

BAB III

IMPLEMENTASI PROGRAM

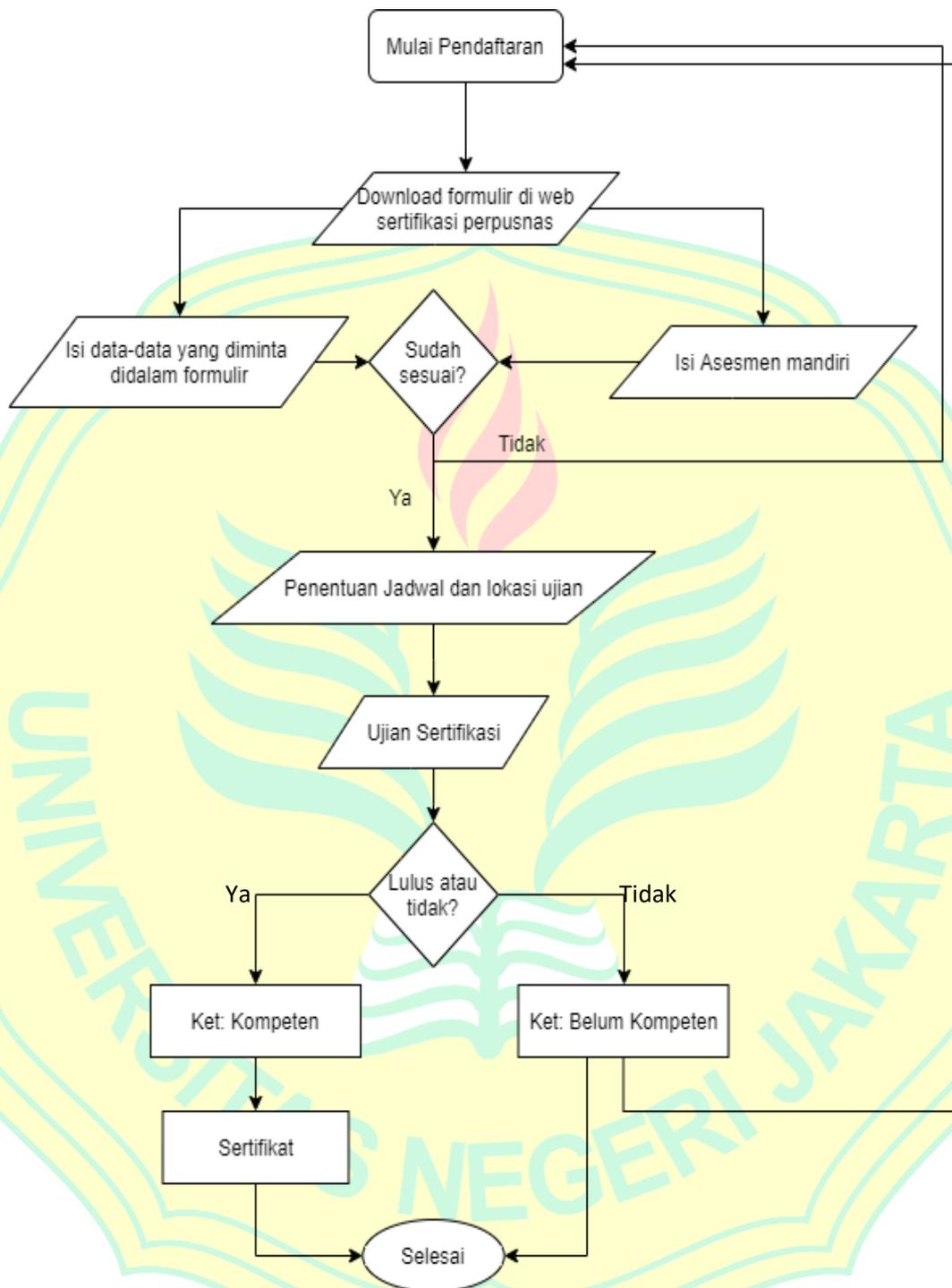
Penelitian ini, penulis mengembangkan perangkat lunak dengan menggunakan Model Spiral. Terdapat iterasi (*cycle*) atau prototipe aplikasi pada model ini. Rentang waktu untuk satu iterasi pada pengembangan sistem ini selama satu minggu. Pada tiap iterasi akan dilakukan pengembangan sebanyak 2 (dua) *use case*, sehingga jumlah keseluruhan iterasi yaitu 10 (sepuluh). Berikut beberapa tahapan dalam pengembangan sistem ini:

3.1 Analisis Kebutuhan

Pada tahap ini penulis melakukan wawancara dengan Bapak Husna selaku Pegawai Pranata Komputer dan Bapak Ranga selaku Pegawai Sertifikasi Pustakawan pada tanggal 27 November 2019, 26 Desember 2019 dan 24 Februari 2020 di Perpustakaan Nasional Republik Indonesia, Salemba Raya.

Berdasarkan hasil wawancara, penulis diminta untuk membuat sistem sertifikasi yang sesuai dengan alur pendaftaran sebelumnya, dimulai dari pendaftaran sertifikasi atau pengajuan sertifikasi hingga penerbitan hasil ujian berupa sertifikat. Sedangkan untuk ujian kompetensi atau asesmen sertifikasinya akan dilakukan secara langsung tanpa melalui sistem, dengan jadwal dan lokasi ujian yang akan diberikan oleh Lembaga Sertifikasi Pustakawan (LSP) perpusnas melalui sistem kepada peserta.

Sebelum itu penulis telah melakukan observasi terhadap sistem sertifikasi sebelumnya dan telah melakukan analisis kebutuhan yang didapatkan dari hasil wawancara dengan kedua pegawai perpustakaan nasional republik Indonesia bagian sertifikasi [Lampiran 1]. Hasil dari observasi dan analisis yang telah dilakukan oleh penulis dicantumkan kedalam *flowchart* yang telah dibuat sebagai berikut:



Gambar 3.1: Gambar *Flowchart* Proses Sertifikasi Secara Manual

Pada sistem yang akan dibuat terdapat tiga aktor yaitu peserta, admin dan asesor. Berikut adalah tabel analisis kebutuhan dari setiap aktor pada sistem yang akan dibuat:

Tabel 3.1: Tabel analisis user Sistem Informasi Sertifikasi Pustakawan

<i>User</i>	Tugas
Peserta	Dapat melakukan pendaftaran akun
	Dapat melakukan pendaftaran sertifikasi dengan mengisi formulir dan melampirkan dokumen persyaratan
	Dapat melihat status tahap sertifikasi yang telah diajukan
	Dapat melakukan ujian asesmen mandiri
	Dapat melihat jadwal sertifikasi yang akan diikuti
	Dapat melihat hasil ujian sertifikasi yang telah dilakukan
Admin	Dapat mengunduh sertifikat jika dinyatakan kompeten
	Dapat melakukan perubahan status pengajuan sertifikasi yang diajukan oleh peserta (diterima, dikembalikan, atau ditolak)
	Dapat mengelola pesan/komentar yang masuk dari <i>user</i> lain
	Dapat mengelola <i>user</i> yang ada pada sistem
	Dapat membuat soal asesmen mandiri
	Dapat memasukkan sertifikat peserta yang dinyatakan kompeten
Asesor	Dapat membuat berita artikel
	Dapat melihat dan menilai asesmen mandiri peserta serta menentukan jadwal ujian sertifikasi peserta
	Dapat memasukkan nilai sertifikasi peserta berupa kompeten atau tidak kompeten

Selanjutnya penulis melakukan analisis menu yang akan dimasukkan ke dalam sistem dan fungsi dari masing-masing menu sesuai dengan tugas dari *user* sebelumnya. Penulis kemudian mengklasifikasikan menu yang ada pada sistem informasi sertifikasi menjadi 8 menu yaitu *dashboard*, pengajuan sertifikasi, asesmen mandiri, asesmen sertifikasi, manajemen *user*, manajemen soal, berita, pesan dan *profile*. Tiap *user* memiliki menu yang berbeda. Pada admin tidak terdapat menu asesmen mandiri. Pada peserta hanya terdapat menu *dashboard*, *profile*, pengajuan sertifikasi, asesmen mandiri dan asesmen sertifikasi. Asesor memiliki menu yang sama dengan peserta, hanya saja untuk menu pengajuan sertifikasi tidak ada. Berikut adalah tabel yang menyajikan menu dan fungsinya pada *user*:

Tabel 3.2: Tabel analisis menu Sistem Informasi Sertifikasi Pustakawan

Menu	Peserta	Asesor	Admin
Dashboard	Melihat status pengajuan sertifikasi, jadwal dan hasil dari proses sertifikasi yang telah diajukan peserta	Menampilkan “Selamat datang, asesor”	Melihat semua status pengajuan sertifikasi peserta, dan proses yang telah dilakukan peserta
Pengajuan Sertifikasi	Melakukan pengajuan sertifikasi	-	Melihat semua daftar pengajuan sertifikasi peserta
Asesmen Mandiri	Mengerjakan asesmen mandiri	Melihat daftar semua peserta yang telah melakukan asesmen mandiri, melakukan	-

Menu	Peserta	Asesor	Admin
		penilaian, memilih metode untuk asesmen sertifikasi, dan menentukan jadwal ujian sertifikasi	
Asesmen Sertifikasi	Melihat hasil asesmen sertifikasi	Melihat daftar peserta yang mengikuti asesmen sertifikasi, memasukkan nilai ujian sertifikasi peserta	Melihat daftar peserta, dan memasukkan sertifikat untuk peserta yang dinyatakan kompeten
Manajemen Soal	-	-	Membuat soal-soal untuk asesmen mandiri
Manajemen User	-	-	Mengelola data <i>user</i>
Profile	Memperbarui atau mengubah data diri dan foto <i>profile</i>		
Berita	Melihat berita		Melihat dan Mengelola berita artikel
Pesan	Mengirim pesan/komentar		Mengelola pesan

3.2 Planning

Dalam fase ini penulis menggambarkan hasil analisis yang telah dilakukan ke dalam bentuk visual berupa pemodelan sistem yang berorientasi objek dengan *Unified Modelling Language* (UML), seperti *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Entity Relationship Diagram* (ERD) dan desain *user interface*.

3.2.1 Use Case Diagram

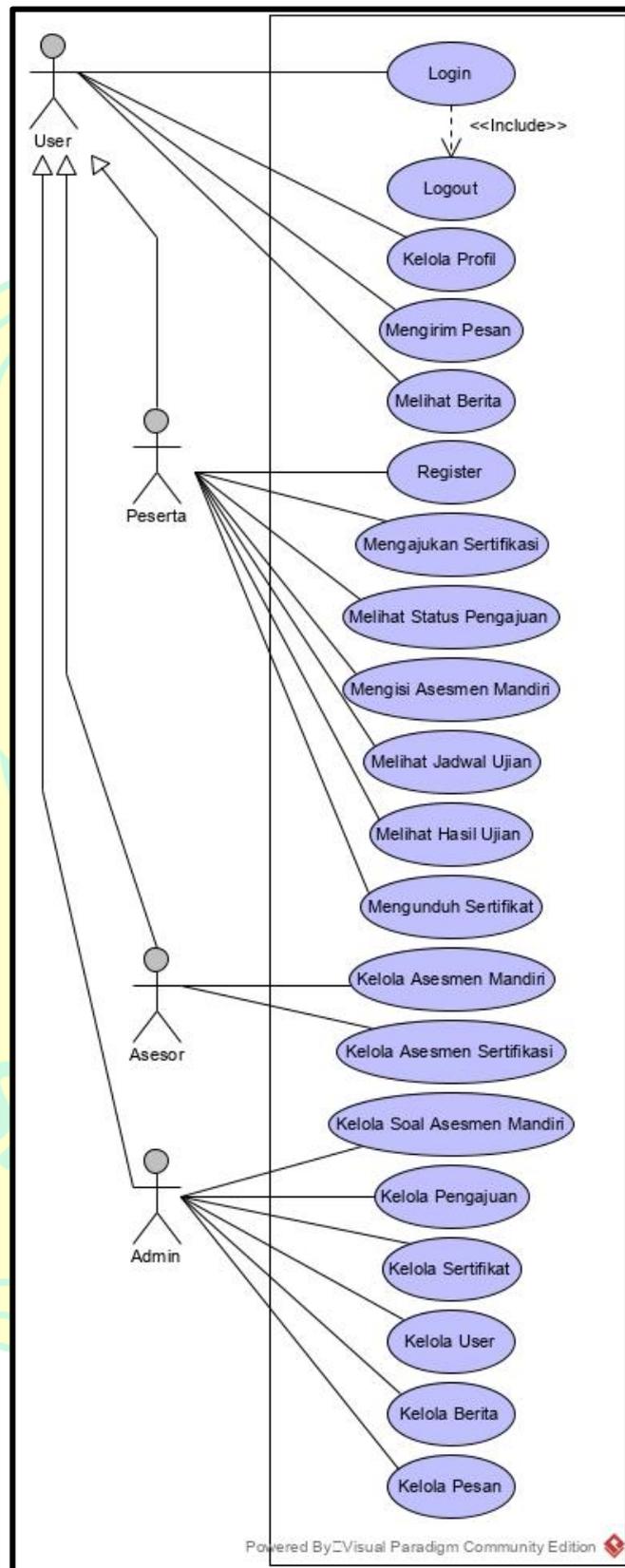
Di dalam *use case diagram* penulis menjelaskan interaksi antara pengguna dari sistem atau aktor, dengan sistem. Setiap aktor memiliki perannya masing-masing di dalam sistem yang sedang dikembangkan oleh penulis. Pada sistem informasi sertifikasi ini terdapat 3 aktor yang digeneralisasi dari *user*. Setiap *user* harus melakukan *login* sebelum masuk ke dalam sistem. Setiap *user* juga dapat mengelola profil, membaca berita, dan mengirim pesan. Aktor-aktor yang terdapat dalam sistem yaitu peserta, asesor dan admin.

