semester
mata_pelajaran	varchar(255)	mata pelajaran

9. Database : pum_elearning.sql

Tabel : Siswa

Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_user	int(11)	id user
id_angkatan	int(11)	id angkatan
nis	varchar(255)	nis siswa
nama	varchar(255)	nama siswa

10. Database : pum_elearning.sql

Tabel : Soal

Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
id_siswa	int(11)	id siswa
id_banksoal	int(11)	id banksoal
waktu_mulai	datetime	waktu soal

11. *Database* : pum_elearning.sql
Tabel : User
Primary Key : id

key	type data	deskripsi
id	int(11)	Collection Primary key
username	varchar(255)	username
password	varchar(255)	password
level	varchar(255)	level user

4.2.7 Construction and Testing

Tahapan ini merupakan tahapan pembangunan sistem dan pengujian. Pembangunan aplikasi dengan pengkodean dan kemudian melakukan pengujian menggunakan *black box testing* untuk mengetahui kelemahan sistem. Berikut ini merupakan pembuatan dan pengujian system.

4.2.7.1 Construction

Pada tahapan *construction*, akan membuat kode program. Kode program yang ditampilkan hanya kode program pada bagian *controller* saja. Diantaranya adalah :

1. Penulisan kode program pada Index.php

```
<!DOCTYPE html>
<html>
<head> <title>403 Forbidden</title>
</head>
<body>
<p>Directory access is forbidden.</p>
</body> </html>
```

2. Penulisan kode program pada Admin.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Admin extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('m_admin');
    } function index() {
        $data_admin = $this->m_admin->ambil_tabel_admin();
        $this->load->view("template/template",      array("isi"=>"admin/index",
        "data"=>$data_admin));
    }
    function tambah(){
        $this->load->view('template/template',array("isi" => "admin/tambah"));
    }
    function aksi_tambah(){
        $username = $this->input->post('username');
        $nama = $this->input->post('nama');
        $this->m_admin->tambah_admin($username,
        $nama); redirect(base_url('admin'));
    }
    function ubah($id){
        $data_admin = $this->m_admin->ambil_data_admin($id);
        $this->load->view('template/template',array("isi" => "admin/ubah", "data"
        => $data_admin));
    }
    function aksi_ubah(){
        $nama = $this->input->post('nama');
        $id = $this->input->post('id');
        $this->m_admin->ubah_admin($nama, $id);
        redirect(base_url('admin'));  }
    function hapus($id){
        $data_admin      =      $this->m_admin-
        >ambil_data_admin($id);
        $this->load->view('template/template',array("isi"      =>      "admin/hapus",
        "data" => $data_admin));
    }
    function aksi_hapus(){
        $id = $this->input->post('id');
        $this->m_admin->hapus_admin($id);
        redirect(base_url('admin'));
    }
}

```

3. Penulisan kode program pada Angkatan.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Angkatan extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('m_angkatan');
    } function index() {
        $data_angkatan = $this->m_angkatan->ambil_tabel_angkatan();
        $this->load->view("template/template",    array("isi"=>"angkatan/index",
        "data"=>$data_angkatan));
    } function tambah(){
        $this->load->view('template/template',array("isi" =>
        "angkatan/tambah"));
    } function aksi_tambah(){
        $angkatan = $this->input->post('angkatan');
        $this->m_angkatan->tambah_angkatan($angkatan);
        redirect(base_url('angkatan'));
    } function impor(){
        $this->load->view('template/template',array("isi" =>
        "angkatan/impor"));
    }
    function aksi_impor(){
        $this->load->library('excelreader/Excel_reader');
        if ($_FILES['excel']['size']==0) {
            redirect(base_url('angkatan/impor?file_kosong=1'));
        }
        $id_admin = $this->m_angkatan->ambil_id_admin($this->session->id);
        $ekstensi_diperbolehkan    = array('xls');
        $nama = $_FILES['excel']['name'];
        $x = explode('.', $nama);
        $ekstensi = strtolower(end($x));

        //end() -> akhir
        $ukuran    = $_FILES['excel']['size'];
        $file_tmp = $_FILES['excel']['tmp_name'];
        if(in_array($ekstensi, $ekstensi_diperbolehkan) === true){
            if($ukuran < 5242880){
                if(move_uploaded_file($file_tmp,
                'berkas/temp/'.$id_admin.'.xls')){
                    // echo 'FILE BERHASIL DI UPLOAD'; }else{
                    // echo 'GAGAL MENGUPLOAD FILE';
                    redirect(base_url('angkatan/impor?upload_gagal=1'));
                } }else{
                    redirect(base_url('angkatan/impor?file_kebesaran=1'));
                    // echo 'UKURAN FILE TERLALU BESAR';
                } }else{
                    redirect(base_url('angkatan/impor?ekstensi_salah=1'));
                }
            }
        }
    }
}

```

```

// echo 'EKSTENSI FILE YANG DI UPLOAD TIDAK DI
PERBOLEHKAN';

        }$this->excel_reader-
>setOutputEncoding('230787');
        $this->excel_reader-
>read('berkas/temp/'.$_id_admin.'.xls');

        $data = $this->excel_reader->sheets[0];
        // echo $data['cells'][3][1];
//['cells'][bawah][samping]
        // echo "<br>";
        // echo $data['numRows']; //jumlah baris (bawah)

        for ($i=4; $i <= $data['numRows']; $i++) {
            // echo $data['cells'][$i][1] . "<br>";
            $this->m_angkatan-
>tambah_angkatan($data['cells'][$i][1]);
        }
redirect(base_url('angkatan')); }
function ubah($_id){
    $data_angkatan = $this->m_angkatan-
>ambil_data_angkatan($_id);

    $this->load->view('template/template',array("isi" =>
"angkatan/ubah", "data" => $data_angkatan));
    }function aksi_ubah(){
        $angkatan = $this->input->post('angkatan');
        $id = $this->input->post('id');

        $this->m_angkatan->ubah_angkatan($angkatan,
$id);

        redirect(base_url('angkatan'));
    } function hapus($_id){
        $data_angkatan = $this->m_angkatan-
>ambil_data_angkatan($_id);
$this->load->view('template/template',array("isi" => "angkatan/hapus",
"data" => $data_angkatan));
    } function aksi_hapus($_id){
        $this->m_angkatan->hapus_angkatan($_id);
        redirect(base_url('angkatan'));
    } }

```

4. Penulisan kode program pada Guru.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Guru extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('m_guru');
    } function index() {
        $data_guru = $this->m_guru->ambil_tabel_guru();
        $this->load->view("template/template", array("isi"=>"guru/index",
        "data"=>$data_guru));
    } function tambah(){
        $this->load->view('template/template',array("isi" =>
        "guru/tambah"));
    }

    function aksi_tambah(){
        $nip = $this->input->post('nip');
        $nama = $this->input->post('nama');

