

Sistem Informasi Ujian Online Dengan Mengimplementasikan Arsitektur HMVC (*Hierarchical Model, View, Controller*) (Studi Kasus: SMK Muhammadiyah 8 Siliragung Banyuwangi)

Roudla Tul Jannah^{*1}, Aminudin², Ilyas Nuryasin³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

roudla_437174@webmail.umm.ac.id^{*1}, aminudin2008@umm.ac.id², ilyas@umm.ac.id³

Abstrak

Seiring dengan perkembangan teknologi saat ini, banyak instansi pendidikan yang mulai untuk memanfaatkan teknologi, salah satunya SMK Muhammadiyah 8 Siliragung Banyuwangi. SMK Muhammadiyah 8 Siliragung Banyuwangi merupakan satu dari sekian instansi pendidikan yang mulai harus memanfaatkan teknologi guna meningkatkan mutu baik dalam prestasi maupun pelayanan, salah satunya dalam kegiatan ujian di Sekolah. Maka perlu dibangunnya sebuah sistem informasi untuk menerapkan sistem pelaksanaan ujian yang secara online. Pada penelitian ini dilakukan pembangunan sistem informasi ujian online dengan menggunakan framework codeigniter yang mengimplementasi arsitektur HMVC (*Hierarchical Model, View, Controller*). Membangun sistem dengan menggunakan pola HMVC untuk menghasilkan sistem yang lebih modular sehingga memberi kemudahan dalam proses pengembangan dan lebih terfokus pada masing-masing modul yang sedang dikerjakan, dan memudahkan dalam penambahan modul atau penggunaan kembali fungsi pada proyek selanjutnya. Sistem dibangun dengan melalui pengujian *integration testing*, *blackbox testing*, dan *UAT (User Accepted Test)*. Hasil dari pengujian menunjukkan bahwa tiap fungsi sistem terintegrasi dengan baik, dan sistem sesuai dengan kebutuhan user.

Kata Kunci: HMVC (*Hierarchical Model, View, Controller*), Framework CodeIgniter, Integration Testing, Blackbox Testing, UAT (*User Accepted Test*)

Abstract

Along with current technological developments. There are several educational institutions that start to utilize technology, one of which is Vocational high school of Muhammadiyah 8 Siliragung Banyuwangi. Vocational high school of Muhammadiyah 8 Siliragung Banyuwangi is one of educational institutions that must start using technology to improve the quality of both achievement and services, one of which is in the examination activities at school. Therefore, it is necessary to build an information system to implement an online test. This research was carried out the development of online exam information system. Framework codeigniter was used as away to implementing HMVC (*Hierarchical Model, View, Controller*) architecture. In building system using HMVC pattern to produce a more modular system can simplify the development process, be more focused on each module being worked on and easy to add a modul or reuse function in future project. This system built by *integration testing*, *blackbox testing*, dan *UAT (User Accepted Test)*. The result of the test showed that every function of the system is well integrated, and the system complies with the user's needs.

Keywords: HMVC (*Hierarchical Model, View, Controller*), Framework CodeIgniter, Integration Testing, Blackbox Testing, UAT (*User Accepted Test*)

1. Pendahuluan

Suatu kegiatan belajar yang selalu dilaksanakan di setiap instansi pendidikan yaitu kegiatan ujian. Kegiatan ini dilaksanakan di akhir proses pembelajaran siswa, guna untuk mengukur sejauh mana siswa memahami materi yang telah diberikan selama proses belajar.

Seiring dengan perkembangan teknologi saat ini, banyak instansi pendidikan yang mulai dituntut untuk memanfaatkan teknologi dalam proses belajar salah satunya dalam pelaksanaan ujian. Salah satunya adalah SMK Muhammadiyah 8 Siliragung Banyuwangi, sekolah ini mulai dituntut untuk memanfaatkan teknologi guna meningkatkan mutu baik dalam prestasi maupun pelayanan. Sehingga SMK Muhammadiyah 8 Siliragung Banyuwangi mulai menerapkan sistem

pembelajaran dan pelayanan sekolah secara online, salah satunya pada pelaksanaan ujian. Maka perlu dibangunnya sebuah sistem informasi guna menerapkan sistem pelaksanaan ujian secara online.

Pada penelitian ini dibangun sebuah sistem informasi ujian online yang menggunakan konsep HMVC (Hierarchical Model, View, Controller). Konsep HMVC lebih bermodular dengan memetakan folder untuk masing-masing layer sehingga memberikan kemudahan dalam proses pengembangan dan lebih terfokus pada masing-masing modul yang sedang dikerjakan, memudahkan dalam penambahan modul atau penggunaan kembali fungsi pada proyek selanjutnya. Selain itu, memungkinkan untuk dilakukan pengintegrasian atau penggabungan sistem, sehingga dengan menggunakan konsep HMVC mempermudah dalam mengintegrasikan atau menggabungkan kedua sistem karena sifatnya yang tidak ketergantungan antar kode.

Konsep HMVC mulai sering diterapkan dalam beberapa pengembangan sistem, salah satunya pada pengembangan sistem informasi rumah sakit. Sistem tersebut dibangun menggunakan bahasa pemrograman PHP dengan framework CodeIgniter, menerapkan konsep HMVC guna untuk menghasilkan sistem yang modular sehingga memudahkan dalam perancangan dan pengembangannya [1]. Selain itu konsep HMVC juga digunakan pada pengembangan aplikasi clickost berbasis web, aplikasi ini dibangun menggunakan bahasa pemrograman PHP dengan framework CodeIgniter dan database MySQL [2].