Aktor peserta di dalam sistem merupakan seorang pustakawan yang akan melakukan pengajuan sertifikasi. Aktor asesor di dalam sistem merupakan pustakawan yang diangkat oleh LSP Perpustakaan sebagai pengawas dan penilai ujian dengan syarat tertentu dan untuk jangka waktu tertentu. Aktor admin di dalam sistem adalah pegawai Lembaga Sertifikasi Pustakawan Perpustakaan.

Peserta memiliki beberapa tugas, yaitu melakukan pengajuan sertifikasi dengan mengisi data, memasukkan bukti kompetensi, memasukkan dokumen pendukung atau dokumen yang telah disyaratkan oleh pihak perpustakaan, mengisi dan mengedit profil akun, melihat status pengajuan sertifikasi yang telah diajukan apakah sudah diverifikasi oleh pegawai LSP atau belum, mengerjakan asesmen mandiri, melihat jadwal ujian, lokasi ujian dan hasil ujian sertifikasi, jika dinyatakan kompeten maka peserta dapat mengunduh sertifikat. Jika dinyatakan tidak kompeten maka peserta dapat mengajukan permohonan sertifikasi ulang kepada pihak sertifikasi perpustakaan.

Asesor merupakan *user* yang telah diberi izin akses oleh admin agar bisa memasukkan nilai asesmen mandiri peserta, menentukan jadwal ujian sertifikasi peserta serta memberikan nilai asesmen sertifikasi peserta beserta keterangan.

Admin merupakan pegawai LSP perpusnas yang memiliki tugas dalam mengelola sistem secara keseluruhan. Admin bertugas mengelola daftar pengajuan sertifikasi yang telah diajukan oleh peserta sertifikasi. Ketika peserta telah mengajukan sertifikasi maka admin akan memeriksa data diri dalam formulir pengajuan dan dokumen yang telah diunggah oleh peserta apakah telah sesuai dan lengkap, atau belum, setelah itu admin akan melakukan perubahan status pengajuan sertifikasi peserta seperti diterima, dikembalikan atau pengajuan ditolak, setelah itu admin juga memiliki tugas memasukkan sertifikat untuk peserta yang dinyatakan kompeten oleh asesor, admin dapat mengelola soal untuk asesmen mandiri peserta, mengelola berita pada *website* sertifikasi perpusnas, mengelola data *user* yang ada di dalam sistem dan dapat mengelola pesan yang masuk. *Use case diagram* dari sistem terdapat pada gambar 3.1.



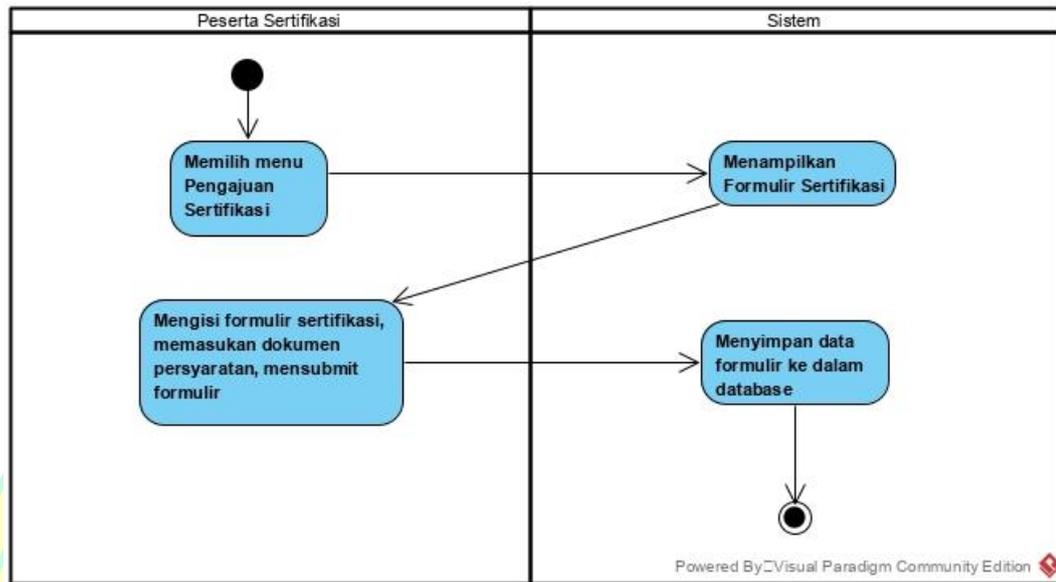
Gambar 3.2: Gambar *Use Case Diagram* Sertifikasi

3.2.2 Activity Diagram

Diagram ini merupakan diagram yang menunjukkan alur kerja dari sistem. Di dalam sistem ini peserta yang ingin melakukan pengajuan sertifikasi harus melakukan *login* terlebih dahulu, apabila peserta belum memiliki akun, maka peserta diharuskan registrasi atau *sign up* untuk membuat akun dengan mencantumkan *e-mail* peserta yang aktif. Sistem akan memberikan *password* ke alamat *e-mail* peserta yang telah didaftarkan. Kemudian, peserta dapat *login* dengan *e-mail* dan *password* yang telah diberikan. Apabila peserta lupa dengan *password* akun, maka peserta dapat menghubungi admin atau mengubahnya sendiri melalui halaman utama *login*.

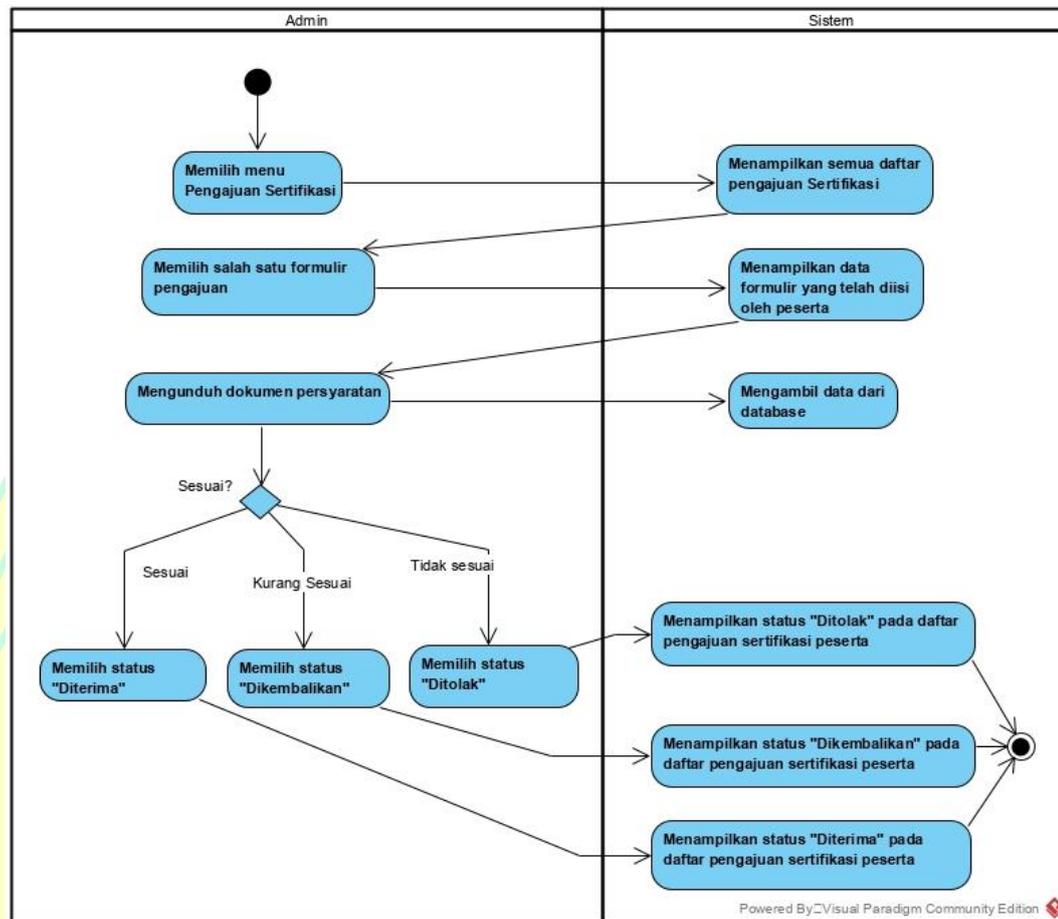
Saat peserta masuk ke dalam sistem, peserta harus melengkapi terlebih dahulu data diri pada sistem, seperti mengisi nomer telepon untuk dapat melakukan pengajuan sertifikasi. Pengajuan dilakukan dengan mengisi formulir data diri dan mengunggah dokumen yang diminta oleh sistem. Peserta juga dapat melihat status pengajuan pada halaman *dashboard* sistem jika pengajuan telah dikirimkan.

Di dalam formulir pengajuan sertifikasi terdapat data diri, data pekerjaan sekarang, data permohonan sertifikasi yang diinginkan, dan dokumen wajib yang harus diunggah. Peserta akan diminta oleh sistem untuk mengunggah beberapa dokumen seperti KTP, Ijazah, SK atau Surat Keterangan Bekerja, Surat Rekomendasi, dan Surat Daftar Riwayat Hidup (DRH). Setelah pengajuan diterima, peserta akan diminta untuk mengerjakan asesmen mandiri dan melampirkan bukti pendukung kompetensi sesuai dengan unit kompetensi yang dipilih, contohnya seperti bukti bahwa peserta dapat mengoperasikan komputer tingkat dasar, menyusun rencana kerja perpustakaan, dan lainnya. Setelah itu peserta hanya menunggu hasil penilaian asesmen dari asesor dan menunggu jadwal ujian sertifikasi yang akan diikuti.



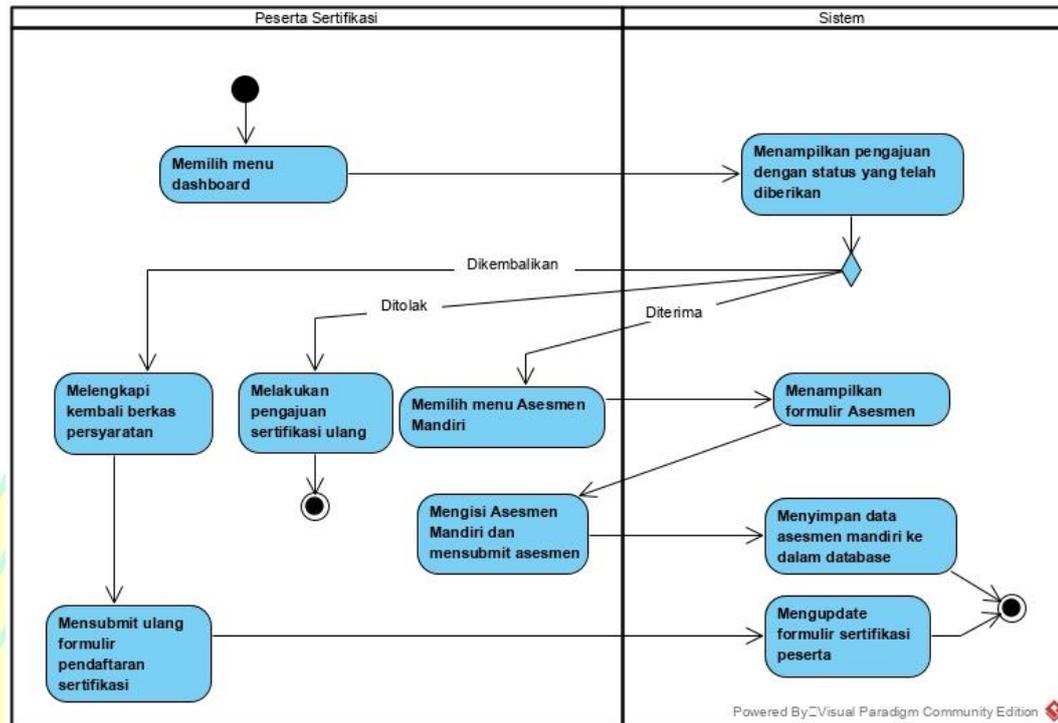
Gambar 3.3: Activity Diagram Pengajuan Sertifikasi

Setelah peserta melakukan pengajuan, admin akan memeriksa kelengkapan data dan dokumen yang telah dimasukkan peserta ke dalam sistem, kemudian mengubah status pengajuan menjadi diterima jika data diri dan dokumen yang dimasukkan peserta dianggap sudah lengkap, mengubah status menjadi dikembalikan jika ada dokumen yang dianggap tidak lengkap atau salah, dan mengubah status menjadi ditolak jika minimal pendidikan dan pengalaman perpustakaananya tidak sesuai atau peserta menggunakan bukti pekerjaan milik orang lain.