        $this->m_guru->tambah_guru($nip, $nama);

        redirect(base_url('guru'));
    } function ubah($id){
        $data_guru = $this->m_guru->ambil_data_guru($id);

        $this->load->view('template/template',array("isi" =>
        "guru/ubah", "data" => $data_guru));
    }

    function aksi_ubah(){
        $nama = $this->input->post('nama');
        $id = $this->input->post('id');
        $this->m_guru->ubah_guru($nama, $id);
        redirect(base_url('guru'));
    } function hapus($id){
        $data_guru = $this->m_guru->ambil_data_guru($id);
        $this->load->view('template/template',array("isi" => "guru/hapus", "data"
        => $data_guru));
    }

    function aksi_hapus($id){
        $this->m_guru->hapus_guru($id);

        redirect(base_url('guru'));
    }

    function impor(){
        $this->load->view('template/template',array("isi" =>
        "guru/impor"));
    }
}

```

```

    }

    function aksi_impор(){
        $this->load->library('excelreader/Excel_reader');
        if ($_FILES['excel']['size']==0) {

            redirect(base_url('guru/impор?file_kosong=1'));
        }
        $id_admin = $this->m_guru->ambil_id_admin($this->session->id);
        $ekstensi_diperbolehkan = array('xls');
        $nama = $_FILES['excel']['name'];
        $x = explode('.', $nama);
        $ekstensi = strtolower(end($x));
        //awal
        //tengah
        //akhir
        //end() -> akhir
        $ukuran = $_FILES['excel']['size'];
        $file_tmp = $_FILES['excel']['tmp_name'];

        if(in_array($ekstensi,
        $ekstensi_diperbolehkan) === true){
            if($ukuran < 5242880){

                if(move_uploaded_file($file_tmp,
                'berkas/temp/'.$id_admin.'.xls')){
                    // echo 'FILE BERHASIL DI UPLOAD';
                }else{
                    // echo 'GAGAL MENGUPLOAD FILE';

                    redirect(base_url('guru/impор?upload_gagal=1'));
                }
            }else{
                redirect(base_url('guru/impор?file_kebesaran=1'));
                // echo 'UKURAN FILE
                TERLALU BESAR';
            }
        }else{
            redirect(base_url('guru/impор?ekstensi_salah=1'));
            // echo 'EKSTENSI FILE YANG DI
            UPLOAD TIDAK DI PERBOLEHKAN';
        }
        $this->excel_reader->setOutputEncoding('230787');
        $this->excel_reader-
        >read('berkas/temp/'.$id_admin.'.xls');

        $data = $this->excel_reader->sheets[0];
        // echo $data['cells'][3][1]; //['cells'][bawah][samping]
    }

```

```

        // echo "<br>";
        // echo $data['numRows']; //jumlah baris (bawah)
for ($i=4; $i <= $data['numRows']; $i++) {
        // echo $data['cells'][$i][1] . "<br>";
        $this->m_guru-
>tambah_guru($data['cells'][$i][1],$data['cells'][$i][2]);
        }redirect(base_url('guru'));
    } }

```

5. Penulisan kode program pada login.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Login extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('m_login');
    } function index() {
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $total = $this->m_login-
>cek_user_password($username, $password);
        if ($total == 1) {
            $data_user = $this->m_login-
>ambil_data_user($username, $password);
            $array_data_user = array(
                'id' => $data_user['id'],
                'nama' => $data_user['nama'],
                'username' => $data_user['username'],
                'level' => $data_user['level'],
                'login' => true
            );
            $this->session-
>set_userdata($array_data_user);
            redirect(base_url());
        } else {
            redirect(base_url("?valid=0"));
        } }
    } }

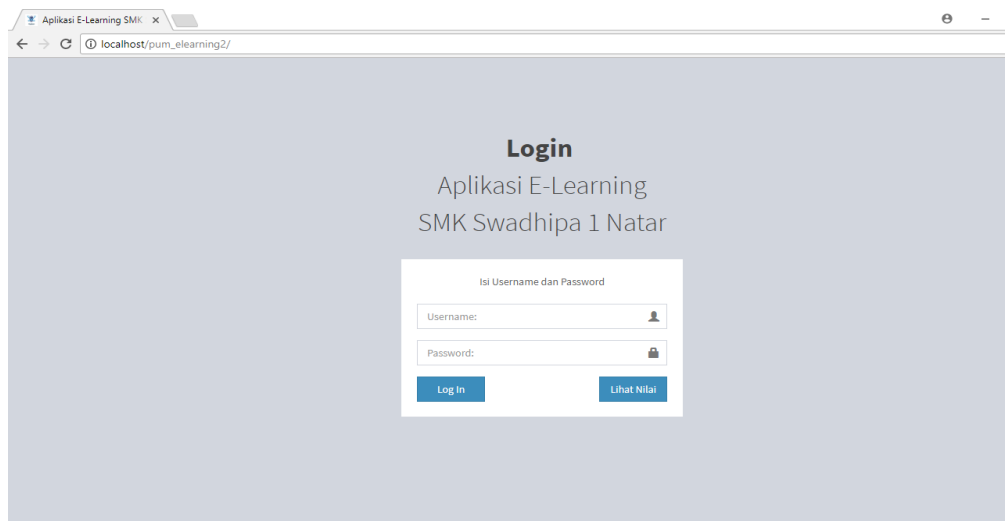
```


4.2.7.2 Tampilan Aplikasi

Pada tahapan ini akan ditampilkan beberapa tampilan program yang dibuat berdasarkan rancangan-rancangan yang telah diusulkan pada tahap sebelumnya.

1. Tampilan *form login*

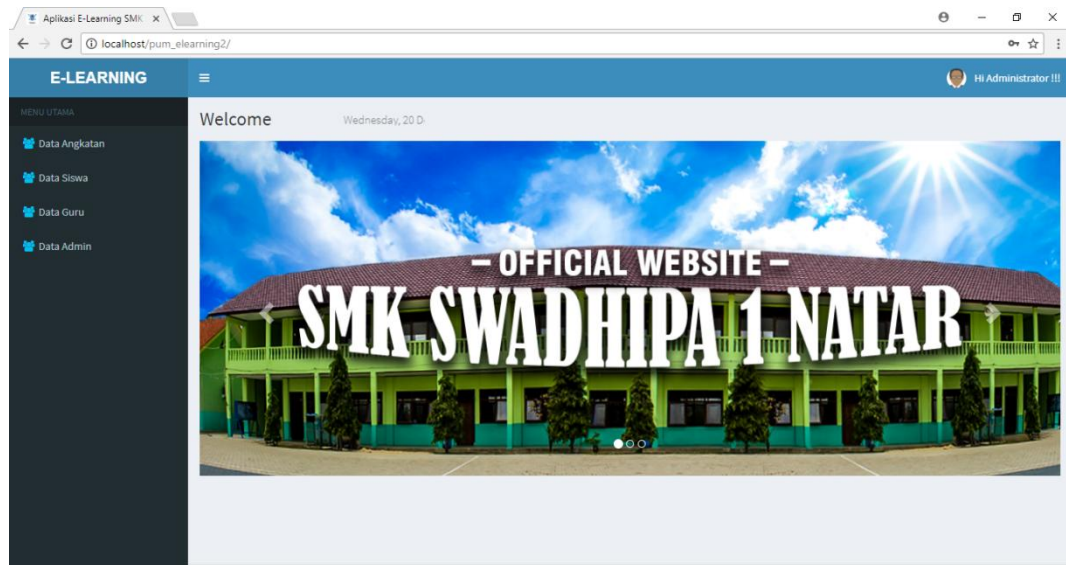
Form login berfungsi untuk mendapatkan hak akses, untuk dapat mengakses aplikasi tersebut admin, guru dan siswa harus memasukkan *user* dan *password*. Seperti disajikan pada Gambar 11.