Banyaknya pengembang yang mulai menggunakan konsep HMVC pada proses pembangunan sistem, maka pada penelitian ini membangun sistem informasi ujian online dengan menerapkan konsep HMVC, guna membantu proses pembangunan menjadi lebih mudah karena sistem yang dihasilkan lebih modular. Dengan adanya sistem informasi ujian online ini diharapkan dapat meningkatkan kualitas proses belajar pada SMK Muhammadiyah 8 Siliragung Banyuwangi.

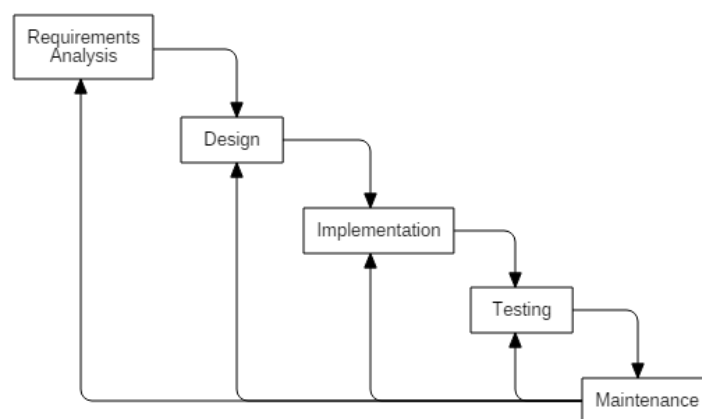
2. Metode Penelitian

2.1 Identifikasi Masalah dan Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi permasalahan tentang bisnis proses pelaksanaan ujian di sekolah SMK 8 Muhammadiyah Siliragung Banyuwangi. Setelah informasi permasalahan didapatkan, kemudian akan dilakukan studi literatur dengan mengumpulkan beberapa referensi seperti jurnal, penelitian sebelumnya, dan dokumen untuk mencari solusi atau metode yang dapat menyelesaikan permasalahan yang terjadi.

2.2 Perancangan Sistem

Pada tahap perancangan sistem dilakukan dengan menggunakan metode *waterfall*. Metode *waterfall* disebut juga *classic life cycle* [3]. Pada tahun 1970 metode *waterfall* mulai diperkenalkan oleh W Royce [4] Gambar 1 berikut tahapan-tahapan dari metode *waterfall*.



Gambar 1 Tahapan Metode Waterfall [3]

2.2.1 Requirements Analysis

Pengumpulan data dilakukan dengan menggunakan teknik *non-participant observer role*, dimana tidak terlibat secara langsung dengan proses pelaksanaan ujian hanya sebagai pengamat semata. Dalam proses pelaksanaan ujian beberapa aktor yang terlibat sekaligus aktifitas yang dilakukan berikut penjabarannya pada Tabel 1.

Tabel 1. Identifikasi Aktor

Aktor	Deskripsi
Guru	<p>Aktor yang bertugas untuk :</p> <ul style="list-style-type: none"> • Mengaktifkan ujian sesuai dengan yang dijadwalkan, • Mengawasi peserta ujian sedang mengerjakan (absensi), • Melihat list siswa/peserta ujian yang telah didaftarkan. • Mencetak hasil ujian peserta
Admin	<p>Aktor yang bertugas untuk :</p> <ul style="list-style-type: none"> • Mendaftarkan peserta ujian ke dalam jadwal ujian, • Mendaftarkan guru untuk menjadi guru pengawas, • Membuat jadwal ujian, • Mengelola data soal, peserta ujian, guru pengawas, jadwal ujian (<i>delete, insert, update, view</i>).
Siswa	<p>Aktor yang bertugas untuk :</p> <ul style="list-style-type: none"> • Melihat list ujian yang sudah dijadwalkan/diaktifkan, • Memilih ujian yang dikerjakan, • Mengerjakan soal ujian, • Melihat hasil ujian yang dikerjakan.

- Kebutuhan Sistem Ujian Online

Pada tahapan ini akan menjelaskan mengenai proses pemetaan dari hasil observasi yang telah dilakukan menjadi rangkuman yang berisikan kebutuhan-kebutuhan sistem. Proses pemetaan kebutuhan dilakukan beberapa tahapan, diantaranya: (1) Elisitasi kebutuhan tahap 1; (2) Eliminasi kebutuhan yang merupakan elisitasi tahap 2. Elisitasi ini menggunakan metode MDI yang berarti (*Mandatory, Desirable, Inessential*).

M (Mandatory) yang berarti Penting, maksudnya adalah kebutuhan tersebut harus ada dan tidak boleh hilang pada sistem yang akan dikembangkan, karena merupakan kebutuhan yang berhubungan dengan aktifitas dalam proses pelaksanaan ujian.

D (Desirable) maksudnya adalah kebutuhan yang tidak terlalu penting dan boleh untuk dihilangkan, karena kebutuhan yang bukan bagian dari proses pelaksanaan ujian, tetapi jika diterapkan dalam sistem dapat membuat sistem menjadi lebih baik.

I (Inessential) maksudnya adalah kebutuhan yang bukan bagian dari sistem, sehingga tidak diterapkan ke dalam sistem yang akan dikembangkan.

(3) Pada tahap 3 semua kebutuhan yang tersisa diklasifikasikan kembali dengan menggunakan metode TOE, yaitu:

T : Teknikal, yang berarti bagaimana tata cara atau teknik yang diusulkan untuk pembuatan sistem.

O : Operasional, yang berarti bagaimana cara penggunaan kebutuhan ke dalam sistem yang akan dikembangkan.