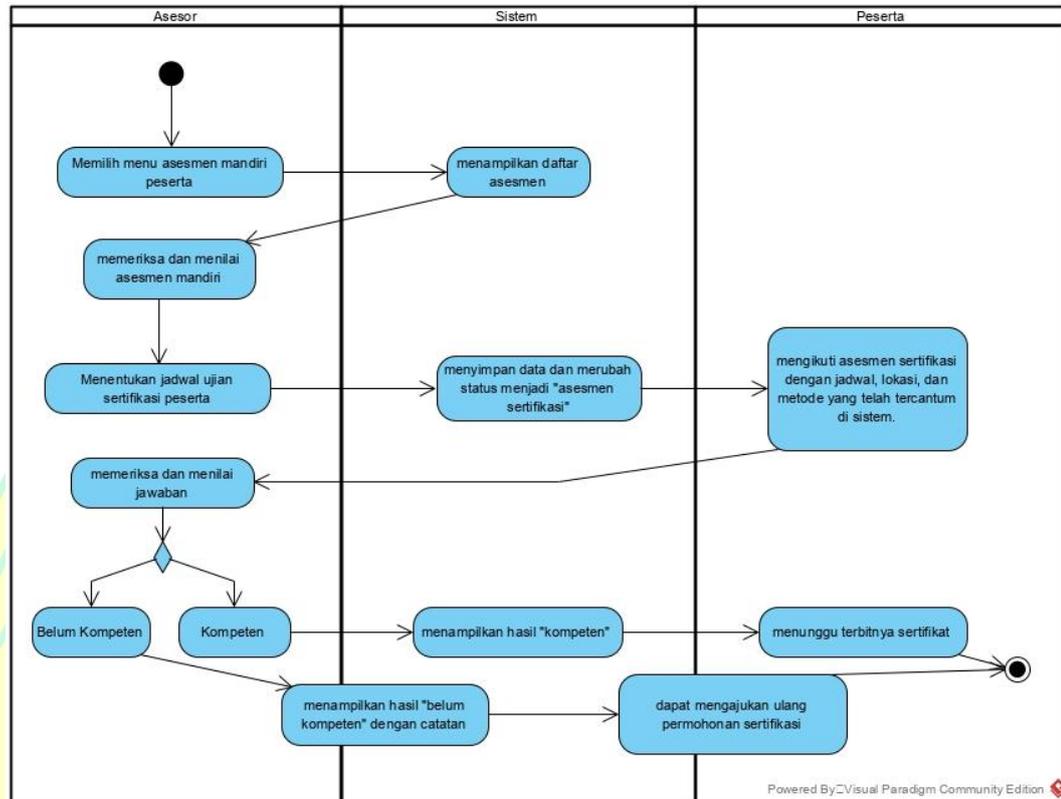
Gambar 3.4: Activity Diagram Verifikasi Data dan Dokumen

Apabila berkas ditolak oleh admin, maka peserta harus melakukan pengajuan sertifikasi ulang, namun jika berkas dikembalikan oleh admin, maka peserta hanya perlu memperbaiki sesuai dengan keterangan yang diberikan oleh admin. Jika berkas diterima, maka status pada *dashboard* peserta berubah menjadi “Diterima” yang mengharuskan peserta mengisi asesmen mandiri untuk tahap selanjutnya. Asesmen mandiri ini hanya bisa diakses peserta apabila peserta telah melakukan pengajuan sertifikasi dan pengajuan diterima oleh admin. Apabila peserta belum melakukan pengajuan maka peserta tidak akan bisa mengerjakan asesmen mandiri.



Gambar 3.5: Activity Diagram Asesmen Mandiri

Setelah peserta mengerjakan asesmen mandiri, asesor akan melihat dan menilai asesmen mandiri peserta dengan kriteria *Valid, Authentic, Current, Sufficient* (VACS), setelah itu asesor akan menentukan jadwal dan lokasi untuk dilakukannya asesmen sertifikasi oleh peserta. Asesmen sertifikasi yang akan diikuti oleh peserta terdiri dari 3 tahapan yaitu, tertulis, wawancara dan demo. Kemudian peserta akan melakukan ujian sesuai dengan jadwal dan lokasi yang telah ditentukan dan dengan 3 tahapan sertifikasi tersebut. Ketika ujian selesai, asesor akan menilai hasil ujian yang telah dilakukan oleh peserta dan memasukkan hasilnya melalui sistem apakah peserta dinyatakan kompeten atau tidak kompeten beserta keterangan yang akan diberikan oleh asesor.



Gambar 3.6: Activity Diagram Penilaian Asesmen

Untuk peserta yang dinyatakan kompeten maka akan diterbitkannya sertifikat oleh BNSP (Badan Nasional Sertifikasi Profesi) yang nantinya akan dimasukkan oleh admin kedalam sistem. Sedangkan untuk peserta yang dinyatakan belum kompeten dapat melakukan pengajuan sertifikasi ulang.

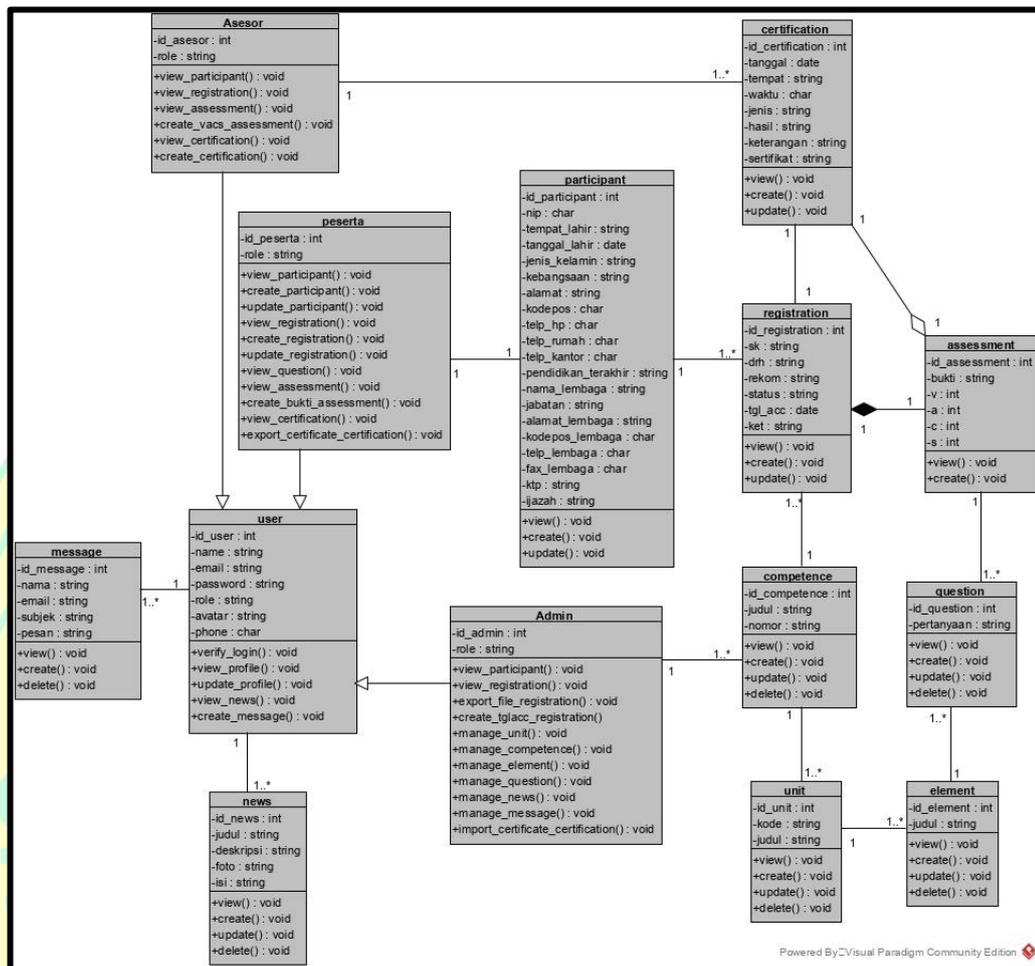
3.2.3 Class Diagram

Class Diagram untuk Sistem Informasi Sertifikasi memiliki 14 *class*. *Class user* berfungsi untuk *login* ke dalam sistem yaitu peserta, asesor dan admin. *Class news* untuk memberikan informasi berita mengenai sertifikasi di perpunas yang akan dikelola oleh admin sertifikasi perpunas dan berita ini dapat dilihat di beranda awal *website*. *Class message* untuk memberikan informasi pesan atau komentar yang diberikan oleh pengunjung, peserta dan asesor, yang akan dikelola oleh admin.

Class Participant berfungsi untuk memberikan informasi mengenai pengajuan yang telah ditambahkan oleh peserta untuk kemudian dikelola oleh admin. *Class Registration* berfungsi untuk memberikan informasi mengenai dokumen wajib yang akan diunggah oleh peserta dan status pengajuan dari admin. *Class Certification* berfungsi untuk memberikan informasi mengenai hasil sertifikasi peserta yang akan dimasukkan oleh asesor dan informasi mengenai sertifikat peserta yang dimasukkan oleh admin. Terdapat *composition* antara *registration* dan *assesment*, ini dikarenakan *class assesment* tidak bisa berdiri tanpa adanya *class registration*.

Class assesment berfungsi untuk memberikan informasi mengenai penilaian asesmen mandiri peserta yang telah dilakukan oleh asesor. *Class competence* berfungsi untuk memberikan informasi mengenai kompetensi yang tersedia untuk dipilih oleh peserta pada saat pengajuan sertifikasi. *Class unit* berfungsi untuk memberikan informasi mengenai unit yang ada di dalam kompetensi yang telah dipilih oleh peserta. *Class elemen* memberikan informasi kepada peserta mengenai elemen yang ada disetiap unit kompetensi pada saat asesmen mandiri. *Class question* memberikan informasi kepada peserta mengenai soal-soal yang harus diisi oleh peserta pada saat peserta melakukan asesmen mandiri.

Class assesment memiliki hubungan *aggregation* dengan *class certification*, karena *class assesment* dapat berdiri sendiri tanpa adanya *class certification*, namun sebaliknya *class certification* tidak dapat berdiri sendiri jika tidak adanya *class assesment*.



Gambar 3.7: Class Diagram Sistem Informasi Sertifikasi

3.2.4 Entity Relationship Diagram (ERD)

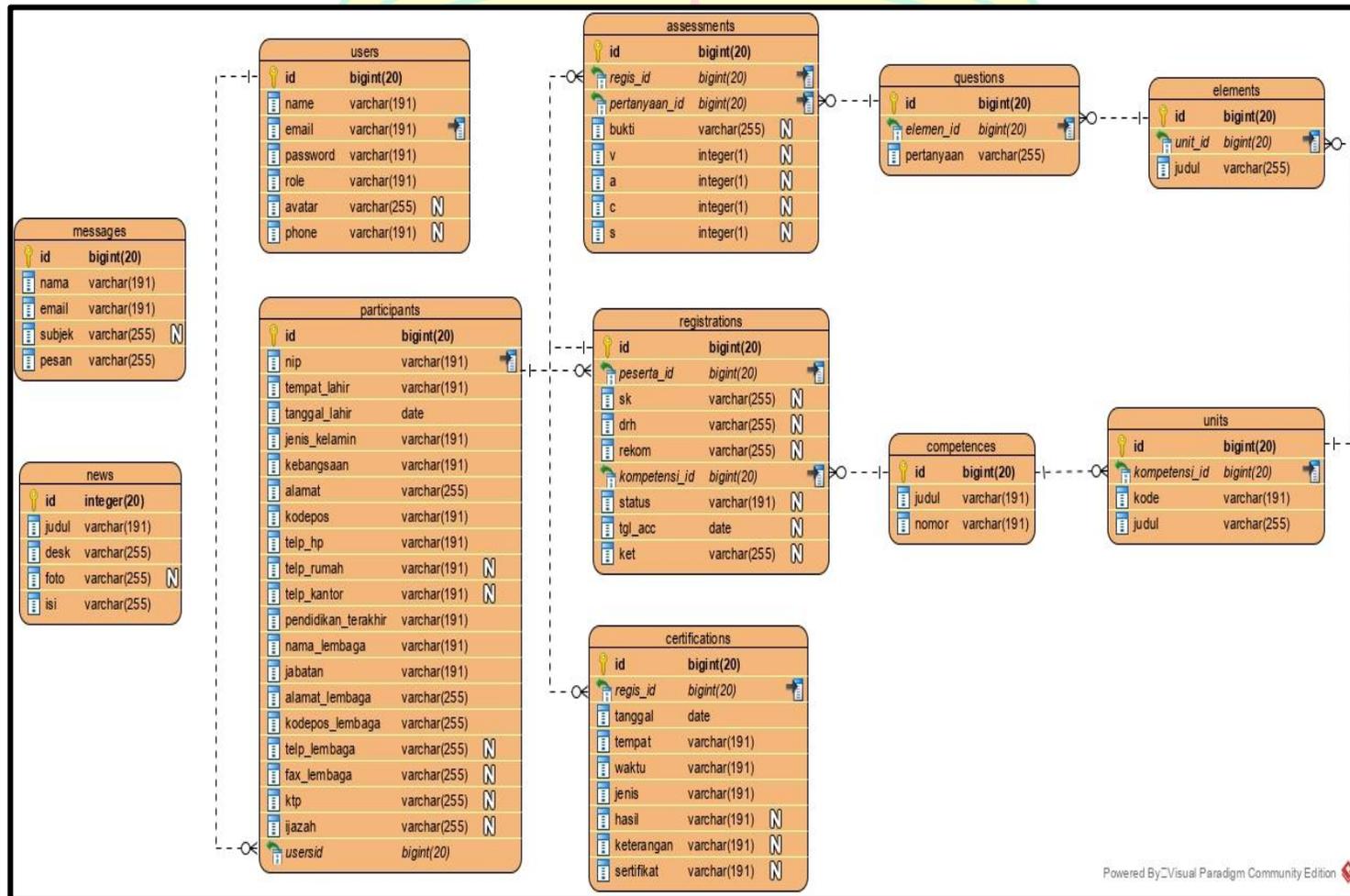
Desain Entity Relationship Diagram (ERD) merupakan diagram yang menggambarkan alur penyimpanan data pada sistem yang dikembangkan. ERD sistem informasi sertifikasi ini memiliki 11 entitas yaitu *users*, *news*, *messages*, *participants*, *registrations*, *certifications*, *assessments*, *questions*, *competences*, *units*, dan *elements*.