Gambar 11. Tampilan *Form Login* untuk admin, siswa dan guru.

2. Tampilan Halaman Utama Admin

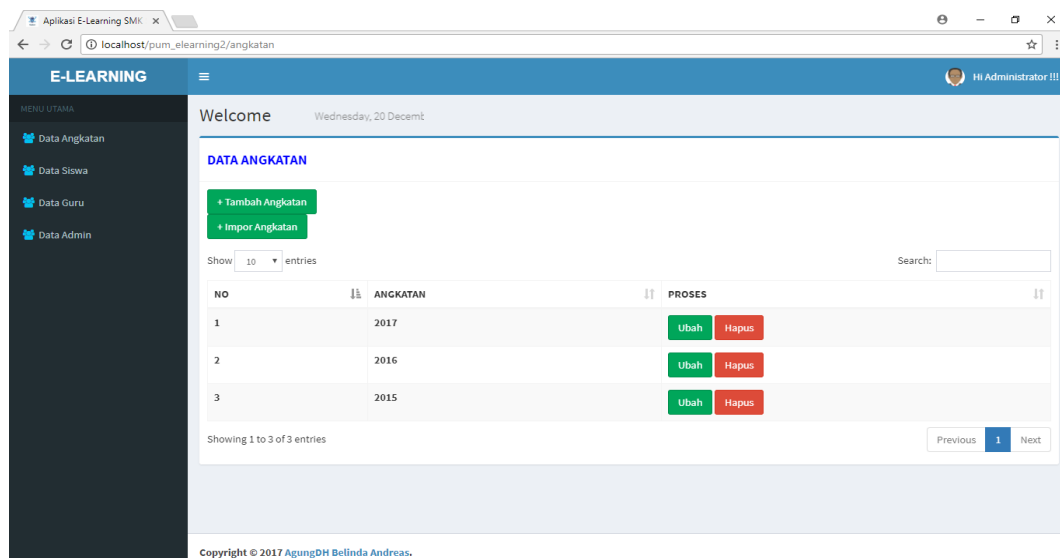
Halaman utama admin menampilkan menu yang dapat dilakukan oleh admin yaitu menu *home*, data angakata, data siswa, data guru dan data admin. Seperti disajikan pada Gambar 12.



Gambar 12. Tampilan Halaman Utama Admin

3. Tampilan Halaman Data Angkatan

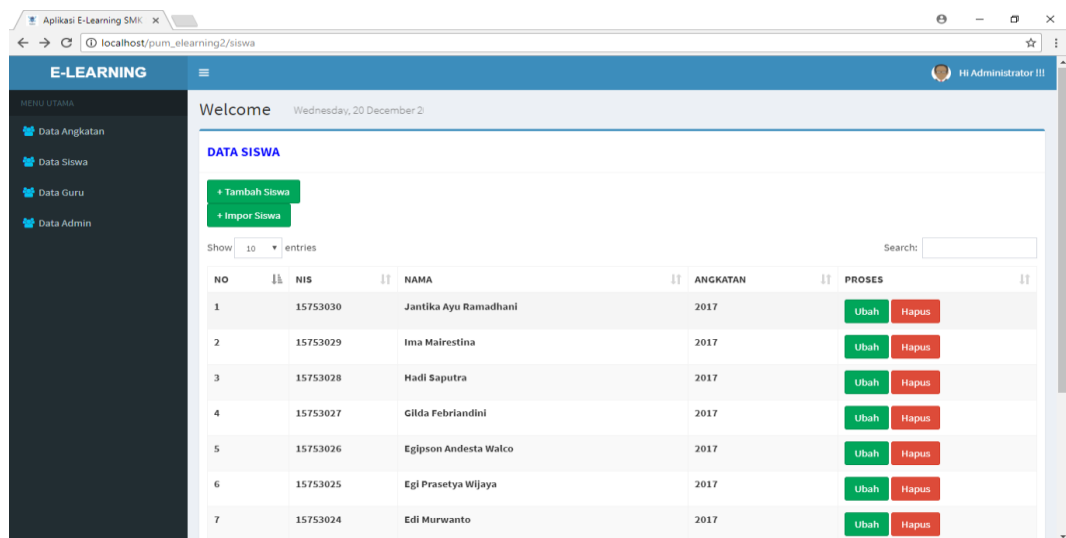
Tampilan halaman data angkatan menampilkan *action* ubah dan hapus data pada proses. Kemudian menambah angkatan serta *import* angkatan menggunakan *file*. Seperti disajikan pada Gambar 13



Gambar 13. Tampilan Halaman Data Angkatan

4. Tampilan Halaman Data Siswa

Tampilan halaman data siswa menampilkan *action* ubah dan hapus data pada proses. Kemudian menambah angkatan serta *import* siswa menggunakan *file*. Seperti disajikan pada Gambar 14.