E : Ekonomi, jumlah biaya yang dikeluarkan untuk kebutuhan membangun sistem sesuai dengan semua keperluan untuk sistem yang akan dikembangkan.

Pada metode TOE dibagi kembali menjadi beberapa *option*, yaitu: **High (H)** : Berarti sulit dikerjakan yang disebabkan karena teknik pembuatan dan penggunaannya sulit serta membutuhkan biaya yang mahal, maka kebutuhan tersebut harus dihilangkan. **Middle (M)** : Berarti mampu untuk dikerjakan. **Low (L)** : Berarti mudah untuk dikerjakan.

Setelah dilakukan proses pemetaan kebutuhan, hasil dari keseluruhan dapat disimpulkan ke dalam kebutuhan fungsional dan non-fungsional yang nantinya digunakan sebagai dasar pengembangan suatu sistem. Ditunjukkan pada Tabel 2, Tabel 3, dan Tabel 4.

Tabel 2. Kebutuhan Fungsional

No	Kebutuhan	Aktor
1	Sistem dapat menerima inputan <i>login user</i>	Siswa, Guru, Admin
2	Sistem dapat menambah, meng- <i>update</i> , dan menghapus data soal ujian	Admin
3	Sistem dapat menampilkan soal ujian	Siswa

4	Sistem dapat menerima inputan jawaban pada setiap soal ujian	Siswa
5	Sistem dapat menambah, meng- <i>update</i> , menghapus data peserta ujian	Admin
6	Sistem dapat menampilkan nilai hasil ujian	Guru, Siswa
7	Sistem dapat menambah, meng- <i>update</i> , dan menghapus jadwal ujian	Admin
8	Sistem dapat menampilkan jadwal ujian	Siswa
9	Sistem dapat menampilkan list siswa peserta ujian	Guru
10	Sistem dapat mencetak nilai hasil ujian	Guru
11	Sistem dapat menerima inputan <i>logout</i>	Siswa, Guru, Admin

Tabel 3. Kebutuhan Non Fungsional

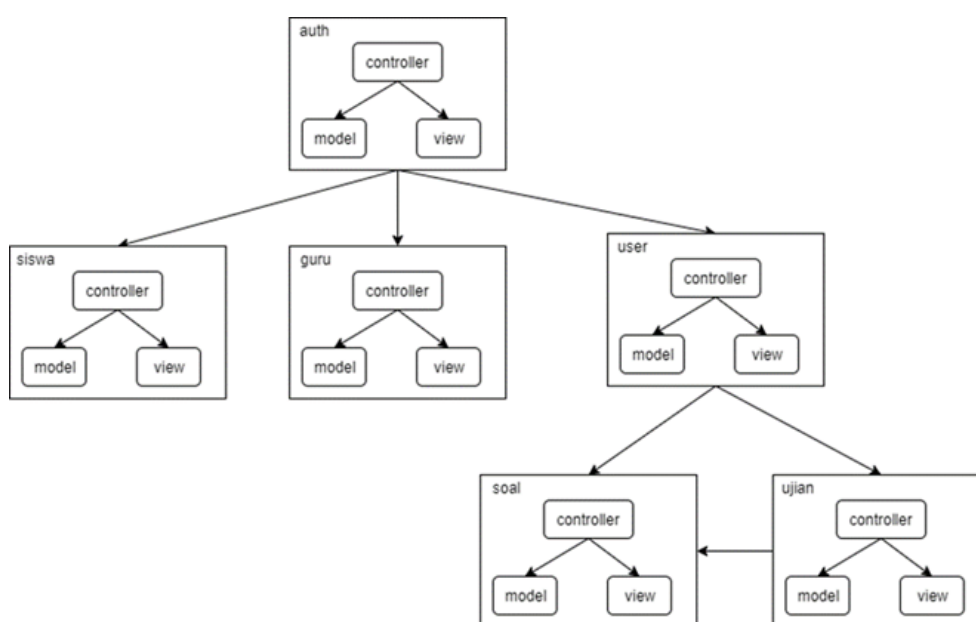
No	Kebutuhan Non Fungsional
1	Sistem memiliki tampilan antarmuka yang mudah dimengerti atau <i>user friendly</i>
2	Sistem dapat memiliki tingkat kecepatan akses yang tinggi
3	Sistem harus ringan, tidak membutuhkan <i>memory space</i> tinggi

Tabel 4. Kebutuhan Non Fungsional dari Aspek Software dan Hardware

No	Kebutuhan Software	Kebutuhan Hardware
1	Xampp	Laptop
2	Sublime Text 3	
3	Browser	
4	Star UML	