Pada entitas *users* terdapat data peserta, asesor dan admin. Di dalam entitas *news* merupakan tempat penyimpanan data berita yang dikelola oleh admin dan akan ditampilkan pada *website*. Pada entitas *messages* terdapat data pesan yang masuk dari pengunjung, asesor, ataupun peserta yang akan

dikelola oleh admin. *Participants* merupakan tabel yang menyimpan data pengajuan sertifikasi yang telah dilakukan oleh peserta beserta data dokumen wajib yang telah diunggah seperti ktp dan ijazah. *Registrations* adalah tabel yang menyimpan dokumen wajib yang telah diunggah peserta pada saat pengajuan, seperti sk, drh, rekom dan juga menyimpan data status pengajuan yang diberikan oleh admin. Untuk tabel *certifications* adalah tabel yang menyimpan hasil sertifikasi peserta yang dimasukkan oleh asesor dan data sertifikat peserta kompeten yang diunggah oleh admin.

Tabel *assesments* merupakan tabel yang menyimpan data penilaian asesor untuk asesmen mandiri peserta dan pada tabel ini tersimpan data bukti pendukung yang telah diunggah oleh peserta. Tabel *competences* berisi data kompetensi yang akan dipilih peserta pada saat pengajuan. Tabel *units* berisi data unit disetiap kompetensi yang tersedia. Tabel *elements* berisikan data elemen disetiap unit pada kompetensi yang tersedia. Tabel *question* merupakan tabel yang berisi pertanyaan-pertanyaan yang akan diajukan saat peserta mengisi asesmen mandiri sesuai dengan kompetensi yang telah dipilih.

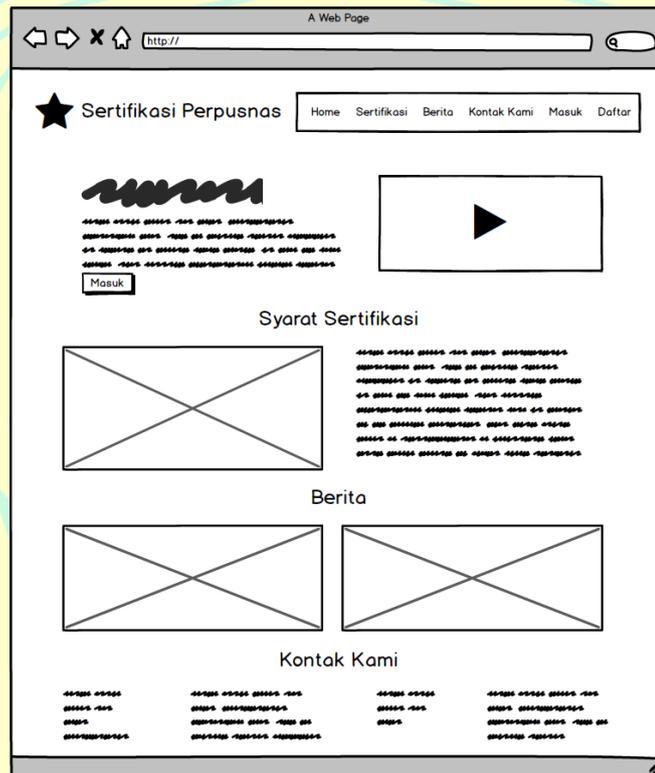
Berikut adalah desain ERD dari sistem informasi sertifikasi pustakawan:



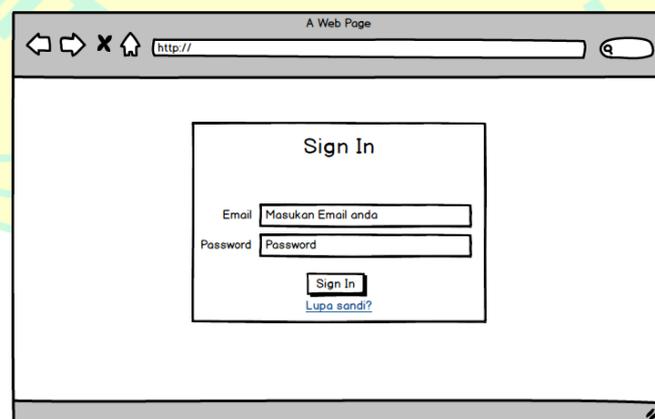
Gambar 3.8: Entity Relationship Diagram Sistem Informasi Sertifikasi

3.2.5 Desain User Interface

Pada tahap ini penulis akan membuat rancangan antar muka program atau *mockup*. Pada tampilan awal *login*, *user* akan memasukkan *e-mail* dan *password* yang telah dikirimkan oleh sistem melalui *e-mail* saat *user* melakukan registrasi.

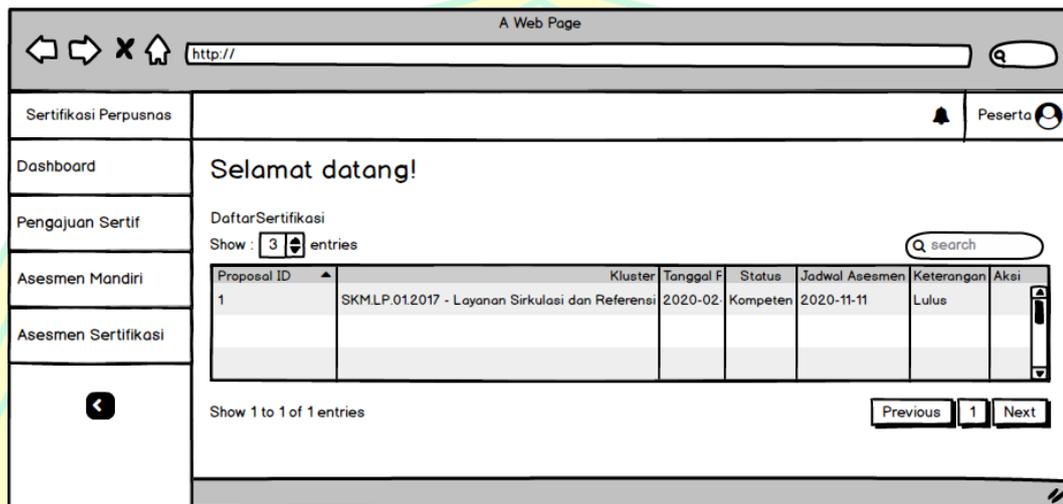


Gambar 3.9: Desain Tampilan Luar Sertifikasi



Gambar 3.10: Desain Halaman Login

Kemudian saat *user* berhasil *login*, maka sistem akan menampilkan *dashboard* yang berbeda antara peserta, asesor dan admin. Untuk menu pada *dashboard* peserta terdiri dari 4 menu yaitu, menu dashboard, pengajuan sertifikasi, asesmen mandiri, asesmen sertifikasi seperti berikut:



Gambar 3.11: Desain Halaman Dashboard Peserta

Menu pengajuan sertifikasi adalah menu untuk melakukan pengajuan sertifikasi peserta. Pada menu asesmen mandiri, peserta diharuskan mengisi beberapa pertanyaan sesuai dengan klaster kompetensi yang telah dipilih sebelumnya pada saat pengajuan. Tetapi menu ini tidak bisa dibuka oleh peserta jika peserta belum melakukan pengajuan sertifikasi. Pada menu asesmen sertifikasi, peserta dapat melihat jadwal ujian kompetensi yang akan diikuti dan melihat hasil ujian yang telah dimasukkan oleh asesor serta dapat mengunduh sertifikasi jika dinyatakan kompeten.

Perbedaan untuk asesor dan admin yaitu tidak terdapat menu untuk mengajukan pengajuan sertifikasi. Pada asesor hanya terdapat tiga menu, *dashboard*, asesmen mandiri dan asesmen sertifikasi. Pada admin terdapat tujuh menu yaitu, *dashboard*, daftar pengajuan, asesmen sertifikasi, manajemen soal, manajemen *user*, berita, dan pesan.

Pada menu pengajuan sertifikasi peserta diharuskan mengisi formulir yang berisi data diri, data pekerjaan, data permohonan sertifikasi dan dokumen wajib untuk diverifikasi oleh admin.

Sertifikasi Perpusnas		Peserta	
Dashboard	Pengajuan Sertifikasi		
Pengajuan Sertifikasi	Data Diri		
Asesmen Mandiri	NIP	<input type="text"/>	
Asesmen Sertifikasi	Nama Lengkap	<input type="text"/>	
	Tempat Lahir	<input type="text"/>	
	Tanggal Lahir	<input type="text"/> / <input type="text"/> / <input type="text"/>	
	Jenis Kelamin	Perempuan	
	Kebangsaan	WNI	
	Alamat	<input type="text"/>	
	Kode Pos	<input type="text"/>	
	No tlp rumah	<input type="text"/>	
	No tlp kantor	<input type="text"/>	
	No hp	<input type="text"/>	
	Pend Terakhir	SMA	
	Data Pekerjaan Sekarang		
	Nama Lembaga	<input type="text"/>	
	Jabatan	<input type="text"/>	
	Alamat	<input type="text"/>	
	Kode Pos	<input type="text"/>	
	No tlp	<input type="text"/>	
	No fax	<input type="text"/>	
	Data Permohonan Sertifikasi		
	Tujuan Ases	Sertifikasi	
	Skema Sertif	SKMLP.01.2017 - Layanan Sirkul	
	Dokumen Wajib		
	KTP	<input type="text"/> Upload	
	Ijazah	<input type="text"/> Upload	
	SK	<input type="text"/> Upload	
	Surat Rekomen	<input type="text"/> Upload	
	DRH	<input type="text"/> Upload	
	Simpan		

Gambar 3.12: Desain Halaman Daftar Pengajuan Sertifikasi Bagian Data Diri

Setelah data diri dan data pekerjaan diisi, selanjutnya *user* akan mengunggah beberapa dokumen persyaratan. Dokumen persyaratan terdiri dari KTP, Ijazah, Surat Keterangan Bekerja, Surat Rekomendasi Pekerjaan, Surat DRH. Peserta diharuskan memilih klaster kompetensi sertifikasi yang akan diikuti. Contoh pada klaster pengadaan bahan perpustakaan peserta harus menyiapkan lima bukti pendukung seperti bukti untuk mengoperasikan komputer, bukti menyusun rencana kerja, bukti membuat laporan kerja, bukti

melakukan seleksi bahan perpustakaan, dan bukti melakukan pengadaan bahan perpustakaan. Bukti ini dapat berbentuk jpeg, pdf ataupun doc.

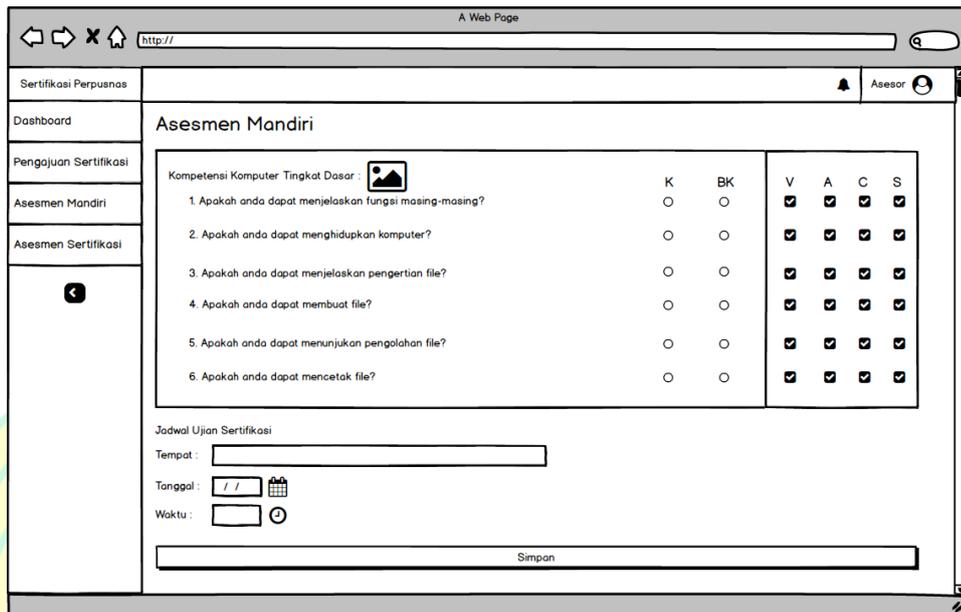
Setelah peserta mengajukan pengajuan. Admin akan memeriksa apakah berkas yang diajukan apakah sudah sesuai dan benar menurut persyaratan atau belum. Kemudian admin akan menentukan diterima, dikembalikan atau ditolaknya suatu pengajuan. Peserta baru bisa mengisi asesmen mandiri jika pemberkasas telah dinyatakan diterima oleh admin. Peserta akan mengukur dirinya sendiri dengan mengisi soal asesmen mandiri. Jika peserta merasa belum kompeten pada salah satu pertanyaan yang diajukan di dalam asesmen mandiri, maka peserta tidak bisa lanjut ke tahap asesmen sertifikasi.