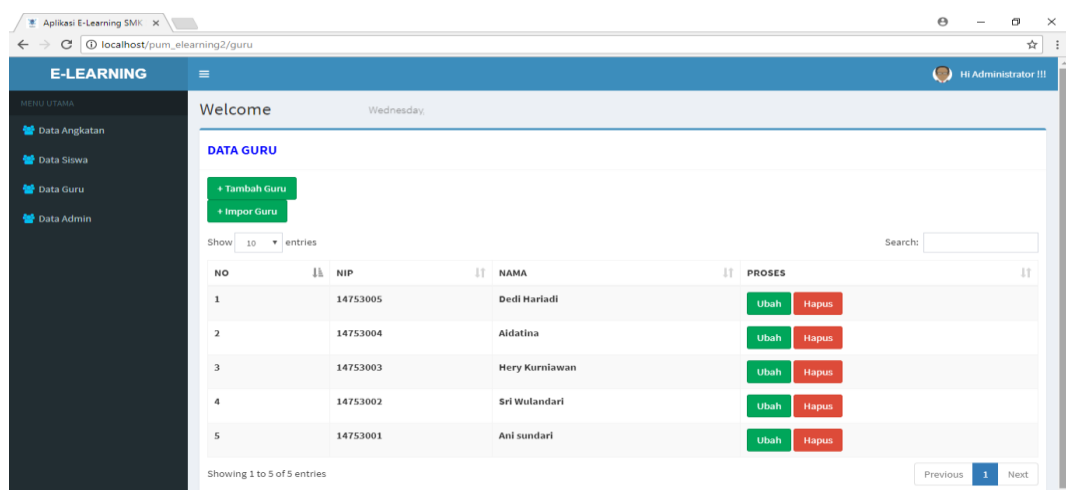


NO	NIS	NAMA	ANGKATAN	PROSES
1	15753030	Jantika Ayu Ramadhani	2017	Ubah Hapus
2	15753029	Ima Mairestina	2017	Ubah Hapus
3	15753028	Hadli Saputra	2017	Ubah Hapus
4	15753027	Gilda Febriandini	2017	Ubah Hapus
5	15753026	Egipson Andesta Walco	2017	Ubah Hapus
6	15753025	Egi Prasetya Wijaya	2017	Ubah Hapus
7	15753024	Edi Murwanto	2017	Ubah Hapus

Gambar 14. Tampilan Halaman Data siswa

5. Tampilan Halaman Data Guru

Tampilan halaman data guru menampilkan *action* ubah dan hapus data pada proses. Kemudian menambah angkatan serta *import* siswa menggunakan *file*. Seperti disajikan pada Gambar 15.

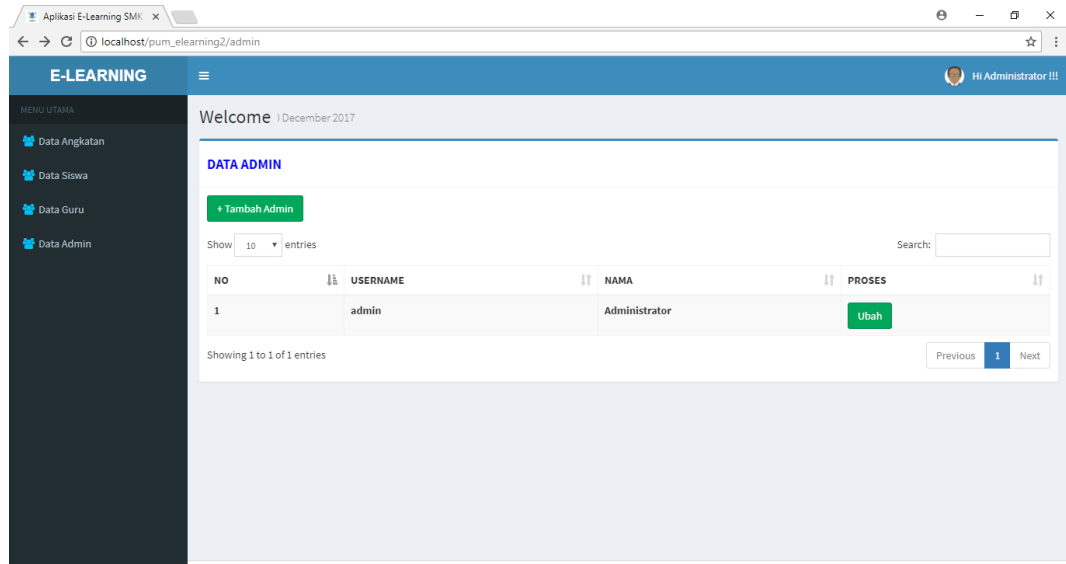


NO	NIP	NAMA	PROSES
1	14753005	Dedi Hariadi	Ubah Hapus
2	14753004	Aldatina	Ubah Hapus
3	14753003	Hery Kurniawan	Ubah Hapus
4	14753002	Sri Wulandari	Ubah Hapus
5	14753001	Ani sundari	Ubah Hapus

Gambar 15. Tampilan halaman data guru

6. Tampilan Halaman Data Admin

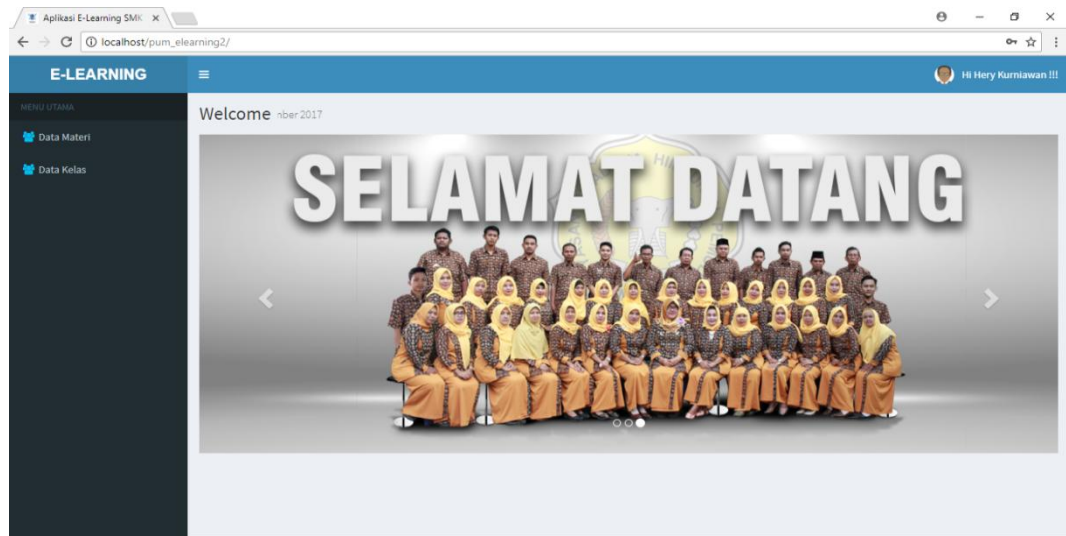
Tampilan halaman data admin menampilkan *action* ubah data pada proses. Kemudian menambah data admin. Seperti disajikan pada Gambar 16.