2.2.2 Desain

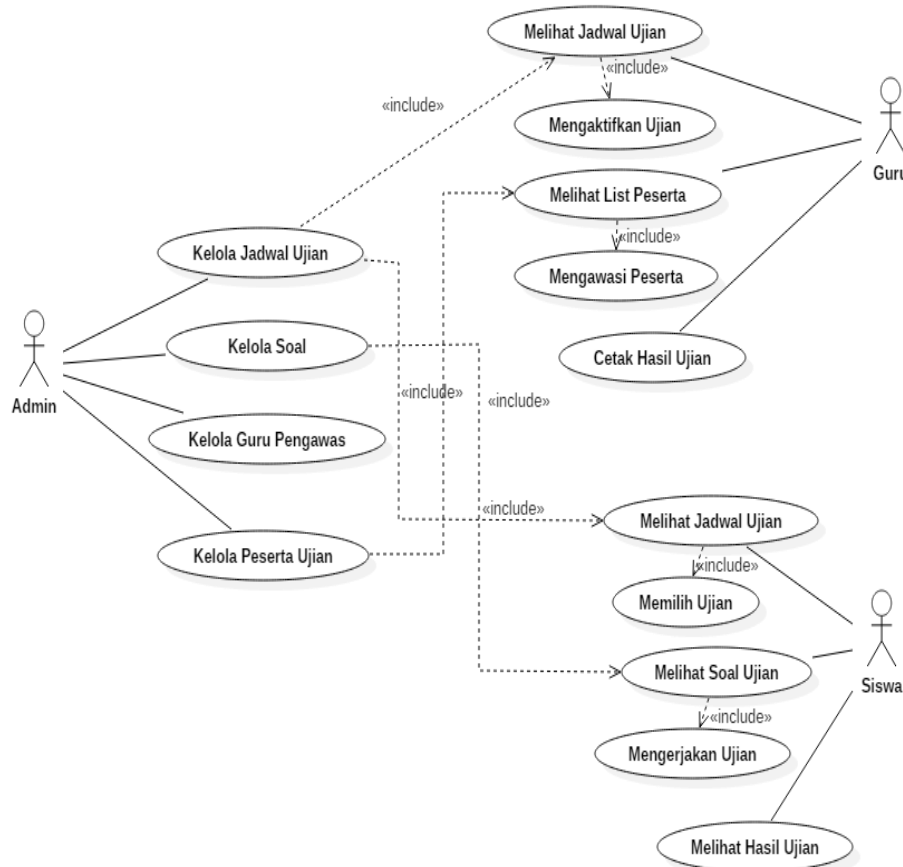
Pada tahap ini dilakukan perancangan struktur dalam sistem yang akan dibuat menggunakan UML (*Unified Modelling Language*) yang meliputi *Use Case Diagram*, *Activity diagram* adalah suatu gambaran aktivitas dari sistem [5], dan *Sequence diagram*, serta menentukan bahasa pemrograman yang digunakan yaitu PHP *Framework CodeIgniter* yang merupakan sebuah *framework* aplikasi web *open source* yang menggunakan bahasa PHP [6], dengan menggunakan konsep HMVC (*Hierarchical Model, Controller, View*) Gambar 2. HMVC merupakan hasil dari pengembangan pola MVC, dari struktur *Model, View, Controller* (MVC) dibuat berlapis menjadi "hierarchy of parent-child MVC layers atau hirarki lapisan orangtua-anak MVC" yang artinya pengembang dapat membuat konsep modular, dan setiap modul menerapkan pola MVC *triad* [1][7].



Gambar 2 Struktur Sistem Implementasi HMVC

- *Usecase Diagram*

Diagram pada Gambar 3 ini merupakan rangkuman dari semua kebutuhan fungsional yang telah didapatkan pada tahapan sebelumnya, didokumentasikan dalam bentuk *usecase diagram* agar lebih mudah untuk memahaminya. Pada *usecase diagram* terdapat dua elemen utama, yaitu sebagai *actor* yang berinteraksi langsung dengan sistem, dan *use case* adalah aktivitas yang dapat dilakukan oleh *actor* yang sudah ditentukan [8].



Gambar 3. Usecase Diagram

2.2.3 Testing

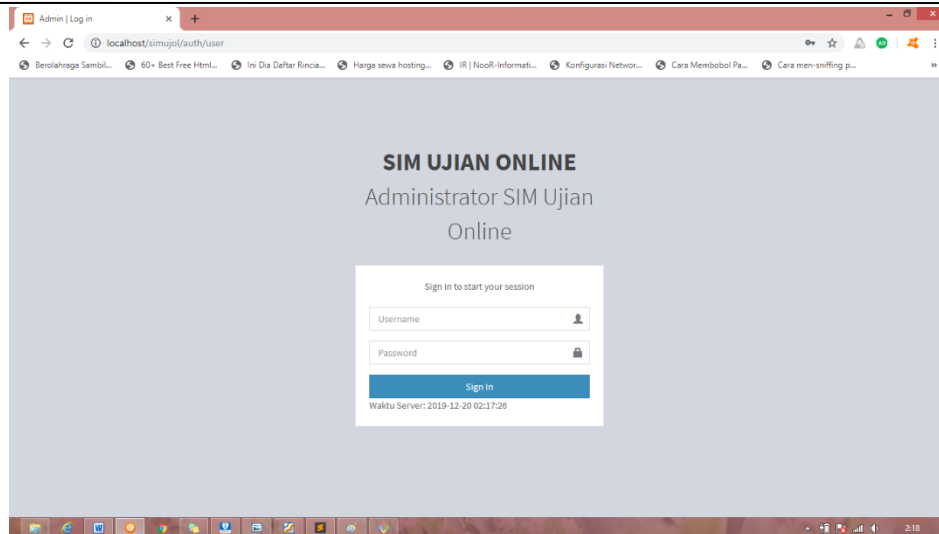
Dilakukan pengujian dengan menggunakan *Integration Testing*, *black box testing* dan UAT (*User Acceptance Test*). *Integration testing* adalah pengujian yang dilakukan pada komponen-komponen yang saling berinteraksi dalam sistem. Pengujian ini dilakukan untuk mengetahui hasil dari integrasi antar komponen apakah berhasil atau tidak. *Black box testing* merupakan sebuah pengujian yang berfokus pada fungsionalitas dari suatu sistem, seperti apakah ada fungsi yang tidak benar atau tidak ada, apakah terdapat kesalahan dari segi *interface*, kesalahan struktur data atau kesalahan performansi. UAT (*User Acceptance Test*) merupakan pengujian yang dilakukan terhadap sistem dengan melibatkan *user* atau pengguna. Pengujian ini biasanya dilakukan dengan cara memberikan beberapa pertanyaan terkait dengan sistem.

