

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

2.1.1 Sistem

Secara etimologis, sistem berasal dari bahasa Yunani (*sustēma*) dan bahasa Latin (*systēma*) yang memiliki arti mengenai suatu kesatuan komponen atau elemen-elemen yang terhubung satu sama lain sehingga terjadinya aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Sistem adalah jaringan prosedur-prosedur yang saling berhubungan, untuk melakukan suatu kegiatan atau untuk melakukan sasaran yang tertentu. (Hutahaean, 2014)

Sistem memiliki dua pendekatan, yaitu pendekatan pada prosedur dan pendekatan pada elemen dan komponen sistem. Pendekatan pada prosedur didefinisikan dengan suatu jaringan kerja dari prosedur yang saling berhubungan, kemudian berkumpul bersama untuk melakukan kegiatan atau untuk menyelesaikan sasaran tertentu. Sedangkan prosedur dengan pendekatan elemen didefinisikan dengan sekumpulan elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu. (Nafiudin, 2019)

Secara umum, sistem dibagi menjadi tiga bagian yaitu masukan, proses, dan keluaran. Masukan terdiri dari semua elemen yang masuk ke dalam sistem. Proses terdiri dari semua elemen yang diperlukan untuk mengkonversikan masukan ke keluaran. Sedangkan, keluaran merupakan hasil produk akhir yang dikeluarkan atau yang terjadi pada suatu sistem. Sistem juga dapat dibedakan berdasarkan bentuk abstrak maupun fisik. Sistem abstrak adalah sistem dengan susunan yang teratur dan saling bergantung satu dengan yang lainnya. Sistem fisik adalah sistem dengan susunan teratur dari unsur-unsur yang berkesinambungan.

Dapat disimpulkan bahwa sistem merupakan bagian-bagian jaringan yang saling berhubungan satu sama lain dan beroperasi secara bersamaan untuk mencapai suatu tujuan yang telah ditetapkan.

2.1.2 Informasi

Secara etimologis, informasi berasal dari bahasa Perancis kuno *informacion* yang diambil dari bahasa Latin *informationem* yang berarti garis besar, konsep, ide. Menurut Gordon B. Davis dalam Jeperson Hutahean (2014) informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi penerimanya dan memiliki nilai aktual atau yang dipersepsikan dalam keputusan saat ini atau di masa depan. Informasi adalah data yang diolah dan bermanfaat bagi penerimanya, jika data yang diolah tidak bermanfaat bagi penerimanya, maka tidak dapat dikatakan sebagai informasi.

Informasi adalah semua informasi yang berguna bagi para pengambil keputusan/manajer untuk mencapai tujuan organisasi yang telah ditetapkan sebelumnya. (Gaol, 2008) Oleh karena itu, dapat disimpulkan bahwa informasi adalah data yang telah diperoleh dan diolah untuk mencapai tujuan tertentu yang telah ditetapkan sebelumnya.

Informasi tersebut dapat dibagi menjadi 3 bagian, yaitu: (Sutabri, 2012)

1. Informasi strategis. Informasi ini digunakan untuk membuat keputusan jangka panjang, dan termasuk informasi eksternal
2. Informasi taktis. Informasi ini diperlukan untuk membuat keputusan jangka menengah.
3. Informasi teknis. Informasi ini digunakan dalam operasional sehari-hari.

Pengolahan data menjadi informasi dilakukan oleh pengolah informasi. Pemrosesan informasi dapat mencakup komputer, elemen non komputer, atau kombinasi keduanya. Data yang diolah sebagai informasi memiliki nilai tambah dan fungsi yang dapat menambah pengetahuan atau mengurangi ketidakpastian pengguna. Informasi dapat berasal dari observasi, dialog

dengan orang lain, rapat komite, majalah, surat kabar atau laporan pemerintah dan sistem informasi itu sendiri.

2.1.3 Sistem Informasi

Menurut pengertian sistem dan informasi, dapat dikatakan bahwa sistem informasi merupakan suatu komponen data yang terintegrasi untuk mencapai tujuan. Sistem informasi memiliki tahapan masukan, pengolahan data dan bentuk keluaran. Sistem informasi juga dapat dikatakan banyak komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), hal-hal yang telah diolah (data menjadi informasi), dan dirancang untuk mencapai suatu tujuan. (Kadir, 2014) Salah satunya dapat dicontohkan dalam bentuk laporan untuk banyaknya pendaftar sertifikasi yang telah mengikuti sertifikasi, dan laporan untuk banyaknya pendaftar yang telah lolos sertifikasi.