Gambar 16. Tampilan Halaman Data Admin

7. Tampilan Halaman Utama Guru

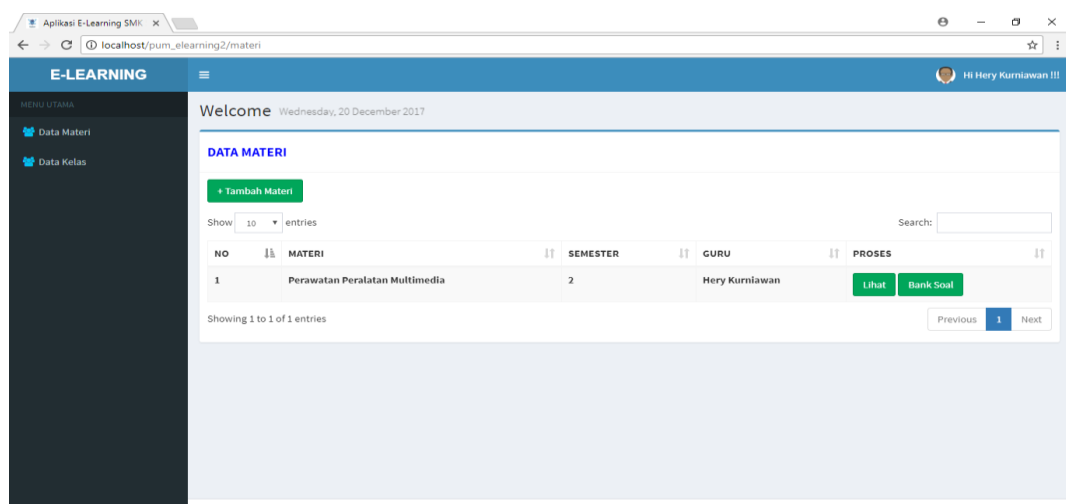
Halaman utama guru menampilkan menu yang dapat dilakukan oleh admin yaitu menu *home*, data materi, dan data kelas. Seperti disajikan pada Gambar 17.



Gambar 17. Halaman utama guru

8. Tampilan halaman data materi

Tampilan halaman data angkatan menampilkan *action* ubah dan hapus data pada proses. Kemudian menambah materi serta *import* materi menggunakan *file*. Seperti disajikan pada Gambar 18.



Gambar 18. Tampilan halaman data materi

9. Tampilan Materi

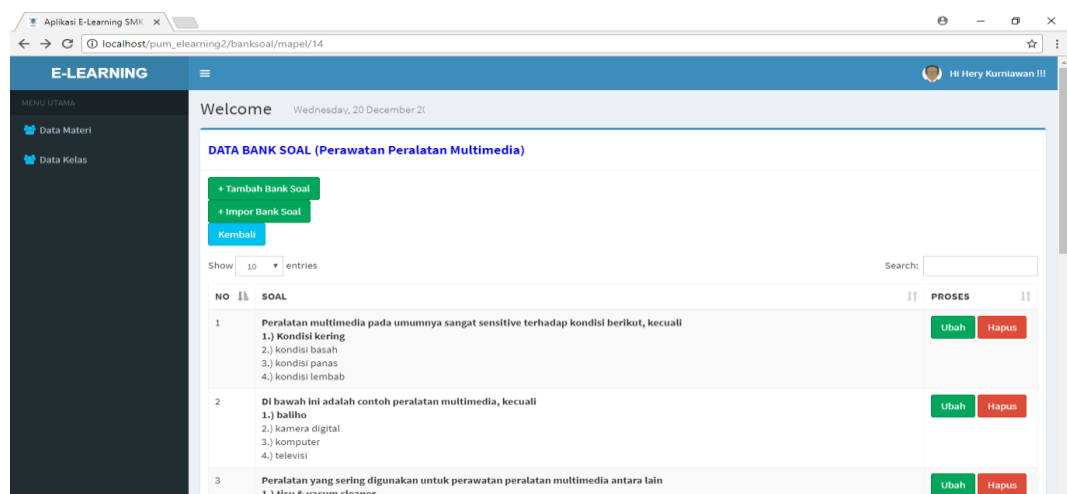
Tampilan halaman materi pada guru, guru dapat menghapus materi dan mengubah materi. Seperti disajikan pada Gambar 19.



Gambar 19. Tampilan halaman materi

10. Tampilan Data Bank Soal

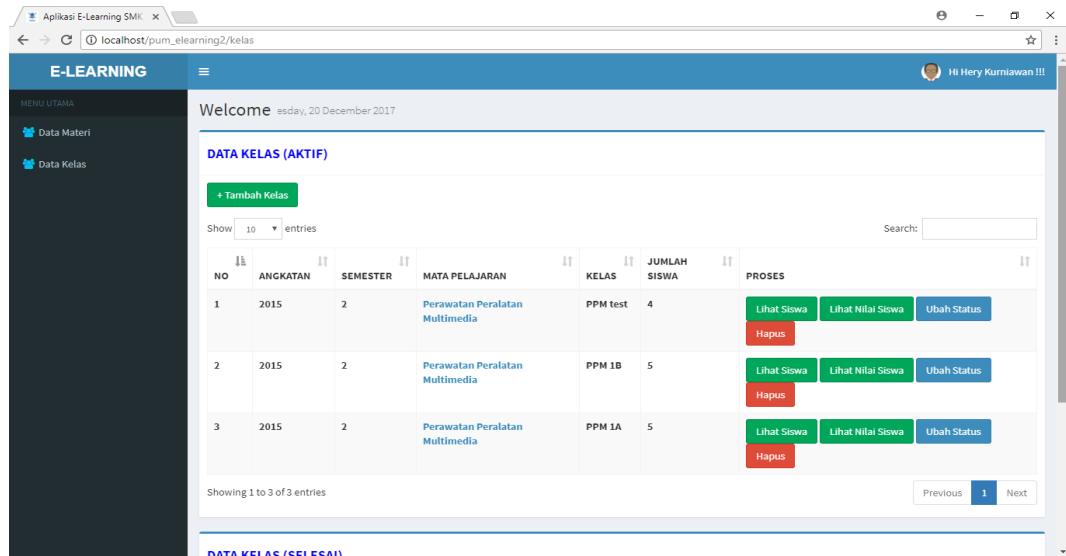
Tampilan halaman Bank Soal pada user guru, guru dapat menambah menghapus, dan mengubah soal serta mengimport soal dengan menggunakan file. Seperti disajikan pada Gambar 20.



Gambar 20. Tampilan Data Bank Soal

11. Tampilan Data Kelas

Tampilan halaman data kelas pada user guru, guru dapat menambah kelas dengan mengimport *file*, kemudian melihat daftar siswa dan nilai siswa, menghapus, dan mengubah status kelas. Seperti disajikan pada Gambar 21.