3. Hasil Penelitian dan Pembahasan

3.1 Implementasi Sistem

3.1.1 Halaman Login

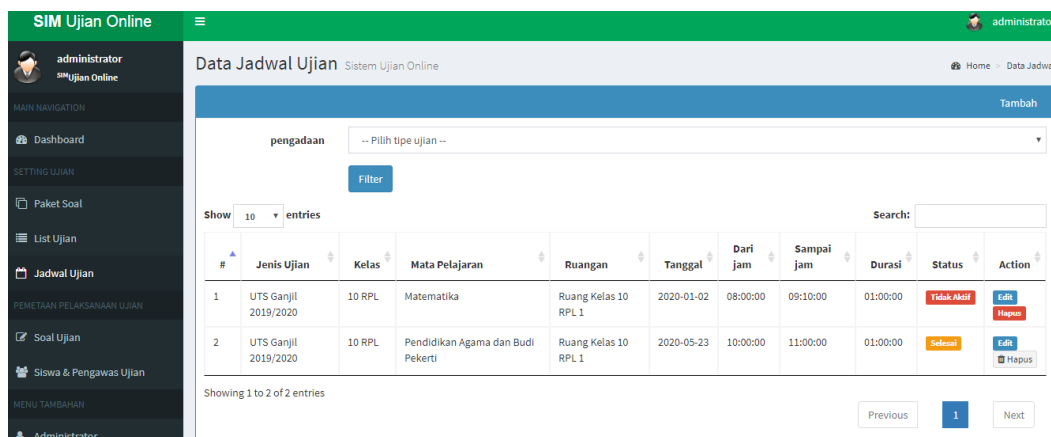
Halaman awal yang ditampilkan sistem pada Gambar 4 adalah halaman *login*. Pengguna harus memasukkan *username* dan *password* yang sudah didaftarkan ke dalam sistem. Jika pengguna berhasil *login*, maka dapat mengakses sistem ke halaman selanjutnya (*dashboard*), jika gagal maka pengguna harus mencoba *login* kembali.



Gambar 4. Halaman Login

3.1.2 Halaman Jadwal Ujian

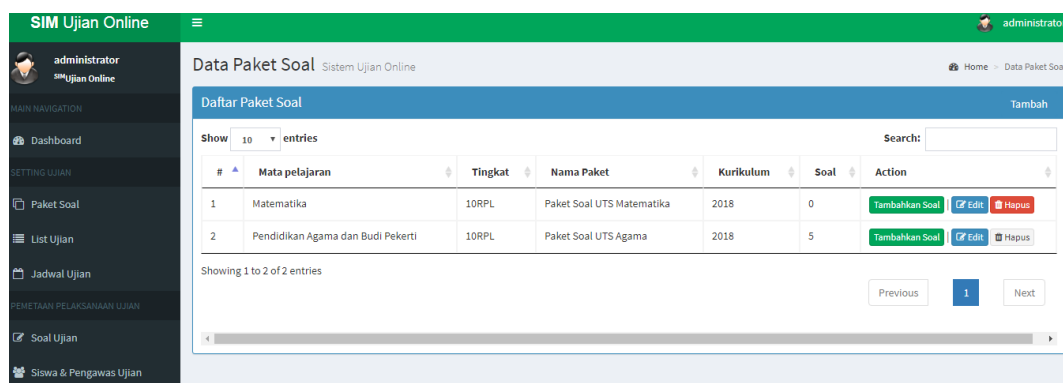
Halaman jadwal ujian pada Gambar 5 adalah halaman yang ditampilkan ketika pengguna memilih menu jadwal ujian. Halaman ini berisi data jadwal ujian yang telah dibuat oleh admin.



Gambar 5. Halaman Jadwal Ujian

3.1.3 Halaman Data Paket Soal Ujian

Setelah selesai menambahkan jadwal ujian baru, pengguna dapat melanjutkan ke menu Paket Soal untuk menambahkan soal pada tiap jadwal yang telah dibuat, seperti pada Gambar 6.



Gambar 6. Halaman Data Paket Soal Ujian

3.1.4 Halaman Menu Peserta & Pengawas Ujian

Pada Gambar 7 menunjukkan ketika pengguna memilih menu peserta dan pengawas ujian, halaman yang ditampilkan adalah list jadwal ujian dan terdapat kolom yang berisikan jumlah peserta dan pengawas ujian pada masing-masing jadwal.

Tingkat	Mata Pelajaran	Ruangan	Tanggal	Waktu	Durasi	Status	Peserta	Pengawas	Action
10 RPL	Pendidikan Agama dan Budi Pekerti	Ruang Kelas 10 RPL 1	2020-05-23	10:00:00 - 11:00:00	01:00:00	Selesai	5 Siswa	1 Guru	+ Peserta + Pengawas
10 RPL	Matematika	Ruang Kelas 10 RPL 1	2020-01-02	08:00:00 - 09:10:00	01:00:00	Tidak Aktif	Kosong	Kosong	+ Peserta + Pengawas

Gambar 7. Halaman Menu Peserta & Pengawas Ujian

3.1.5 Mengaktifkan Ujian

Pengguna sebagai pengawas mengaktifkan ujian. Pengguna membuka menu proses ujian. Pada Gambar 8 menu tersebut menampilkan jadwal ujian untuk pengawas dan pada kolom aksi pengguna dapat memilih aktivasi untuk mengaktifkan ujian dengan status tidak aktif.