Kompetensi Komputer Tingkat Dasar : Upload		K	BK
1. Apakah anda dapat menjelaskan fungsi masing-masing?	<input type="radio"/>	<input type="radio"/>	
2. Apakah anda dapat menghidupkan komputer?	<input type="radio"/>	<input type="radio"/>	
3. Apakah anda dapat menjelaskan pengertian file?	<input type="radio"/>	<input type="radio"/>	
4. Apakah anda dapat membuat file?	<input type="radio"/>	<input type="radio"/>	
5. Apakah anda dapat menunjukan pengolahan file?	<input type="radio"/>	<input type="radio"/>	
6. Apakah anda dapat mencetak file?	<input type="radio"/>	<input type="radio"/>	

Submit

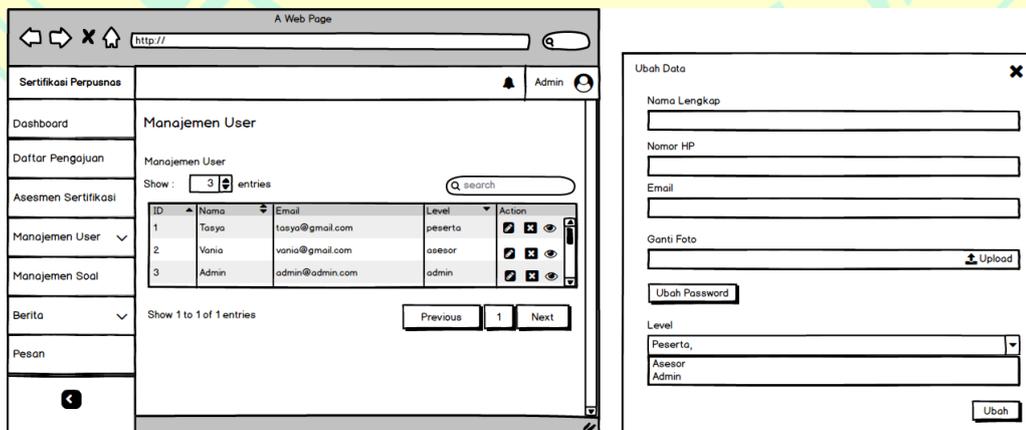
Gambar 3.13: Desain Tampilan Asesmen Mandiri Peserta

Setelah peserta mengerjakan asesmen mandiri, asesor akan menilai dengan indikator VACS.

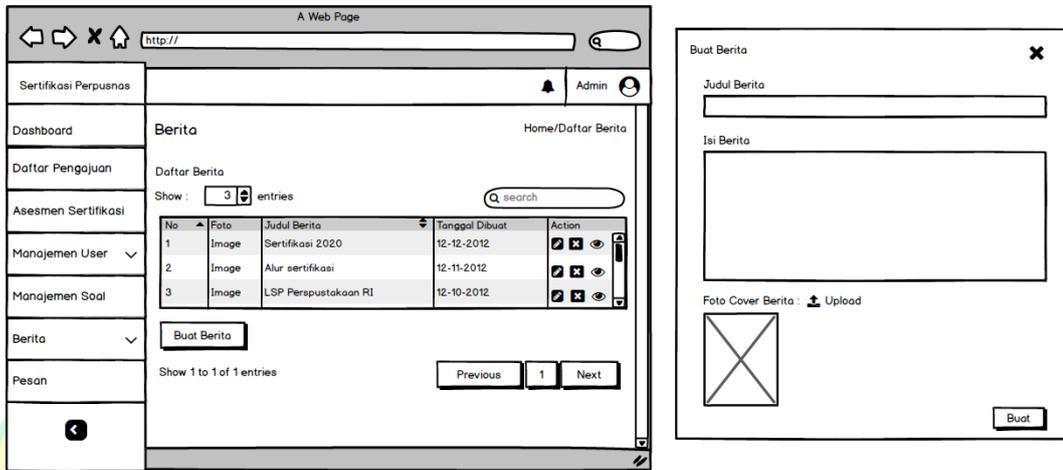


Gambar 3.14: Desain Tampilan Penilaian Asesmen Mandiri

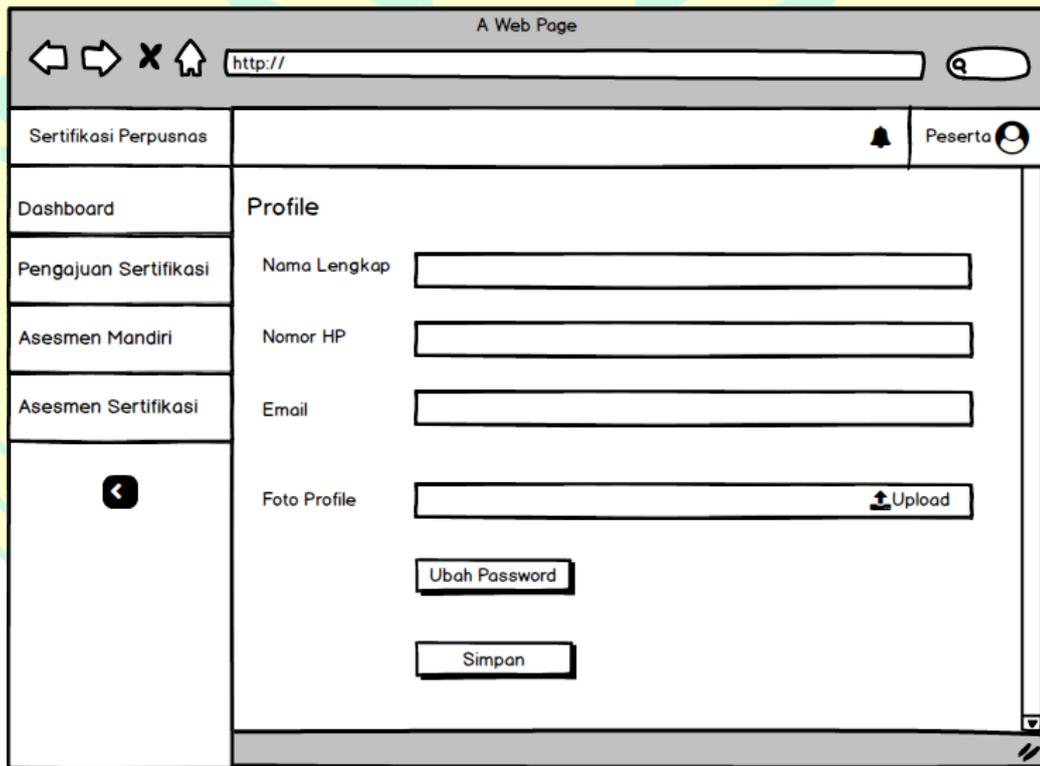
Pada gambar 3.14, selain melakukan penilaian, asesor akan menentukan jadwal ujian sertifikasi peserta. Setelah itu peserta akan mengikuti ujian secara *offline*. Setelah peserta mengikuti ujian sertifikasi pada jadwal yang telah ditentukan, asesor akan memasukkan nilai peserta berupa kompeten atau belum kompeten ke dalam sistem. Sertifikat baru bisa didapat peserta jika asesor menyatakan peserta kompeten. Apabila peserta belum kompeten maka peserta harus mengajukan sertifikasi ulang. Admin dapat mengelola *user*, soal, pesan dan berita pada *website*.



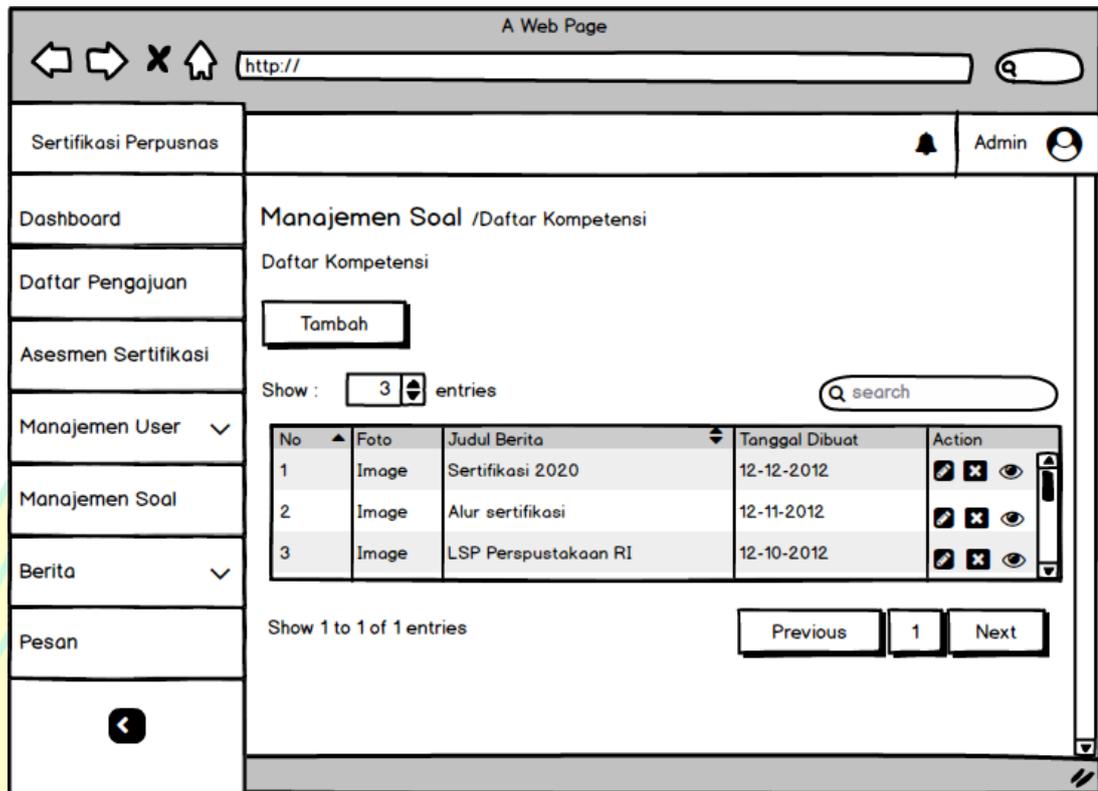
Gambar 3.15: Desain Tampilan Manajemen user



Gambar 3.16: Desain Tampilan Berita



Gambar 3.17: Desain Tampilan Menu User



Gambar 3.18: Desain Tampilan Menu Manajemen Soal

Selanjutnya untuk teknologi yang digunakan oleh penulis dalam pengembangan sistem sertifikasi ini adalah sebagai berikut:

1. Bahasa Pemrograman : PHP

Bahasa Pemrograman yang digunakan adalah PHP (*Hypertext Preprocessor*), bahasa ini digunakan oleh penulis karena bahasa PHP adalah bahasa yang standar untuk programmer web di seluruh dunia untuk membuat dan mengembangkan situs web, bahasa ini dapat digunakan bersama-sama dengan HTML. Bahasa PHP adalah bahasa yang tidak perlu dikompilasi dan dengan mudah dipasangkan pada server web yang mendukung PHP seperti Apache dengan konfigurasi yang mudah. PHP dapat digunakan secara gratis dan *open source*. Bahasa ini juga dapat dieksekusi diberbagai jenis sistem operasi seperti Windows, Linux dan Macintosh.

2. Database Engine : MySQL

Mesin pengelola database yang digunakan penulis adalah MySQL. Penulis menggunakan MySQL sebagai mesin database karena memiliki beberapa kelebihan seperti bersifat *open source* sehingga lebih mudah untuk digunakan, kemudian MySQL memiliki struktur tabel yang mudah dipakai dan fleksibel, juga dapat mendukung berbagai macam data seperti *float*, *integer*, *date*, dan lainnya. MySQL juga mendukung integrasi dengan bahasa pemrograman lain salah satunya PHP. MySQL juga bekerja sangat baik dan cepat.

3. Frontend Scripting : Bootstrap

Untuk *frontend* penulis menggunakan Bootstrap. Penulis menggunakan Bootstrap karena memiliki *library* dengan berbagai macam *script* yang siap digunakan sehingga penulis dapat menghemat waktu dalam mengembangkan sistem. Bootstrap juga memiliki sifat fleksibel karena bootstrap tidak memiliki ketentuan khusus untuk menggunakan *script*. Bootstrap memiliki desain yang berorientasi karena adanya *Grid*, yang memudahkan penulis dalam mengatur objek di dalam sistem. Bootstrap juga dilengkapi dengan *library* JavaScript yang sesuai dengan struktur dari bahasa pemrograman Javascript standar.

4. Backend Scripting : Laravel

Untuk keperluan *backend* penulis menggunakan Laravel. Laravel digunakan penulis karena memiliki dukungan *framework* MVC sehingga dapat dengan mudah mengelola basis data seperti memasukkan data ke basis data, melakukan pembaruan data dan pengolahan data lainnya. Laravel juga memiliki *library object oriented* yang merupakan nilai tambahan dibandingkan dengan *framework* lainnya, contohnya *library authentication* dimana tersedia *hashing* dalam *library* ini yang merupakan salah satu fitur tercanggih di bidang otentifikasi saat ini. Tersedianya fitur artisan juga merupakan salah satu alasan penulis memilih laravel sebagai *Backend*

Scripting karena dengan adanya artisan ini memudahkan penulis dalam memasukkan perintah kedalam Laravel untuk membangun dan mengembangkan *website*.