Apikasi E-Learning SMK

localhost/pum_elearning2/kelas

E-LEARNING

Welcome esday, 20 December 2017

DATA KELAS (AKTIF)

+ Tambah Kelas

Show 10 entries

Search:

NO	ANGKATAN	SEMESTER	MATA PELAJARAN	KELAS	JUMLAH SISWA	PROSES
1	2015	2	Perawatan Peralatan Multimedia	PPM test	4	Lihat Siswa Lihat Nilai Siswa Ubah Status Hapus
2	2015	2	Perawatan Peralatan Multimedia	PPM 1B	5	Lihat Siswa Lihat Nilai Siswa Ubah Status Hapus
3	2015	2	Perawatan Peralatan Multimedia	PPM 1A	5	Lihat Siswa Lihat Nilai Siswa Ubah Status Hapus

Showing 1 to 3 of 3 entries

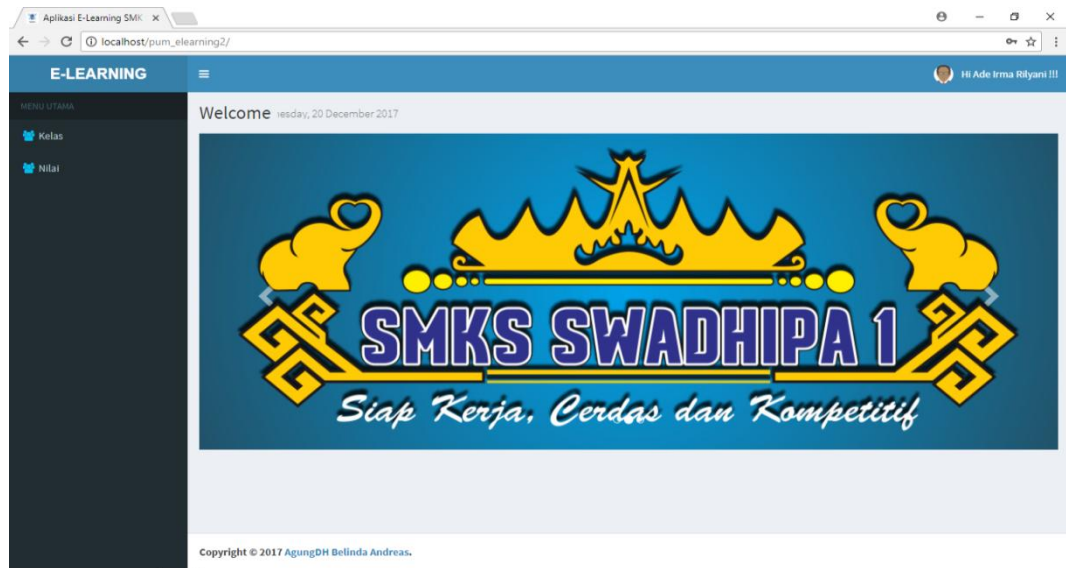
Previous 1 Next

DATA KELAS (SELESAI)

Gambar 21. Tampilan Data Kelas pada user guru

12. Tampilan Halaman Utama Siswa

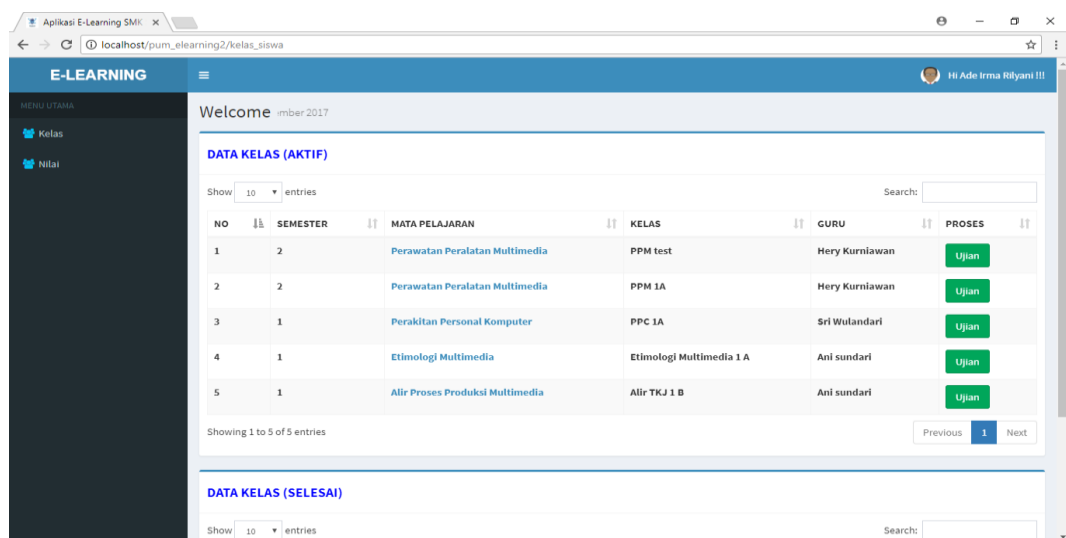
Halaman utama siswa menampilkan menu yang dapat dilakukan oleh siswa yaitu menu *home*, materi, dan kelas. Seperti disajikan pada Gambar 22.



Gambar 22. Halaman utama pada siswa

13. Tampilan Halaman Kelas siswa

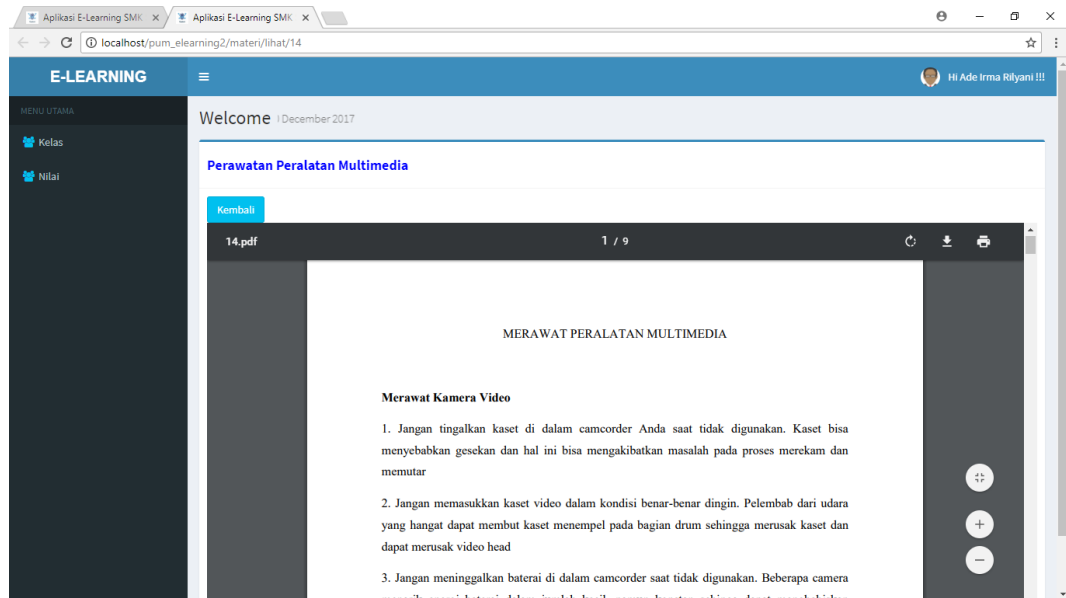
Tampilan menu kelas siswa, siswa dapat mengakses materi dan mengikuti ujian yang telah di aktivasi. Seperti disajikan pada Gambar 23.