Menurut Tata Sutabri dalam Analisis Sistem Informasi, Sistem Informasi adalah sistem dalam organisasi yang menjumlahkan kebutuhan pemrosesan transaksi harian untuk mendukung fungsi manajemen dan operasi organisasi, sehingga dapat memberikan laporan tertentu yang dibutuhkan oleh pihak eksternal.

Sistem informasi terdiri dari beberapa blok bangunan, seperti blok *input*, model, *output*, teknologi, *database*, dan blok *control*. Modul-modul ini saling berkomunikasi untuk mencapai tujuan. Oleh karena itu, sistem informasi memegang peranan penting dalam kinerja seluruh organisasi, karena sistem informasi memberikan banyak keuntungan kepada pengguna, mulai dari proses transaksi yang sederhana hingga tugas-tugas yang sulit di tingkat operasional, seperti membuat tugas penting dan kompetitif di tingkat strategis organisasi yaitu, pengambilan keputusan.

2.2 System Development Life Cycle (SDLC)

Untuk mengembangkan sebuah perangkat lunak diperlukan metode yang dapat mengatur alur atau prosedur agar proses pengembangan dapat dilakukan

dengan teratur dan sesuai dengan tujuan yang diinginkan. Prosedur ini dikenal dengan nama *System Development Life Cycle* (SDLC).

System Development Life Cycle (SDLC) adalah proses yang menggambarkan metode dan strategi, seperti bagaimana mengembangkan, merancang dan memelihara proyek perangkat lunak, dan memastikan bahwa semua tujuan, sasaran, fungsi dan persyaratan pengguna terpenuhi. (Arora R & Arora N, 2016)

Ada beberapa fase yang terdapat di dalam SDLC, yaitu:

- a. Perencanaan sistem (*System planning*)
- b. Analisis sistem (*System Analysis*)
- c. Perancangan sistem (*Systems Design*)
- d. Implementasi sistem (*Systems Implementation*)
- e. Pemeliharaan sistem (*Systems Maintenance*)

Konsep *System Development Life Cycle* adalah dasar dari model pengembangan perangkat lunak lainnya. Model pengembangan perangkat lunak ini meliputi *waterfall*, *prototype*, *iterative*, *spiral*, *rapid application development* (RAD) dan lainnya. Dalam penelitian ini penulis menggunakan model spiral.

2.2.1 Model Spiral

Model spiral merupakan salah satu model dari metode *System Development Life Cycle* dengan proses pengembangan perangkat lunak yang menggabungkan elemen desain dan pembuatan *prototype* untuk menggabungkan keunggulan konsep *topdown* dan *bottom up*. Model spiral berfokus pada penilaian risiko dan dapat meminimalkan risiko proyek yang akan dilakukan dengan memecahkan suatu proyek menjadi segmen yang lebih kecil.

Langkah-langkah berikut merupakan fase model spiral yang harus dilakukan, yaitu: (Alshamrani & Bahattab, 2015)

1. Perencanaan

Fase ini mencakup pemahaman tentang persyaratan sistem dengan melakukan komunikasi terus-menerus dengan pelanggan.