Semua teknologi yang sudah dijelaskan sebelumnya merupakan teknologi yang dipilih berdasarkan dari hasil pengalaman dan observasi yang telah dilalui oleh penulis sebelum dan selama penelitian ini berlangsung. Sehingga penulis dapat menyakinkan bahwa teknologi tersebut dapat diterapkan dengan baik pada sistem sertifikasi yang akan dibangun dan dikembangkan nantinya.

3.3 Pengembangan

Pada tahap pengembangan semua desain yang telah dirancang sebelumnya akan diubah kedalam kode pemrograman. Dimulai dengan membuat *database*, kemudian mengimplementasikan desain tampilannya (*Front End*) dan terakhir membangun sistem (*Back End*).

3.3.1 Membangun Database

Pada tahap ini dibuat *database* berdasarkan desain ERD yang telah dibuat pada desain sistem. *Database* dibuat menggunakan MySQL dan memanfaatkan aplikasi phpMyAdmin. Berikut bentuk *database* Sistem Informasi Sertifikasi Pustakawan Perpunas RI yang memiliki 14 tabel terdapat pada gambar 3.19 berikut:

Table	Action	Rows	Type	Collation	Size	Overhead
assessments	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
certifications	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
competences	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
elements	★ Browse Structure Search Insert Empty Drop	72	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
messages	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	★ Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
news	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
participants	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
password_resets	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
questions	★ Browse Structure Search Insert Empty Drop	238	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
registrations	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
units	★ Browse Structure Search Insert Empty Drop	27	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
users	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
14 tables	Sum	360	MyISAM	latin1_swedish_ci	464.0 KiB	0 B

Gambar 3.19: Database Sistem Informasi Sertifikasi Pustakawan Perpunas RI

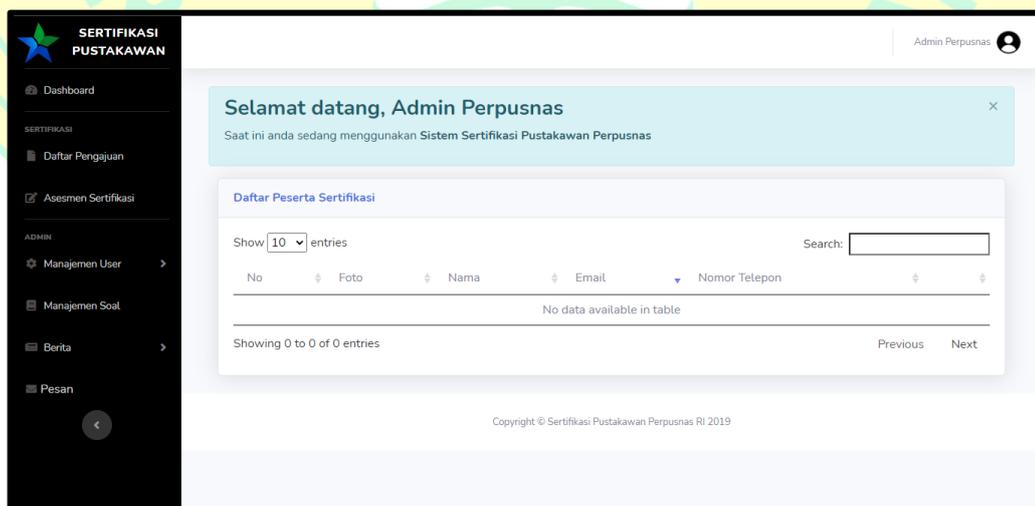
3.3.2 Implementasi Desain Tampilan

Pada tahap ini, penulis menggunakan bootstrap 4.0 sebagai *framework* untuk membuat tampilan atau *front-end* dan Laravel sebagai *framework* untuk *back-end*. Berikut tampilan *website* Sistem Informasi Sertifikasi Pustakawan Perpustakaan Nasional RI yang dapat dilihat pada browser.



Gambar 3.20: Tampilan Halaman Utama Untuk Pengunjung

Tampilan awal yang ditampilkan yaitu halaman beranda untuk pengunjung. Jika, pengunjung ingin mengajukan sertifikasi maka pengunjung harus mengisi formulir dengan memasukkan nama dan *e-mail* yang digunakan. Sistem akan mengirimkan *username* dan *password* ke alamat *e-mail* yang telah didaftarkan.



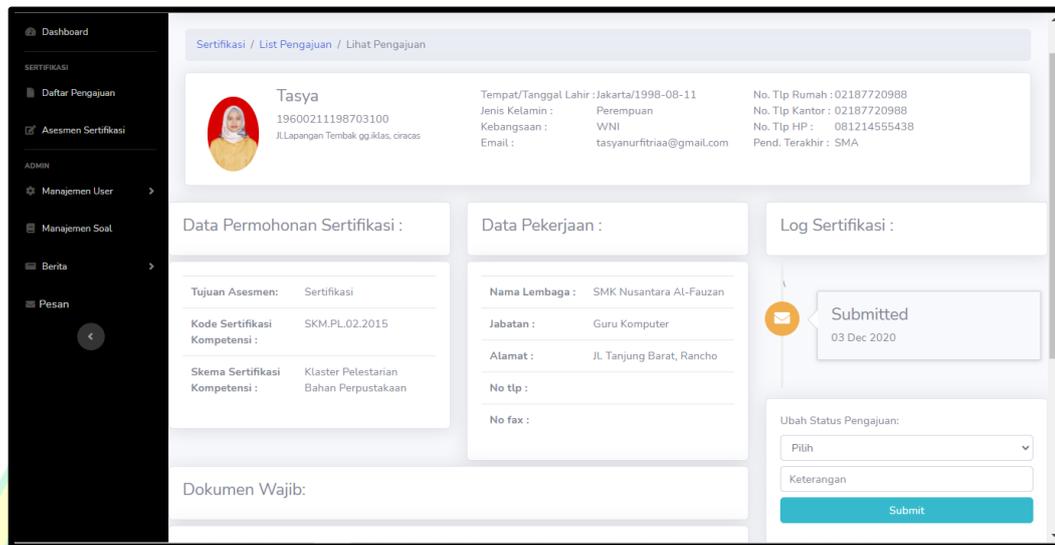
Gambar 3.21: Tampilan Halaman Untuk Beranda

Pada gambar 3.21 adalah beranda setelah *login* berhasil dilakukan untuk admin dan terdapat 7 submenu yaitu *dashboard*, daftar pengajuan, asesmen sertifikasi, manajemen *user*, manajemen soal, berita dan pesan. Untuk asesor terdapat 3 submenu yaitu *dashboard*, asesmen mandiri, dan asesmen sertifikasi. Untuk peserta terdapat 4 submenu yaitu *dashboard*, pengajuan sertifikasi, asesmen mandiri dan asesmen sertifikasi.

Gambar 3.22 merupakan tampilan halaman peserta untuk melakukan pengajuan sertifikasi. Peserta harus mengisi data diri, data pekerjaan dan mengunggah beberapa dokumen wajib yang diminta oleh sistem, seperti ktp, drh, dan lainnya. Peserta juga akan memilih klaster kompetensi sertifikasi yang ingin diikuti. Terdapat 5 klaster kompetensi yang disediakan oleh perpunas. Salah satu contohnya yaitu klaster pengatalogan bahan pustaka.

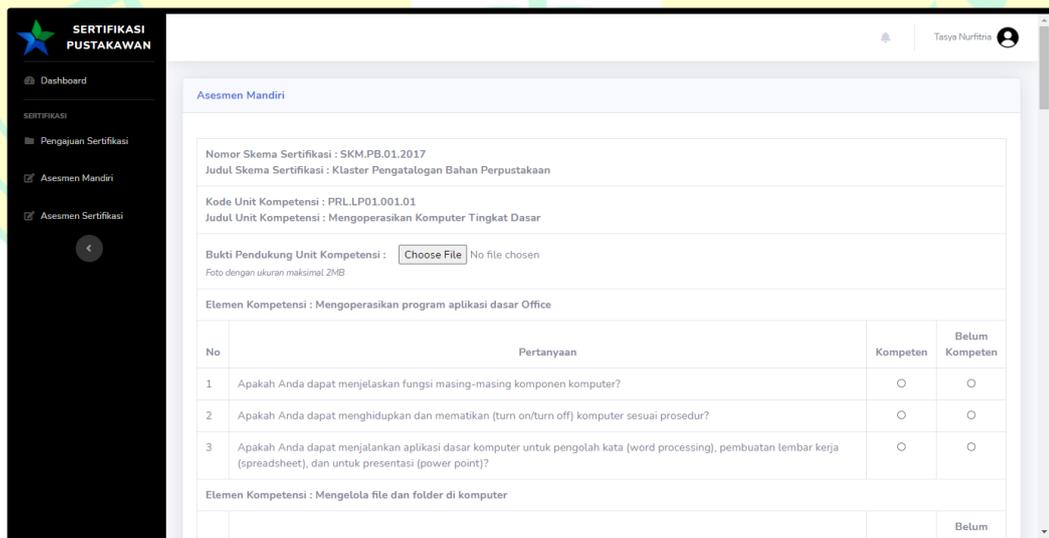
Data Pribadi		Data Pekerjaan Sekarang	
NIP *	<input type="text" value="NIP"/>	Nama Lembaga *	<input type="text" value="Masukkan nama lembaga/perusahaan"/>
Tempat lahir *	<input type="text" value="Tempat Lahir"/>	Jabatan *	<input type="text" value="Masukkan jabatan terakhir anda"/>
Tanggal Lahir *	<input type="text" value="dd/mm/yyyy"/>	Alamat *	<input type="text" value="Masukkan alamat lembaga/perusahaan"/>
Jenis Kelamin *	<input type="text" value="Pilih"/>	Kodepos *	<input type="text" value="Kodepos"/>
Kebangsaan *	<input type="text" value="Pilih"/>	No Telp	<input type="text" value="Masukkan nomor telepon lembaga/perusahaan"/>
Alamat *	<input type="text" value="Masukkan alamat rumah"/>	No Fax	<input type="text" value="Masukkan nomor fax lembaga/perusahaan"/>
Kodepos *	<input type="text" value="Kodepos"/>	Data Permohonan Sertifikasi	
		Tujuan	<input type="text" value="Sertifikasi"/>

Gambar 3.22: Tampilan Halaman Untuk Pengajuan Sertifikasi Peserta



Gambar 3.23: Tampilan Halaman Untuk Daftar Pengajuan Peserta pada Admin

Gambar 3.23 merupakan tampilan untuk halaman daftar pengajuan peserta pada admin. Terdapat tombol untuk menentukan status pengajuan pada halaman sebelah kanan. Admin dapat memilih untuk merubah status pengajuan menjadi diterima, dikembalikan, atau ditolak dan dapat memberikan keterangan. Di halaman tersebut admin juga dapat mengunduh dokumen wajib peserta yang telah diunggah oleh peserta pada saat pengajuan.



Gambar 3.24: Tampilan Halaman Untuk Asesmen Mandiri Peserta

Pada gambar 3.24 terdapat halaman untuk pengerjaan asesmen mandiri pada sistem peserta. Asesmen mandiri terdiri dari pertanyaan-pertanyaan yang sesuai dengan pilihan klaster kompetensi yang telah dipilih peserta pada saat pengajuan sertifikasi. Pada halaman ini peserta dapat mengunggah bukti-bukti pendukung disetiap kompetensinya. Bukti yang dilampirkan tidak boleh melebihi kapasitas yang telah diberitahukan oleh sistem. Peserta dapat mengunggah file berupa PDF, DOC ataupun JPG. Kemudian peserta dapat memilih jawaban berupa kompeten atau belum kompeten disetiap pertanyaan yang diajukan oleh sistem.

Gambar 3.25 adalah tampilan halaman asesor saat akan melakukan penilaian terhadap asesmen mandiri yang telah dikerjakan oleh peserta sebelumnya, di halaman ini asesor dapat melihat dan mengunduh bukti yang telah diunggah oleh peserta. Asesor dapat memilih VACS (*Valid, Authentic, Current, Sufficient*) sesuai dengan bukti yang telah diunggah oleh peserta pada setiap kompetensinya. Asesor juga dapat langsung menentukan jadwal untuk asesmen sertifikasi yang akan diikuti peserta dalam tiga tahapan yaitu, tertulis berupa pilih ganda, wawancara, dan demo yang akan dilakukan secara *offline* oleh peserta.