Gambar 23. Tampilan halaman kelas aktif untuk siswa

14. Tampilan Akses Materi Pelajaran

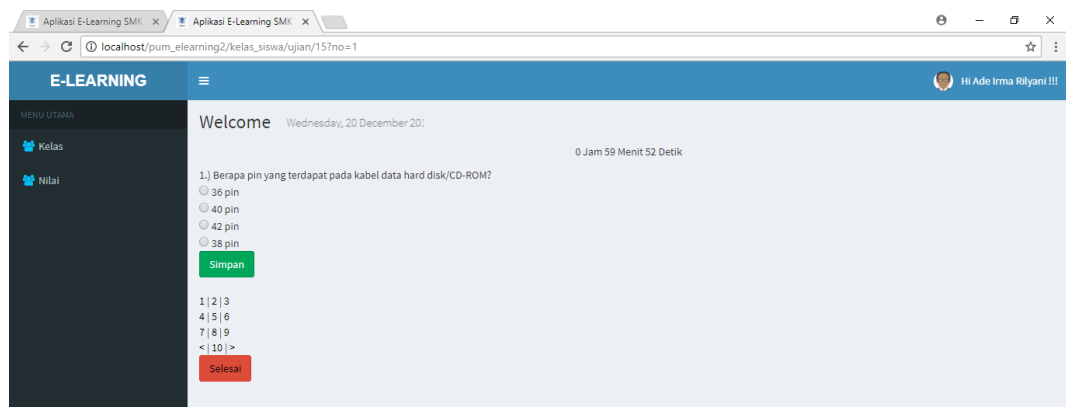
Tampilan halaman menu materi yang di akses oleh siswa. Seperti disajikan pada Gambar 24.



Gambar 24. Tampilan materi untuk siswa

15. Tampilan Ujian Siswa

Tampilan halaman ujian siswa, terdiri dari soal-soal dan pilihan jawaban. Seperti disajikan pada Gambar 25.



Gambar 25. Tampilan Ujian untuk siswa

16. Tampilan Nilai yang diperoleh dari ujian

Halaman tampilan nilai siswa yang diperoleh dari ujian, dapat dilihat di kolom nilai. Seperti disajikan pada Gambar 26.

NO	WAKTU	NILAI
1	2017-12-20 07:43:50	40.00

Gambar 26. Tampilan nilai siswa

4.2.7.3 Pengujian Sistem

Tahap pengujian sistem secara lengkap dilakukan untuk menjamin bahwa syarat dan spesifikasi sistem telah terpenuhi berdasarkan persyaratan-persyaratan yang didapat pada tahap selanjutnya.

4.2.7.3.1 Metode Pengujian

Metode pengujian yang digunakan pada “Aplikasi Pembelajaran *Online* Berbasis Web” adalah *Black box testing* yaitu dengan data-data yang digunakan dalam pengujian aplikasi adalah berdasarkan prinsip *Error Guessing*.

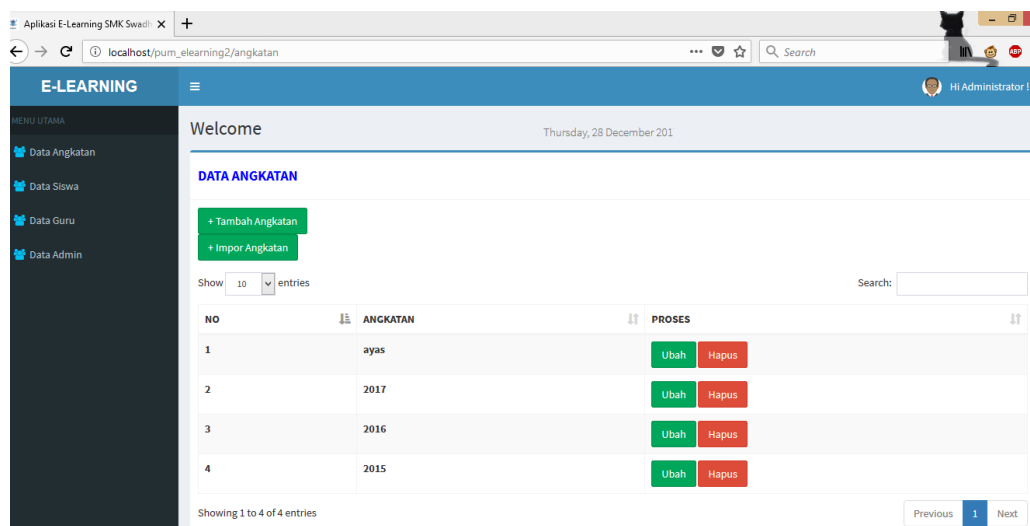
4.2.7.3.2 Hal-hal yang diuji

Error Guessing adalah pengujian berdasarkan intuisi dari sisi penguji. Dalam hal ini *error guessing* dimaksudkan untuk mengetahui kemungkinan *error*

yang terjadi ketika program di uji coba. Hal-hal yang diuji dalam aplikasi yang berjudul “Aplikasi Pembelajaran *Online* Berbasis Web” ini adalah sebagai berikut :

1. Pengujian Untuk Penambahan Data Angkatan

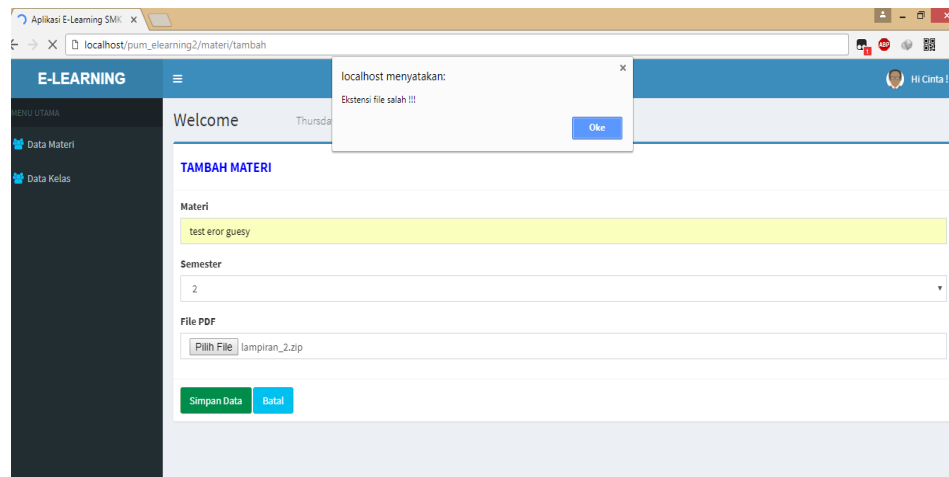
Pada pengujian ini penguji menginputkan data angkatan dengan huruf, dengan diinputkan karakter huruf seharusnya tidak dapat masuk ke dalam *database*. Hasil pengujian disajikan pada Gambar 27.