2. Analisis Risiko

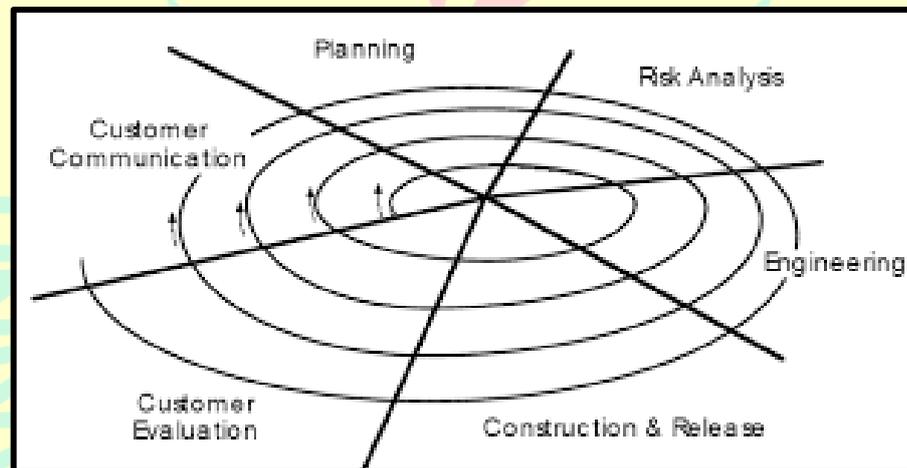
Dalam fase ini, suatu proses dilakukan untuk mengidentifikasi risiko dan solusi alternatif. Hasil akhir dari fase ini adalah sebuah prototipe.

3. Pengembangan/Rekayasa

Pada fase ini perangkat lunak akan diproduksi bersama dengan dilakukannya pengujian.

4. Tahap Evaluasi

Tahapan ini memungkinkan pelanggan untuk mengevaluasi output proyek sebelum proyek berlanjut.



Gambar 2.1: Metode Spiral

Meskipun jangka waktu pengembangan yang dibutuhkan model ini tidak cocok untuk jangka waktu pendek, tetapi model ini memiliki efisiensi yang sangat besar untuk suatu proyek dan memiliki manajemen risiko yang lebih baik. Dalam model spiral perangkat lunak yang digunakan masih berbentuk *prototype* sehingga dapat dikomunikasikan langsung dengan pelanggan atau pengguna sistem. Apabila belum dapat mencapai tujuan yang diinginkan oleh pengguna maka proses dilanjutkan dengan perencanaan untuk pengembangan selanjutnya agar dapat mencapai tujuan secara optimal.

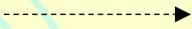
2.3 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa grafis yang secara formal didefinisikan oleh grup manajemen objek untuk memvisualisasikan, mendefinisikan, membangun, dan merekam item yang dibuat atau dikumpulkan selama pengembangan sistem perangkat lunak. Saat ini UML telah menjadi standar bahasa visual untuk memodelkan suatu sistem yang berorientasi objek. Dalam penelitian ini, penulis menggunakan tiga jenis diagram yaitu diagram *use case*, diagram *class*, dan diagram *activity*.

2.3.1 Diagram Use Case

Diagram *use case* menunjukkan serangkaian *user* dan *actor* serta hubungannya. Diagram ini digunakan untuk mengatur dan memodelkan aspek dinamis yang diperlukan dalam penggunaan suatu sistem. Diagram *use case* dilengkapi dengan spesifikasi kasus penggunaan lengkap dari suatu sistem, urutan kejadian, kondisi tertentu dan kondisi akhir dari suatu sistem. Sebuah *use case* menggambarkan interaksi antara pengguna atau *actor* dengan sistem. Contohnya *use case login*, yang mengharuskan *user* atau *actor* melakukan interaksi dengan sistem agar dapat *login* ke dalam sistem.

Tabel 2.1: Simbol yang ada pada *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menjelaskan pengguna yang akan berinteraksi dengan <i>use case</i> di sistem.
	<i>Include</i>	Menentukan bahwa <i>use case</i> yang ditunjuk adalah bagian dari <i>use case</i> lain.
	<i>Extend</i>	Menentukan bahwa <i>use case</i> yang ditunjuk adalah tambahan/perluasan dari <i>use case</i> lainnya.
	<i>Association</i>	Menghubungkan suatu objek dengan objek lainnya.

Simbol	Nama	Keterangan
	<i>Use case</i>	Deskripsi dari aksi yang dapat dilakukan oleh <i>actor</i> .

2.3.2 Diagram Class

Diagram *Class* adalah diagram yang paling umum ditemukan dalam sistem berorientasi objek dan digunakan untuk menggambarkan tampilan statis sistem. Diagram ini termasuk dalam spesifikasi UML sejak versi pertama dan merupakan salah satu diagram utama. Analisis pembentukan dari diagram *class* merupakan aktivitas inti yang akan memengaruhi arsitektur sistem hingga tahap pengkodean.