Tingkat	Mata Pelajaran	Ruangan	Tanggal	Waktu	Durasi	Status	Peserta	Action
10 RPL	Pendidikan Agama dan Budi Pekerti	Ruang Kelas 10 RPL 1	2020-05-23	10:00:00 - 11:00:00	01:00:00	Selesai	5 Siswa	Aktivasi Reset
10 RPL	Matematika	Ruang Kelas 10 RPL 1	2020-01-02	08:00:00 - 09:10:00	01:00:00	Tidak Aktif	3 Siswa	Aktivasi Reset

Gambar 8. Halaman Untuk Mengaktifkan Ujian

3.1.6 Halaman List Nilai Ujian

Halaman ditampilkan ketika pengguna yang sebagai pengawas memilih menu nilai ujian. Menampilkan jadwal ujian yang telah selesai, kemudian memilih aksi nilai untuk membuka list hasil ujian peserta. Pada Gambar 9, halaman ini dapat dilakukan pencetakan hasil.

No	Nis	Nama	Jenis Kelamin	Kelas	Status	Nilai
1	21124358	Al Morissette	Perempuan	10 RPL - 1	Selesai	100,00
2	65773277	Alberta Reichel	Laki-laki	10 RPL - 1	Selesai	60,00
3	53081630	Archibald Littel	Laki-laki	10 RPL - 1	Selesai	80,00
4	53980820	Betty Murray	Perempuan	10 RPL - 1	Belum Mula	
5	58882302	Bruce Shanahan MD	Laki-laki	10 RPL - 1	Belum Mula	

Gambar 9. List Nilai Ujian Peserta

3.1.7 Memilih Ujian

Halaman memilih ujian pada Gambar 10 adalah halaman yang berisikan jadwal ujian untuk peserta. Peserta diharuskan untuk memilih ujian dengan status aktif agar dapat memulai mengerjakan soal ujian.

Tingkat	Mata Pelajaran	Ruangan	Tanggal	Waktu	Durasi	Status	Action
10 RPL	Pendidikan Agama dan Budi Pekerti	Ruang Kelas 10 RPL 1	2020-05-23	10:00:00 - 11:00:00	01:00:00	Selesai	Selesai, Nilai
10 RPL	Matematika	Ruang Kelas 10 RPL 1	2020-01-02	08:00:00 - 09:10:00	01:00:00	Aktif	Mulai Ujian

Gambar 10. Halaman Jadwal Ujian Peserta

3.1.8 Mengerjakan Soal Ujian

Pada Gambar 11 menampilkan lembar soal ujian peserta, setelah pengguna memilih aksi mulai ujian maka halaman ini dapat ditampilkan.

Gambar 11. Halaman Lembar Soal Peserta

3.1.9 Halaman Hasil Ujian

Pada Gambar 12 halaman ini, ditampilkan ketika peserta selesai dalam mengerjakan soal ujian, atau ujian telah berakhir. Hasil ujian akan ditampilkan secara langsung. Namun peserta jika ingin melihat lagi, dapat memilih aksi nilai pada halaman jadwal ujian.

Gambar 12. Halaman Hasil Ujian Peserta

3.2 Hasil Pengujian

3.2.1 Integration Testing

Integration testing adalah pengujian yang dilakukan pada komponen-komponen yang saling berinteraksi dalam sistem. Pengujian Tabel 5 ini dilakukan untuk mengetahui hasil dari integrasi antar komponen apakah berhasil atau tidak.

Tabel 5. Hasil Integration Testing

ID Pengujian	Deskripsi Pengujian	Hasil Pengujian
A.1 Pengujian modul admin		
A.1.1	testLoginAsAdmin()	Success
A.1.2	testInsertJadwal()	Success
A.1.3	testUpdateJadwal()	Success
A.1.4	testDeleteJadwal()	Success
A.1.5	testInsertSoal()	Success
A.1.6	testUpdateSoal()	Success
A.1.7	testDeleteSoal()	Success
A.1.8	testregister_peserta()	Success
A.1.9	testregister_pengawas()	Success
A.2 Pengujian modul guru		
A.2.1	testLoginAsGuru()	Success
A.2.2	testViewJadwal()	Success
A.2.3	testAktivasi()	Success
A.2.4	testReset()	Success
A.2.5	testViewSiswa()	Success
A.2.6	testViewNilai()	Success
A.3 Pengujian modul siswa		
A.3.1	testLoginAsSiswa()	Success
A.3.2	testcheck_peserta()	Success
A.3.3	testLembarkerja()	Success
A.3.4	testsave_jawaban()	Success
A.3.5	testupdate_timer()	Success
A.3.6	testakhiri_ujian()	Success
A.3.7	testgenerate_soal()	Success

3.2.2 Blackbox

Pengujian Tabel 6, sistem dilakukan dengan menggunakan pengujian *blackbox*, yaitu pengujian yang didasarkan pada spesifikasi sistem untuk mengetahui apakah sudah berfungsi dengan benar [9].