Gambar 27. Pengujian tambah angkatan dengan karakter huruf

2. Pengujian Untuk *Format File*

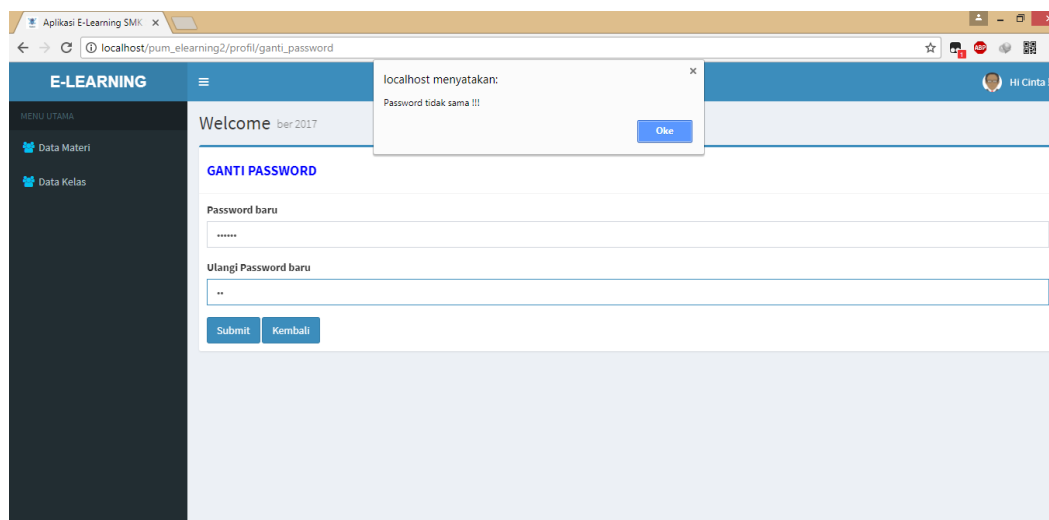
Pada pengujian yang kedua ini, penguji melakukan pengujian dengan mencoba *format file* yang tidak sesuai dengan *format* yang sudah ditentukan. Hasil pengujian disajikan pada Gambar 28.



Gambar 28. pengujian memasukan *format file* yang tidak sesuai.

3. Pengujian Untuk *Password*

Pada pengujian yang ketiga, penguji mencoba mengubah *password* pada *user*, namun *password* yang dimasukan pada konfirmasi ulang *password* baru tidak sama dengan *password* yang dimasukkan *password* pertama. Hasil pengujian disajikan pada Gambar 29.



Gambar 29. Pengujian *Password*

4.2.7.3.3 Hasil Pengujian

Hasil pengujian dengan *error guessing* disajikan pada table 2.

Tabel 2. Tabel Hasil Pengujian

Input	Hasil Uji Yang diharapkan	Output	Kesimpulan
Pengujian Data Angkatan	Karakter huruf tidak dapat masuk kedalam data angkatan	Angkatan dapat diinputkan dengan karakter huruf	Hasil pengujian tidak dapat memenuhi harapan
Pengujian <i>Format File</i>	Dapat menampilkan <i>pop-up</i> bahwa ekstensi <i>file</i> yang diinputkan tidak sesuai	<i>File</i> tidak dapat diunggah	Hasil pengujian berhasil
Pengujian dengan <i>Password</i> secara acak	Dapat menampilkan <i>pop-up</i> bahwa <i>password</i> yang diinputkan tidak sama	<i>Password</i> tidak berhasil diinputkan	Hasil Pengujian berhasil

4.2.8 Installation and Delivery

Tahapan ini merupakan proses terjadinya perubahan dari sistem yang lama, yakni sistem pembelajaran yang masih menggunakan dokumen, dan akan berubah ke sistem yang baru dan membantu pengguna dalam pelatihan penggunaan sistem yang baru. Untuk perubahan sistem pembelajaran ini menggunakan perencanaan sistem yang matang agar berjalan dengan baik.

BAB V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat dari sistem pembelajaran di SMK SWADHIPA 1 adalah dihasilkannya suatu aplikasi untuk memudahkan guru dan siswa dalam memberi materi dan mengakses materi pelajaran di sekolah kemudian siswa dapat mengikuti ujian secara *online*, lalu guru dapat menyimpan soal-soal ujian pada bank soal yang telah disediakan, dan memudahkan baik guru ataupun siswa untuk menyimpan materi berbentuk *softcopy*.

5.2 Saran

Saran yang dapat diberikan pada pengguna aplikasi pembelajaran *online* berbasis *website* ini adalah diharapkan *website* ini dapat dikembangkan dan dapat menambahkan fitur materi berupa materi berupa gambar atau animasi, dan simulasi praktikum, serta adanya pesan notifikasi untuk mengetahui aktivasi ujian.

DAFTAR PUSTAKA

- Hendrayudi. (2009). *VB 2008 untuk Berbagai Keperluan Programming*. Jakarta: PT. Elex Media Komputindo.
- Rosa, & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Setiawan, F. A. (2012). *Pemrograman Internet*. Yogyakarta: Graha Ilmu.
- Yuhefizar. (2013). *Cara Mudah dan Murah Membangun dan Mengelola Website*. Yogyakarta: Graha Ilmu.
- Prayitno, I. (2010). *Kupas Tuntas Malware*. Jakarta: PT. Elex Media Komputindo.
- Supardi, Y. (2015). *Buku Kuliah Web Programming 1*. Jakarta: @Dotakom Lintas Buana.
- Anisya. (2013, Agustus). Aplikasi Sistem Database Rumah Sakit Terpusat. *Jurnal Momentum*, 15, 10.
- Sulistiyawan, Rubianto, & Rahmad, S. (2008). *Modifikasi Blog Multiply dengan CSS*. Jakarta: PT. Elex Media Komputindo.
- Rahman, S. (2013). *Cara Gampang Bikin CMS Tanpa Ngoding*. Jakarta: Mediakita.
- Rozi, Z. A. (2015). *Bootstratp Design Framework*. Jakarta: PT. Elex Media Komputindo.
- Wibowo, K. (2015, Desember). Analisa Konsep Object Oriented Programming Pada Bahasa Pemrograman PHP. *JURNAL KHATULISTIWA INFORMATIKA*, 3, 1.
- Pressman, R. S. (2010). *Rekayasa Perangkat Lunak - Buku Satu, Pendekatan Praktisi (Edisi 7)*. Yogyakarta: Penerbit ANDI.
- Jeffery L. Whitter, L. D. (2004). *Metode Desain & Analisis Sistem*. Yogyakarta: Penerbit ANDI.
- Sagala, Syaiful (2010). *Konsep dan Makna Pembelajaran*. Bandung: Pei Alfabeta.
- Masruri, M. H. (2015). *Membangun SMS Gateway dengan Gammu dan Kalkun*. Jakarta: PT. Elex Media Komputindo.

Handayani, H. (2012). XAMPP. *XAMPP*, 4. Retrieved from imulti.org.

Winarno, E., Zaki, A., & Community, S. (2014). *24 Jam Belajar PHP*. Jakarta: PT. Elex Media Komputindo.