Elemen Kompetensi : Melakukan verifikasi pengalokasian subyek

No	Pertanyaan	Kompeten	V	A	C	S
1	Apakah Anda dapat memverifikasi notasi kelas?	<input checked="" type="radio"/> Sudah <input type="radio"/> Belum	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Apakah Anda dapat memperbaiki notasi kelas yang tidak tepat?	<input checked="" type="radio"/> Sudah <input type="radio"/> Belum	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Apakah Anda dapat memperbaiki notasi kelas yang tidak tepat?	<input checked="" type="radio"/> Sudah <input type="radio"/> Belum	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

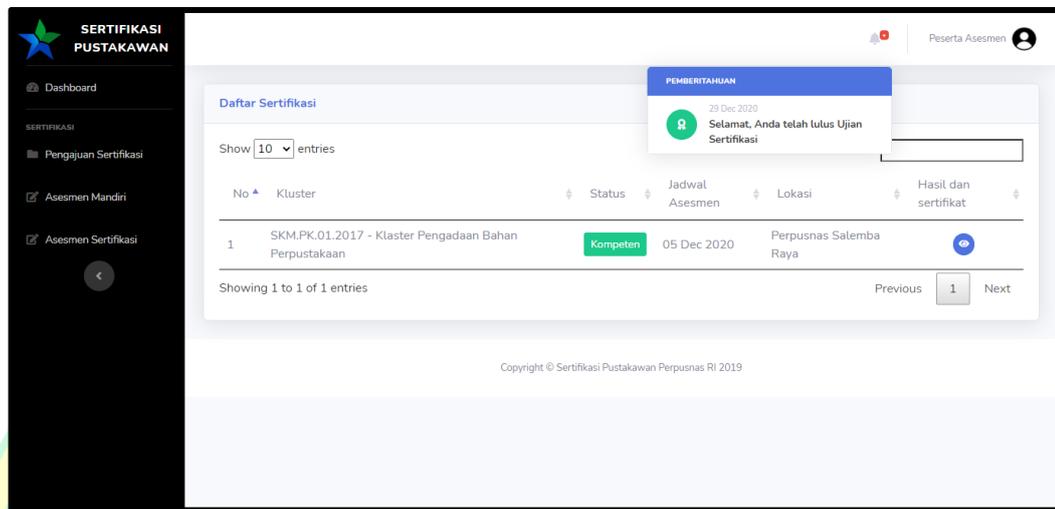
Jadwal Ujian Sertifikasi :

Tempat :

Tanggal: 

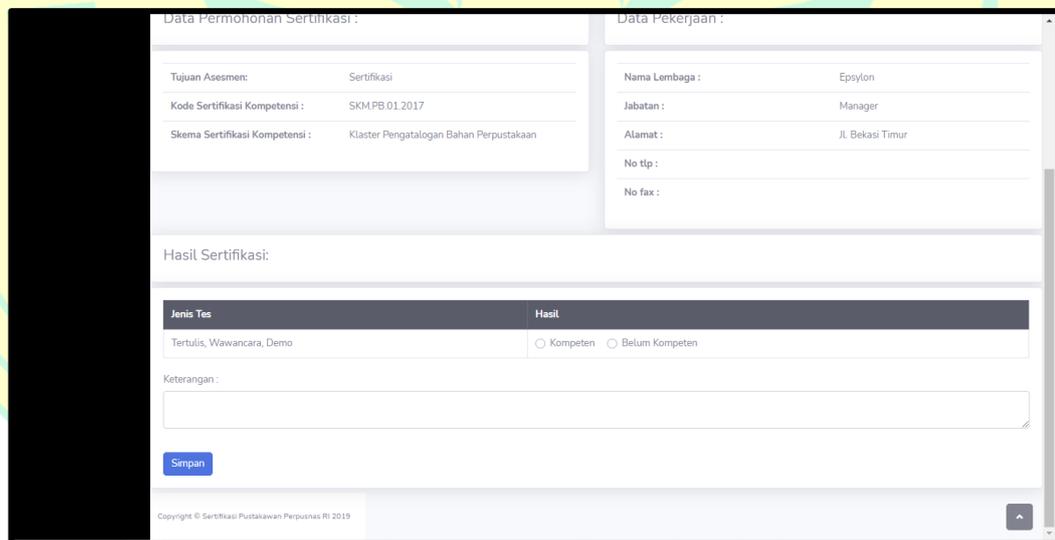
Waktu : 

Gambar 3.25: Tampilan Halaman Penilaian Asesmen Mandiri pada Asesor



Gambar 3.26: Tampilan Halaman Asesmen Sertifikasi pada Peserta

Pada gambar 3.26 peserta akan mendapatkan notifikasi apabila jadwal sertifikasi telah ditentukan oleh asesor dan pada halaman ini peserta dapat melihat hasil asesmen mandiri dengan menekan tombol biru di sebelah tabel.



Gambar 3.27: Tampilan Halaman Penilaian Asesmen Sertifikasi pada Asesor

Gambar 3.27 adalah tampilan halaman pada asesor ketika asesor akan melakukan penilaian untuk asesmen sertifikasi yang telah dilakukan peserta secara *offline*. Asesor dapat memilih hasil sertifikasi berupa kompeten atau

belum kompeten, dan memasukkan beberapa keterangan untuk ditujukan kepada peserta jika diperlukan.

The screenshot shows the Admin interface for participant certification assessment. It features a sidebar menu with options like 'Manajemen User', 'Manajemen Soot', 'Berita', and 'Pesan'. The main content area is divided into two columns: 'Data Permohonan Sertifikasi' and 'Data Pekerjaan'. The 'Data Permohonan Sertifikasi' section includes fields for 'Tujuan Asesmen' (Sertifikasi), 'Kode Sertifikasi Kompetensi' (SKM PB.01.2017), and 'Skema Sertifikasi Kompetensi' (Klaster Pengatalogan Bahan Perpustakaan). The 'Data Pekerjaan' section includes fields for 'Nama Lembaga' (Epsilon), 'Jabatan' (Manager), 'Alamat' (JL Bekasi Timur), 'No tlp', and 'No fax'. Below these sections is the 'Hasil Sertifikasi' section, which contains a table with columns 'Jenis Tes' and 'Hasil'. The table shows a test type 'Tertulis, Wawancara, Demo' with a result of 'Kompeten'. Below the table is a 'Keterangan' section with a message: 'Selamat anda dinyatakan lulus sertifikasi oleh BNSP. Sertifikat akan segera diterbitkan. Terimakasih'. There is also an 'Upload Sertifikat' section with a 'Choose File' button and a 'Simpan' button.

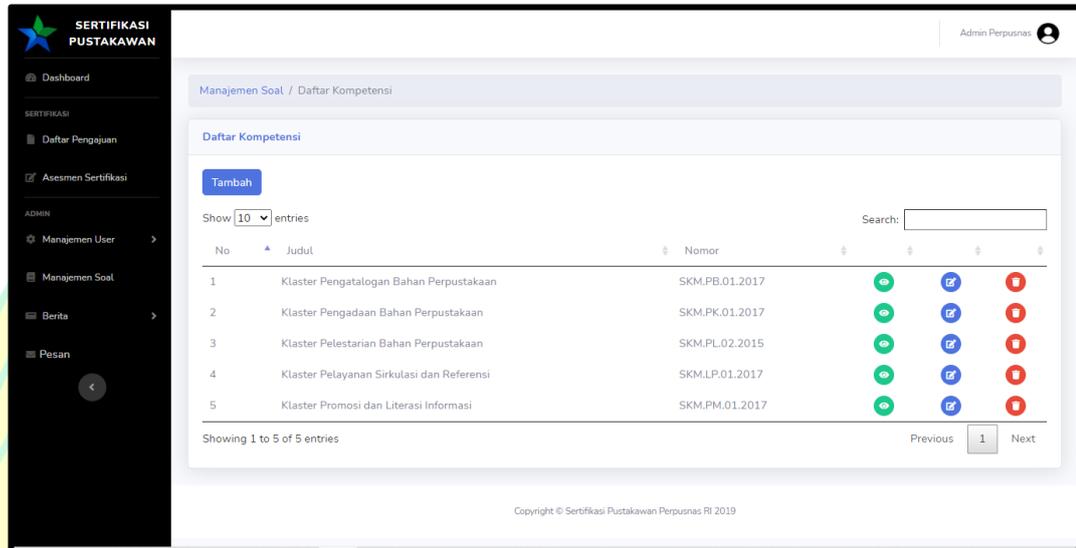
Gambar 3.28: Tampilan Halaman Asesmen Sertifikasi Peserta pada Admin

Gambar 3.28 merupakan tampilan pada halaman admin ketika akan mengunggah sertifikat untuk peserta yang telah dinyatakan kompeten oleh asesor. Sertifikat yang diunggah dapat berbentuk PDF ataupun JPEG/JPG.

The screenshot shows the Participant interface for certification assessment. It features a sidebar menu with options like 'Manajemen User', 'Manajemen Soot', 'Berita', and 'Pesan'. The main content area is divided into two columns: 'Data Permohonan Sertifikasi' and 'Data Pekerjaan'. The 'Data Permohonan Sertifikasi' section includes fields for 'Tujuan Asesmen' (Sertifikasi), 'Kode Sertifikasi Kompetensi' (SKM PB.01.2017), and 'Skema Sertifikasi Kompetensi' (Klaster Pengatalogan Bahan Perpustakaan). The 'Data Pekerjaan' section includes fields for 'Nama Lembaga' (Epsilon), 'Jabatan' (Manager), 'Alamat' (JL Bekasi Timur), 'No tlp', and 'No fax'. Below these sections is the 'Hasil Sertifikasi' section, which contains a table with columns 'Jenis Tes' and 'Hasil'. The table shows a test type 'Tertulis, Wawancara, Demo' with a result of 'Kompeten'. Below the table is a 'Keterangan' section with a message: 'Selamat anda dinyatakan lulus sertifikasi oleh BNSP. Sertifikat akan segera diterbitkan. Terimakasih'. There is also an 'Upload Sertifikat' section with a 'Choose File' button and a 'Simpan' button. The certificate image shows the text: 'CERTIFICATE OF APPRECIATION', 'Muhammad Yusuf Al Hafizi', and 'Nama Diri', 'Jabatan'.

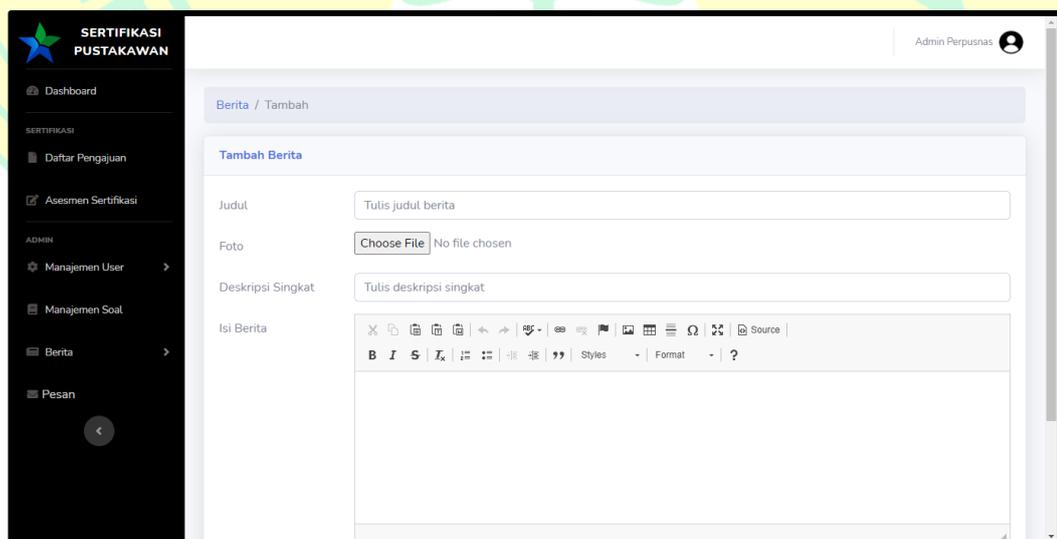
Gambar 3.29: Tampilan Halaman Hasil Asesmen Sertifikasi pada Peserta

Pada gambar 3.29 peserta dapat melihat hasil ujian sertifikasi yang telah dimasukkan oleh asesor. Pada halaman ini juga peserta dapat melihat dan mengunduh sertifikat yang telah diunggah oleh admin.



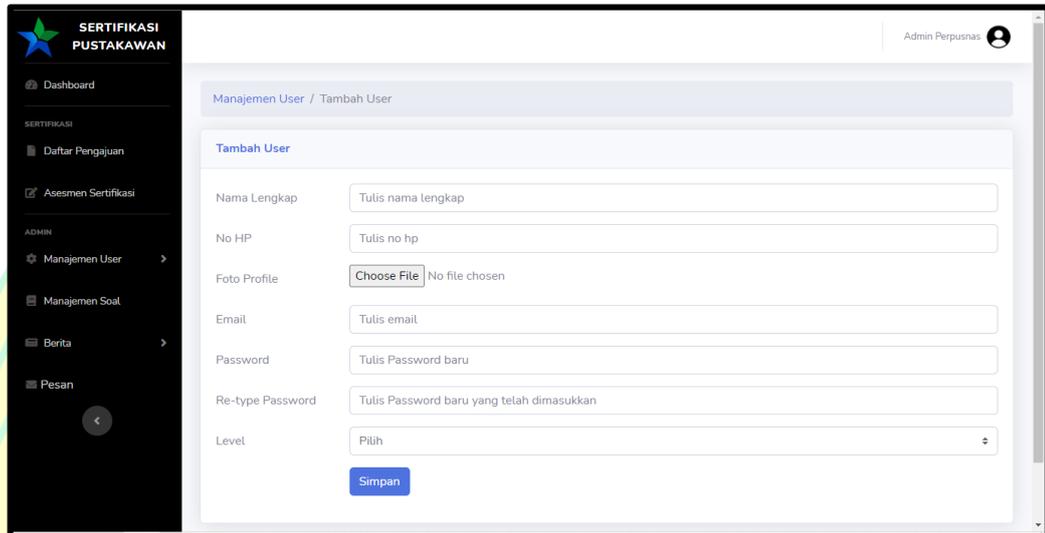
Gambar 3.30: Tampilan Halaman Manajemen Soal pada Admin

Pada gambar 3.30 merupakan tampilan manajemen soal asesmen mandiri yang hanya bisa dilakukan oleh admin. Pada halaman ini admin dapat melihat, memperbarui, dan menghapus kompetensi, unit, elemen, dan pertanyaan.