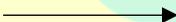
Tabel 2.2: Simbol pada *Diagram Class*

Simbol	Nama	Keterangan
	Generalisasi	Simbol ini berfungsi menghubungkan <i>class</i> yang memiliki hubungan umum ke khusus
	<i>Aggregation</i>	Menunjukkan bahwa suatu <i>class</i> merupakan semua bagian dari <i>class</i> lain
	<i>Dependency</i>	Relasi ini digunakan atas dasar ketergantungan antar <i>class</i>
	<i>Association</i>	Digunakan sebagai penghubung antar <i>class</i> dengan makna umum
	<i>Composition</i>	Menunjukkan bahwa <i>class</i> tidak dapat berdiri sendiri dan merupakan bagian dari <i>class</i> lain

2.3.3 Diagram Activity

Diagram *activity* merupakan diagram yang menggambarkan alur kerja atau aktivitas suatu sistem atau proses bisnis. Pada UML akan dibuat diagram *activity* untuk menjelaskan aktivitas komputer dan proses aktivitas dalam organisasi, proses dimulai dari aktivitas peristiwa apa yang mungkin terjadi, dan bagaimana berakhirnya. Diagram *activity* juga menggambarkan alur kontrol secara garis besar yang di dalamnya terdapat tindakan berupa pilihan atau pengulangan.

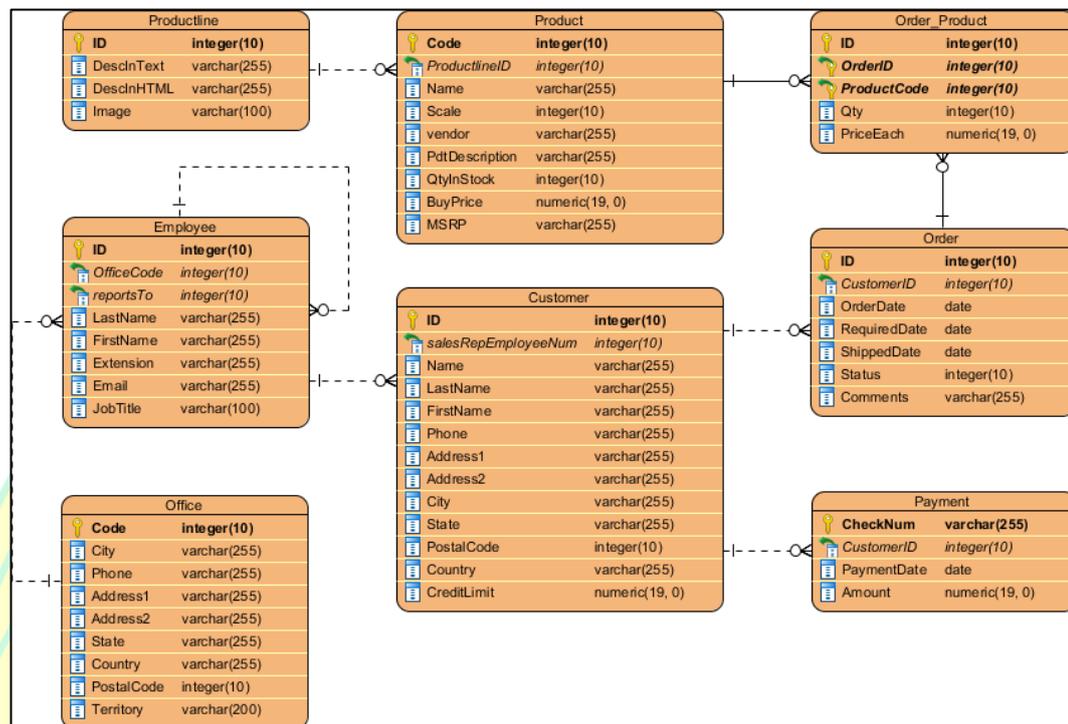
Tabel 2.3: Simbol-simbol yang ada pada *Diagram Activity*

Simbol	Nama	Keterangan
	<i>Activity</i>	Menggambarkan aktivitas sistem yang saling berinteraksi.
	<i>Decision</i>	Menunjukkan keputusan berdasarkan kondisi tertentu.
	<i>Line Connector</i>	Sebagai penghubung antara satu simbol dengan simbol lainnya
	<i>Initial Node</i>	Sebagai tanda awal dimulainya aktivitas pada sistem
	<i>Activity Final Node</i>	Sebagai tanda akhir berakhirnya aktivitas pada sistem