Tabel 6. Hasil Pengujian Blackbox

No	Pengujian	Fungsi	Data	Hasil Pengujian
1	Login	Dapat memvalidasi pengguna	Pengguna	Sukses
2	Memilih menu jadwal ujian	Menampilkan data jadwal ujian	Jadwal ujian	Sukses
3	Memilih <i>button</i> tambah jadwal ujian	Menampilkan <i>form</i> tambah data	Jadwal ujian	Sukses
4	Memilih menu paket soal ujian	Menampilkan data paket soal	Paket soal ujian	Sukses
5	Memilih <i>button</i> tambah soal pada halaman soal ujian	Menampilkan <i>form</i> tambah soal	Soal	Sukses
6	Memilih aksi tambahkan jawaban pada halaman data soal	Menampilkan list jawaban soal	Jawaban ujian	Sukses
7	Memilih aksi tambahkan soal	Menampilkan list soal ujian	Soal ujian	Sukses

		pada halaman paket soal			
8	Memilih menu siswa & pengawas ujian	Menampilkan list jadwal dengan jumlah peserta & pengawas ujian	Jadwal ujian	Sukses	
9	Memilih aksi tambah peserta dan pengawas	Menampilkan data siswa dan guru	Siswa dan guru	Sukses	
10	Memilih menu proses ujian dan memilih aksi aktivasi	Menampilkan jadwal ujian untuk pengawas	Jadwal ujian	Sukses	
11	Memilih aksi peserta pada jadwal ujian untuk pengawas	Menampilkan halaman peserta ujian	Siswa	Sukses	
12	Memilih menu nilai ujian	Menampilkan list hasil ujian peserta	Nilai ujian	Sukses	
13	Memilih menu jadwal ujian untuk siswa	Menampilkan jadwal ujian	Jadwal ujian	Sukses	
14	Memilih aksi mulai ujian	Menampilkan lembar soal	Soal	Sukses	
15	Memilih aksi nilai	Menampilkan nilai ujian peserta	Nilai ujian	Sukses	

3.2.3 UAT (*User Accepted Test*)

Pengujian UAT (*User Accepted Test*) pada Tabel 7 merupakan pengujian sistem yang dilakukan untuk melihat seberapa tingkat kemudahan dan kepuasan dari pengguna terhadap sistem yang dikembangkan [10]. Perwakilan dari aktor yang terlibat yaitu *Admin, Guru, Siswa*.

Tabel 7. Hasil Pengujian UAT (*User Accepted Test*)

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Hasil Pengujian		
			Diterima	Diterima Dengan Catatan	Ditolak
A.1	Kelas pengujian modul akses				
A.1.1	Pengujian Halaman <i>Login</i>	<ul style="list-style-type: none"> Memasukkan <i>username</i> dan <i>password</i> Klik <i>button login</i> 	Admin, Guru, Siswa		
A.2	Kelas pengujian modul <i>dashboard</i>				
A.2.1	Pengujian menampilkan halaman <i>dashboard</i>	<ul style="list-style-type: none"> Setelah melakukan <i>login</i> 	Admin, Guru, Siswa		
A.3	Kelas pengujian modul jadwal ujian				
A.3.1	Pengujian menampilkan halaman jadwal ujian				

	Pengujian menampilkan jadwal ujian	<ul style="list-style-type: none"> • Pilih jadwal ujian 	Admin, Guru, Siswa
A.3.2	Pengujian tambah jadwal ujian		
	Pengujian tambah jadwal ujian	<ul style="list-style-type: none"> • Pilih <i>button</i> tambah • Memasukkan data jadwal • Pilih simpan 	Admin
A.3.3	Pengujian edit jadwal ujian		
	Pengujian mengedit jadwal ujian	<ul style="list-style-type: none"> • Pilih aksi edit • Memasukkan data <i>update</i> • Pilih simpan 	Admin
A.3.4	Pengujian hapus jadwal ujian		
	Pengujian menghapus jadwal ujian	<ul style="list-style-type: none"> • Pilih aksi <i>delete</i> • Menampilkan konfirmasi 	Admin
A.4	Kelas pengujian modul soal ujian		
A.4.1	Pengujian menampilkan soal ujian		
	Pengujian menampilkan list soal ujian	<ul style="list-style-type: none"> • Pilih menu paket soal • Pilih aksi tambahkan soal 	Admin
A.4.2	Pengujian tambah soal ujian		
	Pengujian tambah soal	<ul style="list-style-type: none"> • Pilih menu paket soal • Pilih aksi tambahkan soal • Klik <i>button</i> tambah 	Admin
A.4.3	Pengujian edit soal ujian		
	Pengujian mengedit soal ujian	<ul style="list-style-type: none"> • Pilih menu paket soal • Pilih aksi tambahkan soal • Pilih aksi edit 	Admin
A.4.4	Pengujian hapus soal ujian		
	Pengujian menghapus soal ujian	<ul style="list-style-type: none"> • Pilih menu paket soal • Pilih aksi tambahkan soal • Pilih aksi hapus 	Admin
A.5	Kelas pengujian modul peserta & pengawas		
A.5.1	Pengujian menampilkan jumlah peserta & pengawas		
	Pengujian menampilkan peserta & pengawas	<ul style="list-style-type: none"> • Pilih menu siswa & pengawas • Pilih aksi <i>view</i> jadwal ujian 	Admin
A.5.2	Pengujian tambah jumlah peserta & pengawas		