Gambar 3.31: Tampilan Halaman Berita pada Admin

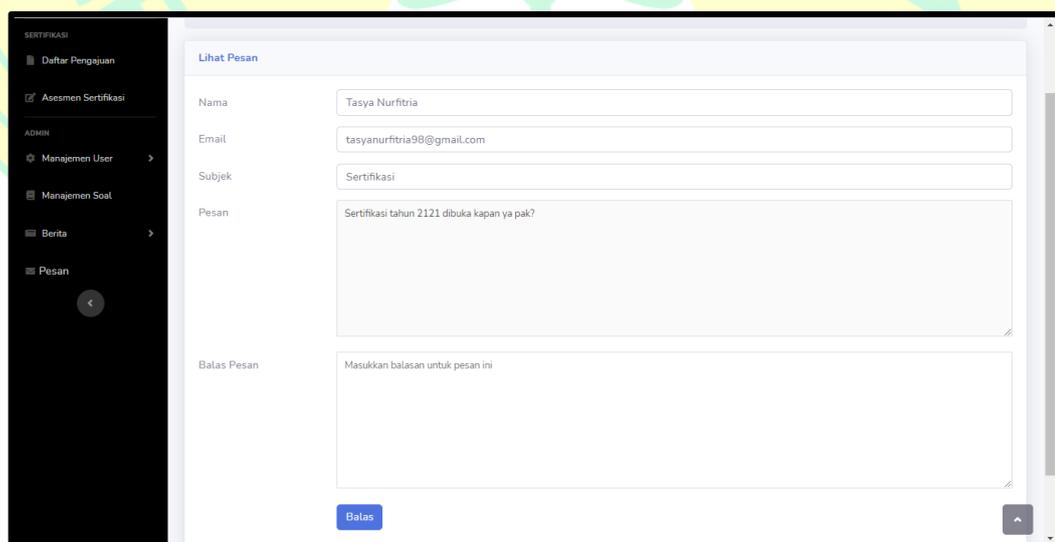
Pada gambar 3.31 adalah tampilan halaman berita pada admin. Di halaman ini admin dapat membuat, memperbarui, dan menghapus berita artikel yang akan ditampilkan di beranda depan *website*.



The screenshot shows the 'Manajemen User / Tambah User' page. The left sidebar contains the 'ADMIN' menu with 'Manajemen User' selected. The main content area is a form titled 'Tambah User' with the following fields: 'Nama Lengkap' (text input), 'No HP' (text input), 'Foto Profile' (file upload), 'Email' (text input), 'Password' (text input), 'Re-type Password' (text input), and 'Level' (dropdown menu). A 'Simpan' button is at the bottom.

Gambar 3.32: Tampilan Halaman Manajemen *User* pada Admin

Pada gambar 3.32 merupakan tampilan halaman untuk manajemen *user* pada admin. Di menu manajemen *user* ini admin dapat menambah, memperbarui, dan menghapus *user*. Di halaman ini juga admin dapat memberikan akses kepada *user*.



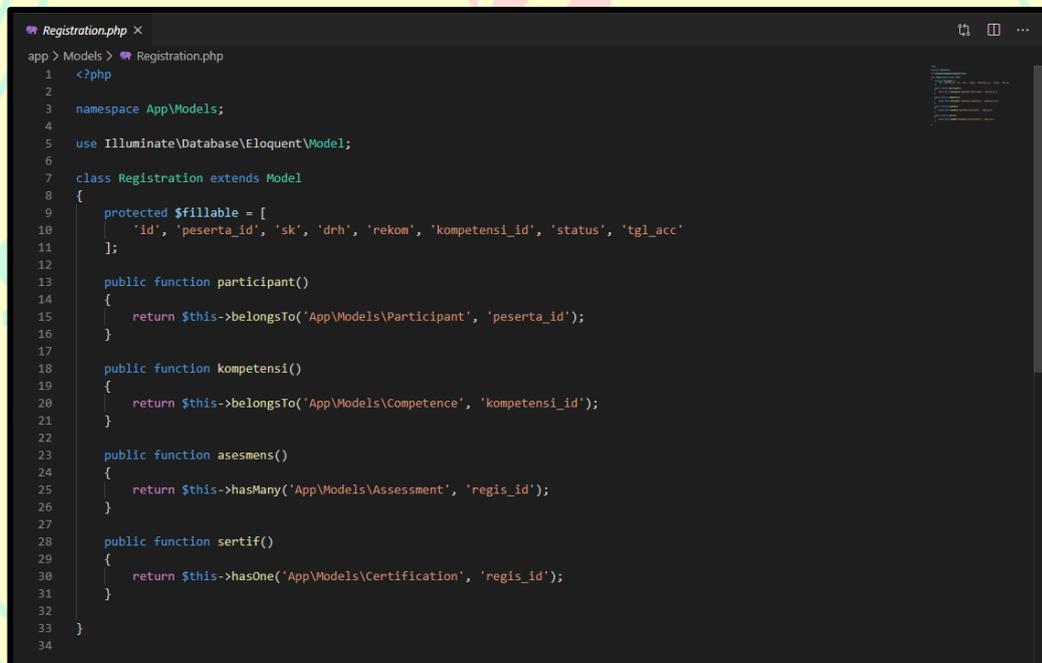
The screenshot shows the 'Lihat Pesan' page. The left sidebar contains the 'ADMIN' menu with 'Manajemen User' selected. The main content area displays message details for 'Tasya Nurfitri' with email 'tasyanurfitri98@gmail.com' and subject 'Sertifikasi'. The message body contains the text 'Sertifikasi tahun 2121 dibuka kapan ya pak?'. There is a 'Balas Pesan' section with a text input field and a 'Balas' button.

Gambar 3.33: Tampilan Halaman Pesan pada Admin

Pada gambar 3.33 admin dapat membalas pesan yang masuk dari pengunjung, peserta ataupun asesor. Pada menu pesan juga admin dapat melihat dan menghapus pesan yang masuk.

3.3.3 Implementasi Sistem (*Back End*)

Penulis menggunakan bahasa pemrograman PHP dan *framework* Laravel untuk memudahkan dalam mengimplementasikan arsitektur MVC (*Model, View, Controller*) pada saat penerapan fungsi-fungsi di dalam sistem. Berikut terdapat beberapa sampel kode pemrograman yang terdiri dari *Model, View, Controller*, dan *Route*.



```
Registration.php x
app > Models > Registration.php
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Registration extends Model
8 {
9     protected $fillable = [
10         'id', 'peserta_id', 'sk', 'drh', 'rekom', 'kompetensi_id', 'status', 'tgl_acc'
11     ];
12
13     public function participant()
14     {
15         return $this->belongsTo('App\Models\Participant', 'peserta_id');
16     }
17
18     public function kompetensi()
19     {
20         return $this->belongsTo('App\Models\Competence', 'kompetensi_id');
21     }
22
23     public function asesmens()
24     {
25         return $this->hasMany('App\Models\Assessment', 'regis_id');
26     }
27
28     public function sertif()
29     {
30         return $this->hasOne('App\Models\Certification', 'regis_id');
31     }
32
33 }
34
```

Gambar 3.34: Struktur Pemrograman *Model*

```

welcome.blade.php x
resources > views > welcome.blade.php
1 @extends('layouts.landing-page')
2
3 @section('title')
4 Sertifikasi Pustakawan
5 @endsection
6
7 @section('content')
8 <!-- ===== Hero Section ===== -->
9 <section id="hero" class="d-flex align-items-center">
10
11 <div class="container">
12 <div class="row">
13 <div class="col-lg-6 pt-5 pt-lg-0 order-2 order-lg-1 d-flex flex-column justify-content-center">
14 <h1 data-aos="fade-up" style="color:blue">Sertifikasi <br>Pustakawan</h1>
15 <h2 data-aos="fade-up" data-aos-delay="400">Untuk mengetahui pola dan potensi pengembangan kompetensi untuk ra
16 strategis kebijakan Perpustakaan Nasional Republik Indonesia</h2>
17 @if (Route::has('login'))
18 @auth
19 @else
20 <div data-aos="fade-up" data-aos-delay="800">
21 <a href="{{route('login')}}" class="btn-get-started scrollto">Masuk disini</a>
22 </div>
23 @endauth
24 @endif
25 </div>
26 <div class="col-lg-6 order-1 order-lg-2 hero-img" data-aos="fade-left" data-aos-delay="200">
27 <iframe class="animated" width="560" height="315" src="https://www.youtube.com/embed/JSitfKM-3W1" frameborder=
28 </div>
29 </div>
30 </div>
31
32 </section><!-- End Hero -->
33
34 <main id="main">
35

```

Gambar 3.35 : Struktur Pemrograman View

```

AdminController.php x
app > Http > Controllers > AdminController.php
263 public function add_user(Request $request){
264     $this->validate($request, [
265         'name' => ['required', 'string', 'max:255'],
266         'email' => ['required', 'string', 'email', 'max:255', 'unique:users']
267     ]);
268
269     if($request->avatar == null) {
270         $user = User::create([
271             'name' => $request['name'],
272             'phone' => $request['phone'],
273             'email' => $request['email'],
274             'role' => $request['level'],
275             'password' => Hash::make($request['password']),
276         ]);
277     }else{
278         $avatar = $request->avatar->getClientOriginalName();
279         $filename = time().$avatar;
280         $request->avatar->storeAs('public/upload', $filename);
281         $user = User::create([
282             'name' => $request['name'],
283             'phone' => $request['phone'],
284             'email' => $request['email'],
285             'role' => $request['level'],
286             'password' => Hash::make($request['password']),
287             'avatar' => $filename
288         ]);
289     }
290
291     $password = $request['password'];
292     $nama = $user->name;
293     $email = $user->email;
294     Mail::to($email)->send(new RegisMail($nama, $email, $password));
295
296     return redirect('/manajemen-user')->with('msg', 'Data berhasil ditambahkan');

```

Gambar 3.36 : Struktur Pemrograman Controller

```
web.php x
routes > web.php
20 //Admin
21 Route::middleware(['auth', 'admin'])->group(function () {
22 //beranda
23 Route::get('admin/lihat/peserta-sertifikasi/{id}', 'AdminController@lihatpeserta')->name('admin.lihat.peserta');
24 Route::get('admin/lihat/registrasi/peserta-sertifikasi/{id}', 'AdminController@regispeserta')->name('admin.regis.pes');
25 //pengajuan
26 Route::get('admin/pengajuan-sertifikasi', 'AdminController@pengajuan')->name('admin.pengajuan');
27 Route::get('admin/pengajuan-sertifikasi/lihat/{id}', 'AdminController@lihatpengajuan')->name('admin.lihat.pengajuan');
28 Route::post('admin/pengajuan-sertifikasi/konfirm/{id}', 'AdminController@konfirmpengajuan')->name('admin.konfirm.peng');
29 //sertifikasi
30 Route::get('admin/asesmen-sertifikasi', 'AdminController@asesmensertifikasi')->name('admin.sertifikasi');
31 Route::get('admin/asesmen-sertifikasi/lihat/{id}', 'AdminController@lihatasesmensertifikasi')->name('admin.lihat.ser');
32 Route::put('admin/asesmen-sertifikasi/{id}', 'AdminController@editasesmensertifikasi')->name('admin.edit.sertifikasi');
33 //manajemen-soal - kompetensi
34 Route::get('manajemen-soal', 'AdminController@kompetensi')->name('kompetensi');
35 Route::post('manajemen-soal', 'AdminController@add_kompetensi')->name('add_kompetensi');
36 Route::put('manajemen-soal/{id}', 'AdminController@edit_kompetensi')->name('edit_kompetensi');
37 Route::delete('manajemen-soal/{id}', 'AdminController@delete_kompetensi')->name('delete_kompetensi');
38 //manajemen-soal - unit
39 Route::get('manajemen-soal/kompetensi/unit/{id}', 'AdminController@unit')->name('unit');
40 Route::post('manajemen-soal/kompetensi/unit/{id}', 'AdminController@add_unit')->name('add_unit');
41 Route::put('manajemen-soal/kompetensi/unit/{id}', 'AdminController@edit_unit')->name('edit_unit');
42 Route::delete('manajemen-soal/kompetensi/unit/{id}', 'AdminController@delete_unit')->name('delete_unit');
43 //manajemen-soal - elemen
44 Route::get('manajemen-soal/kompetensi/unit/elemen/{id}', 'AdminController@elemen')->name('elemen');
45 Route::post('manajemen-soal/kompetensi/unit/elemen/{id}', 'AdminController@add_elemen')->name('add_elemen');
46 Route::put('manajemen-soal/kompetensi/unit/elemen/{id}', 'AdminController@edit_elemen')->name('edit_elemen');
47 Route::delete('manajemen-soal/kompetensi/unit/elemen/{id}', 'AdminController@delete_elemen')->name('delete_elemen');
48 //manajemen-soal - pertanyaan
49 Route::get('manajemen-soal/kompetensi/unit/elemen/pertanyaan/{id}', 'AdminController@pertanyaan')->name('pertanyaan');
50 Route::post('manajemen-soal/kompetensi/unit/elemen/pertanyaan/{id}', 'AdminController@add_pertanyaan')->name('add_pe');
51 Route::put('manajemen-soal/kompetensi/unit/elemen/pertanyaan/{id}', 'AdminController@edit_pertanyaan')->name('edit');
52 Route::delete('manajemen-soal/kompetensi/unit/elemen/pertanyaan/{id}', 'AdminController@delete_pertanyaan')->name('d');
53 //manajemen-user
```

Gambar 3.37 : Struktur Pemrograman Routing