2.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram untuk menggambar desain konseptual, model konseptual *database* relasional. ERD juga merupakan gambaran hubungan antara satu benda dengan benda lainnya di dunia nyata, biasa disebut hubungan antar entitas. (Robi Yanto, 2016)

ERD terdiri dari 3(tiga) bagian utama, yaitu Entitas, Atribut, dan Relasi. Di dalam ERD terdapat beberapa relasi, disebut sebagai derajat kardinalitas yang akan digunakan yaitu relasi *one-to-one*, *one-to-many*, dan *many-to-many*.



Gambar 2.2: Contoh *Entity Relationship Diagram*

2.5 Arsitektur Model View Controller (MVC)

MVC adalah konsep yang merangkum data dan pemrosesan (*model*), dan mengisolasinya dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada antarmuka pengguna. (Simanjuntak & Kasnady, 2016) Pengertian teknis arsitektur MVC memiliki 3(tiga) lapisan, yaitu:

2.5.1 Model

Model tersebut merepresentasikan struktur data. *Model* ini digunakan untuk mengelola informasi dan memberitahu pengamat ketika informasi berubah. Hanya *model* yang berisi data dan fungsi yang terkait dengan pemrosesan data.

2.5.2 View

View atau tampilan bertanggung jawab untuk memetakan grafik ke perangkat. Tampilan ini bertanggung jawab untuk menampilkan konten model ke permukaan layer. Saat model berubah, tampilan akan secara otomatis menggambar ulang bagian layar yang terpengaruh untuk mencerminkan

perubahan tersebut. Dalam sebuah model, mungkin terdapat beberapa tampilan yang akan menampilkan permukaan tampilan yang berbeda.

2.5.3 Controller

Ini bertanggung jawab untuk menerima masukan dari pengguna dan menginstruksikan *model* dan tampilan untuk mengambil tindakan berdasarkan masukan tersebut. *Controller* bertanggung jawab untuk memetakan tindakan pengguna akhir ke tanggapan aplikasi. Misalnya, ketika pengguna menekan tombol atau memilih *item menu*, maka *controller* bertanggung jawab untuk menentukan bagaimana aplikasi harus merespon.

2.6 Framework Laravel

Laravel adalah *framework* PHP yang dirilis di bawah lisensi MIT, yang dibangun dengan konsep MVC. Laravel bertujuan untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan biaya pemeliharaan, serta meningkatkan pengalaman menggunakan aplikasi yang menyediakan sintaks ekspresif, jelas dan menghemat waktu.

Adapun kelebihan yang dimiliki oleh Laravel antara lain: (Yu, 2015)

1. Mengurangi beban pada *server* dan tekanan pada jaringan
2. Ketika bisnis perlu diperluas, cukup masukan *function* yang sesuai dengan *modul* perangkat lunak asli, *compiler* dapat diterapkan tanpa mengkompilasi ulang keseluruhan sistem
3. Desain dapat diselesaikan dengan lebih baik karena afinitas untuk pengguna dan pemogram diperkuat
4. *Modul* dapat digunakan kembali, hanya perlu disalin ketujuan dan langsung dapat digunakan setelah di-*compile* kembali
5. Mudah diperbarui
6. Implementasi pemisahan dari antarmuka

Singkatnya metode desain *web* berdasarkan kerangka Laravel dapat dilakukan untuk menyelesaikan penyeimbangan muatan, keamanan, dan masalah lainnya. Laravel dapat dikatakan sebagai metode desain yang *scalable* dan memiliki skalabilitas yang kuat, dan bagus untuk meningkatkan efisiensi pengembangan

proyek. Laravel akan memakan memori yang cukup besar, tetapi waktu eksekusi Laravel lebih cepat daripada *framework* lainnya.

2.7 Basis Data

Basis data terdiri dari dua kata, yaitu basis dan data. Basis didefinisikan sebagai gudang, pangkalan, dan tempat pertemuan. Pada saat yang sama, data adalah fakta yang merepresentasikan objek seperti manusia, benda, hewan, dan peristiwa.