	Pengujian tambah peserta & pengawas	<ul style="list-style-type: none"> • Pilih menu siswa & pengawas • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi tambah peserta/pengawas • Pilih aksi <i>add</i> 	Admin
A.5.3	Pengujian hapus jumlah peserta & pengawas		
	Pengujian hapus peserta & pengawas	<ul style="list-style-type: none"> • Pilih menu siswa & pengawas • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi tambah peserta/pengawas • Pilih aksi <i>remove</i> 	Admin
A.6	Kelas pengujian modul aktivasi ujian, mengawasi peserta ujian		
A.6.1	Pengujian mengaktifkan ujian		
	Pengujian aktivasi ujian	<ul style="list-style-type: none"> • Pilih proses ujian • Pilih aksi jadwal ujian • Pilih aksi aktivasi 	Guru
A.6.2	Pengujian melihat & mengawasi peserta		
	Pengujian mengawasi peserta ujian	<ul style="list-style-type: none"> • Pilih proses ujian • Pilih aksi jadwal ujian • Pilih aksi peserta 	Guru
A.7	Kelas pengujian modul hasil ujian dan mencetak		
A.7.1	Pengujian menampilkan hasil ujian		
	Pengujian menampilkan data hasil	<ul style="list-style-type: none"> • Pilih menu nilai ujian • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi nilai siswa 	Guru
A.7.2	Pengujian mencetak hasil ujian peserta		
	Pengujian hasil ujian	<ul style="list-style-type: none"> • Pilih menu nilai ujian • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi nilai siswa • Klik <i>button excell/pdf/print</i> 	Guru
A.8	Kelas modul memilih & mengerjakan soal ujian		
A.8.1	Pengujian memilih mata pelajaran yang diujikan		

	Pengujian memilih ujian	<ul style="list-style-type: none"> • Pilih jadwal ujian • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi mulai ujian 	Siswa
A.8.2	Pengujian mengerjakan soal ujian		
	Pengujian mengerjakan soal	<ul style="list-style-type: none"> • Pilih jadwal ujian • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi mulai ujian • Menampilkan lembar soal • Mulai mengerjakan 	Siswa
A.9	Kelas pengujian modul nilai ujian		
A.9.1	Pengujian menampilkan nilai ujian		
	Pengujian menampilkan nilai	<ul style="list-style-type: none"> • Pilih jadwal ujian • Pilih aksi <i>view</i> jadwal ujian • Pilih aksi nilai 	Siswa

4. Kesimpulan

1. Perancangan sistem informasi ujian online dengan menggunakan framework codeigniter yang mengimplementasi HMVC (Hierarchical Model, View, Controller) dapat menghasilkan sistem yang lebih modular.
2. Hasil dari implementasi pola HMVC memberi kemudahan dalam pembangunan sistem, karena sifatnya yang lebih modular dengan memetakan folder untuk masing-masing layer sehingga dapat lebih terfokus pada masing-masing modul yang sedang dikerjakan.
3. Dengan adanya sistem informasi ujian online yang telah dikembangkan dapat membantu SMK Muhammadiyah 8 Siliragung Banyuwangi dalam melakukan segala proses pelaksanaan ujian, mulai dari informasi jadwal, lembar kerja soal, sampai dengan penilaian.
4. Sistem ini telah dibangun dengan melalui proses pengujian dari segi integrasi sistem menggunakan integration testing pada phpunit, sedangkan pengujian kebutuhan fungsional menggunakan blackbox testing dan UAT (User Accepted Test). Hasil dari kedua segi pengujian menunjukkan keberhasilan, sistem sesuai dengan kebutuhan dari user.

5. Saran

1. Sistem informasi ujian *online* dapat mengelola soal *esai* yang bersifat *random*.
2. Pembangunan sistem kedepannya dapat melakukan pengujian pada HMVC dengan metode yang berbeda.

Referensi

- [1] U. D. Bramantya, "Perancangan Dan Implementasi Sistem Informasi Rumah Sakit Menggunakan Arsitektur Hierarchical Model, View, Controller Berbasis Web (Studi Kasus: Rumah Sakit Puri Asih Salatiga)," *Peranc. Dan Implementasi Sist. Inf. Rumah Sakit Menggunakan Arsit. Hierarchical Model. View, Controll. Berbas. Web (Studi Kasus Rumah Sakit Puri Asih Salatiga)*, no. April, p. 32, 2017.
- [2] Y. Sandra, Randi Proska; Hadi, Ahmaddul; Huda, "Perancangan Aplikasi Clickost Berbasis Web Menggunakan Metode Hmvc (Hierarchical Model View Controller) Sebagai Sistem Pencarian Dan Promosi Indekos Di Kota Padang," *Peranc. Apl. Clickost Berbasis Web Menggunakan Metod. Hmvc (Hierarchical Model View Control. Sebagai Sist. Pencarian Dan Promosi Indekos Di Kota Padang*, vol. 4, no. 1, pp. 1–12, 2017.
- [3] W. Suryan, *Software Quality Engineering: A Practitioner's Approach*, vol. 9781118592. 2014.

-
- [4] J. Cadle and D. Yeates, *Project Management for Information Systems (5th Edition)*. 2008.
 - [5] A. Hendini, "Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)," *Crop Sci.*, vol. 4, no. 2, p. 201, 2016.
 - [6] A. Griffiths, *CodeIgniter 1.7 Professional Development*. 2010.
 - [7] C. Supaartagorn, "PHP Framework for Database Management Based on MVC Pattern," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 2, pp. 251–258, 2011.
 - [8] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 1, p. 77, 2018.
 - [9] G. W. Setiawan, "Pengujian Perangkat Lunak Menggunakan Metode Black Box Studi Kasus Exelsa Universitas Sanata Dharma," p. 286, 2011.
 - [10] W. S. Wardhono, L. P. Kusuma, W. S. Wardhono, and K. Malang, "Evaluasi User Acceptance Augmented Reality," pp. 978–979, 2015.