Basis data adalah proses pengolahan data secara manual menjadi proses berbasis pendekatan basis data. Di mana proses pengolahan data dilakukan dengan teknik penyimpanan data pada satu direktori yaitu basis data. (Robi Yanto, 2016)

Data yang digunakan dalam sebuah basis data, harus mempunyai ciri sebagai berikut: (Lubis, 2016)

1. Disimpan secara terintegrasi dengan menghilangkan bagian yang tidak perlu (*redundant*)
2. Data digunakan dalam aplikasi yang berbeda pada waktu yang bersamaan

Di dalam basis data terdapat beberapa operasi dasar yang akan digunakan untuk mengolah data, yaitu: (Pamungkas, 2017)

1. *Create Database*, perintah untuk membuat basis data dengan nama yang akan diberikan
2. *Drop database*, perintah untuk menghapus basis data dengan nama yang diberikan
3. *Drop table*, perintah untuk menghapus suatu tabel dalam basis data
4. *Insert*, perintah untuk memasukkan data ke dalam tabel
5. *Update*, perintah untuk memperbarui data pada tabel
6. *Delete*, perintah untuk menghapus data pada tabel

Disatu sisi terdapat basis data memiliki keuntungan juga kerugian bagi para penggunanya. Keuntungan yang diberikan basis data bagi para penggunanya adalah sebagai berikut:

- a. Pengendalian terhadap data terpusat
- b. Redudansi data dapat dikurangi
- c. Terciptanya data yang konsisten
- d. Data dapat dipakai bersama
- e. Dapat dilakukan pembatasan keamanan data
- f. Integritas data dapat dipelihara
- g. Independensi data

Basis data juga memiliki beberapa kerugian yaitu seperti biaya yang dibutuhkan sangat mahal dan bersifat kompleks sehingga membutuhkan tingkat keahlian yang tinggi. Saat ini terdapat sistem manajemen basis data, biasanya disebut *Database Management System (DBMS)*, yaitu perangkat lunak yang dirancang untuk menyimpan dan mengelola basis data. Sistem tersebut juga menerapkan mekanisme keamanan data, membagikan data, dan meningkatkan akurasi data.

Oleh karena itu, dapat disimpulkan bahwa sistem basis data harus ada dalam suatu sistem informasi, karena suatu sistem informasi tidak dapat dilaksanakan tanpa melibatkan basis data.

2.8 Testing

Testing merupakan proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi persyaratan tertentu atau untuk mengidentifikasi perbedaan antara hasil yang diharapkan dan yang sebenarnya. (Chehal & Singh, 2012)

Sedangkan menurut Lewis, pengujian perangkat lunak adalah menjalankan serangkaian aktivitas yang dijalankan secara dinamis pada program perangkat lunak setelah kode sumber perangkat lunak dikembangkan. Sebelum pelanggan atau pengguna akhir menggunakan perangkat lunak, pengujian perangkat lunak akan

dilakukan untuk menemukan dan memperbaiki sebanyak mungkin kesalahan potensial.

Menurut definisi di atas, pengujian adalah kegiatan atau proses pemeriksaan dan evaluasi sistem, yang bertujuan untuk menemukan kesalahan pada sistem. Metode *testing* yang akan digunakan dalam penelitian ini adalah *black box testing*.

Black box testing adalah metode yang dipakai untuk menguji sebuah perangkat lunak tanpa harus memperhatikan detail dari perangkat lunak. Teknik pengujian *black box* memungkinkan untuk mendapatkan satu kondisi *input* yang sepenuhnya akan menjalankan semua persyaratan fungsional suatu program. (Pressman, 2010) Proses pengujian pada *black box* dilakukan dengan cara mencoba program dengan memasukkan data ke dalam formulir yang disediakan. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan. (Wahyudi, Utami, & Arief, 2016)

2.9 Proses Alur Sertifikasi Pustakawan Perpustakaan Nasional RI

Pustakawan yang ingin mengajukan permohonan sertifikasi harus mengisi formulir permohonan terlebih dahulu yang dapat diunduh pada *website* sertifikasi perpustakaan dengan pilihan klaster yang ingin disertifikasi. Terdapat 5 (lima) klaster yang tersedia, yakni pengadaan bahan pustaka, katalogisasi bahan pustaka, layanan sirkulasi dan referensi, literasi informasi dan promosi perpustakaan, serta klaster pelestarian bahan pustaka.

Setelah itu peserta juga harus mengisi formulir asesmen mandiri yang sama dengan klaster yang telah dipilih. Contohnya jika peserta memilih klaster untuk pengadaan bahan perpustakaan maka formulir asesmen mandiri yang harus diisi adalah formulir asesmen mandiri untuk pengadaan bahan perpustakaan. Setelah formulir permohonan dan formulir asesmen mandiri diisi, peserta harus mengirimkannya ke alamat email sertifikasipustakawan@gmail.com.

Setelah itu peserta diminta untuk melengkapi bukti-bukti pendukung kompetensi yang sesuai dengan klaster yang telah dipilih dan melampirkannya ke dalam formulir asesmen mandiri pada kolom bukti kompetensi. Kemudian, berkas

yang telah diajukan akan dinilai dengan kesesuaian bukti yang telah dicantumkan menggunakan kriteria *Valid, Authentic, Current, Sufficient* (VACS).

Peserta juga harus melengkapi dokumen administrasi pendaftaran seperti berikut: (Perpusnas, 2014)

- a. Salinan ijazah ilmu perpustakaan atau diploma ilmu non perpustakaan, dan sertifikat pelatihan pelamar pustakawan profesional
- b. Salinan bukti pengangkatan pustakawan (berupa surat keterangan pengangkatan pustakawan dari instansi)
- c. Salinan identitas diri (KTP/SIM/Paspor)
- d. 2 lembar pasfoto terbaru ukuran 3 x 4 (bagian belakang dituliskan nama jelas dan nama klaster yg dipilih)
- e. 2 lembar materai 6000
- f. Lampirkan surat rekomendasi/izin, keterangan bekerja/pekerjaan
- g. Lampirkan resume, yang berisi informasi tentang data pribadi, riwayat pendidikan/pelatihan, pekerjaan, pelatihan, seminar, atau makalah yang terkait. Format DRH dapat diatur sesuai kebutuhan pribadi

Kemudian kirimkan berkas administrasi ke alamat LSP Pustakawan Sekretariat Perpustakaan Nasional RI. Ged. B lantai 3 (Deputi II), Jl. Salemba Raya 28A Jakarta Pusat, 10430. Tahap selanjutnya, LSP akan melakukan verifikasi terhadap dokumen lamaran yang diajukan peserta. Peserta yang memenuhi syarat akan diatur untuk mengikuti penilaian sertifikasi di Perpustakaan Nasional Republik Indonesia atau tempat yang sudah ditentukan oleh LSP. Peserta yang mengikuti ujian sertifikasi akan diberitahu melalui telepon atau *e-mail*. Melalui telepon atau *e-mail*, peserta akan diberikan jadwal dan lokasi untuk melaksanakan ujian sertifikasi. Kemudian peserta akan melakukan ujian sesuai dengan jadwal dan lokasi yang telah ditentukan LSP Perpusnas dengan membawa bukti-bukti pendukung sesuai dengan klaster yang dipilihnya.

Pada saat ujian peserta akan dites oleh LSP perpusnas dengan metode demo, wawancara, dan tertulis, atau portofolio. Kemudian asesor akan melakukan

penilaian. Hasil ujian yang telah dinilai oleh asesor akan dibahas pada sidang pleno untuk menetapkan apakah peserta tersebut dinyatakan kompeten atau belum kompeten. Setelah sidang selesai, untuk peserta yang dinyatakan kompeten maka akan diberikan sertifikat yang akan dikirim langsung oleh perpunas ke alamat yang telah tercantum pada saat peserta melakukan pendaftaran. Untuk peserta yang dinyatakan belum kompeten dan tidak lulus dapat melakukan pengajuan sertifikasi ulang.



BAB III

IMPLEMENTASI PROGRAM

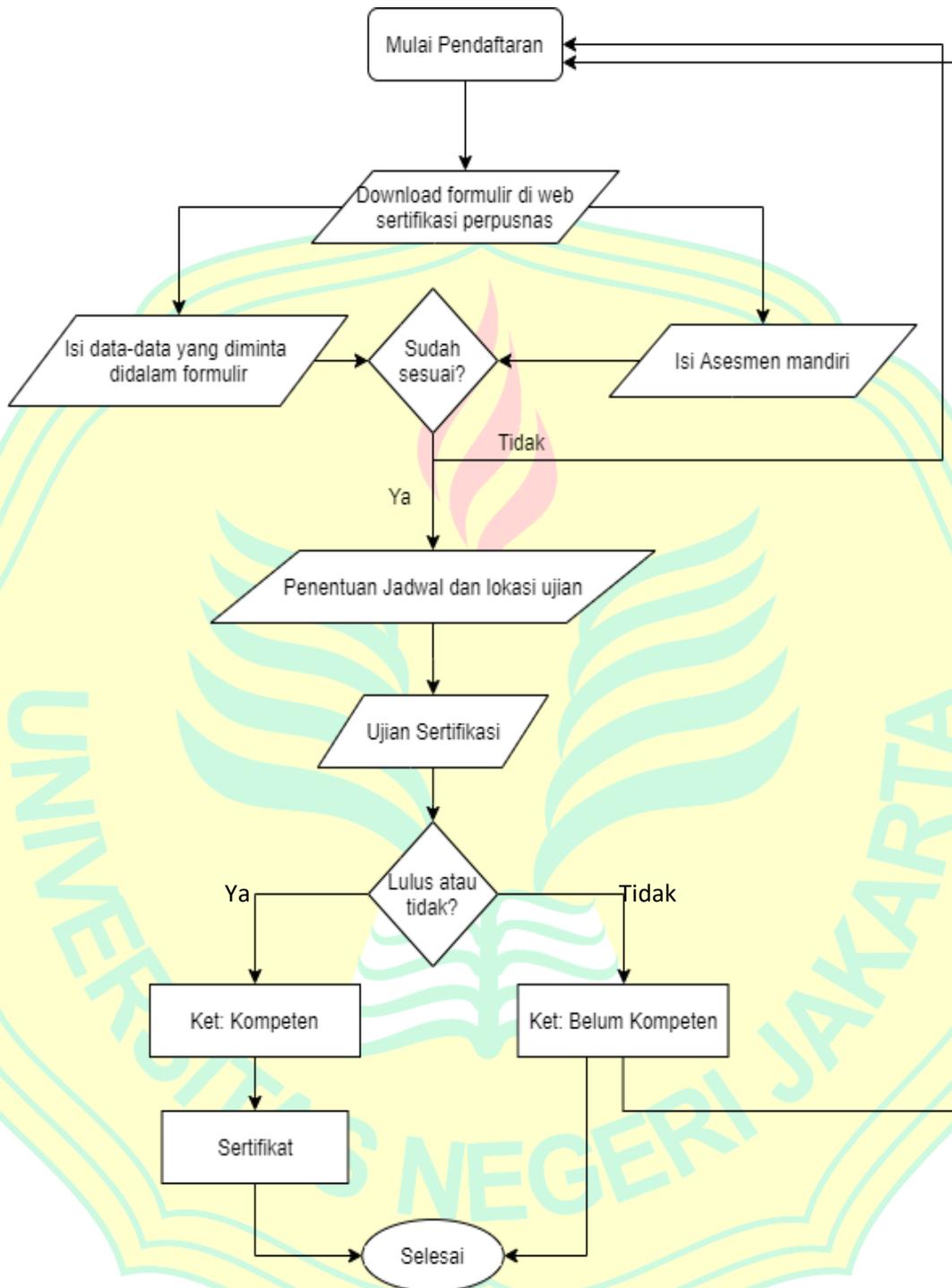
Penelitian ini, penulis mengembangkan perangkat lunak dengan menggunakan Model Spiral. Terdapat iterasi (*cycle*) atau prototipe aplikasi pada model ini. Rentang waktu untuk satu iterasi pada pengembangan sistem ini selama satu minggu. Pada tiap iterasi akan dilakukan pengembangan sebanyak 2 (dua) *use 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
	Dapat mengunduh sertifikat jika dinyatakan kompeten
Admin	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
	Dapat membuat berita artikel
Asesor	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

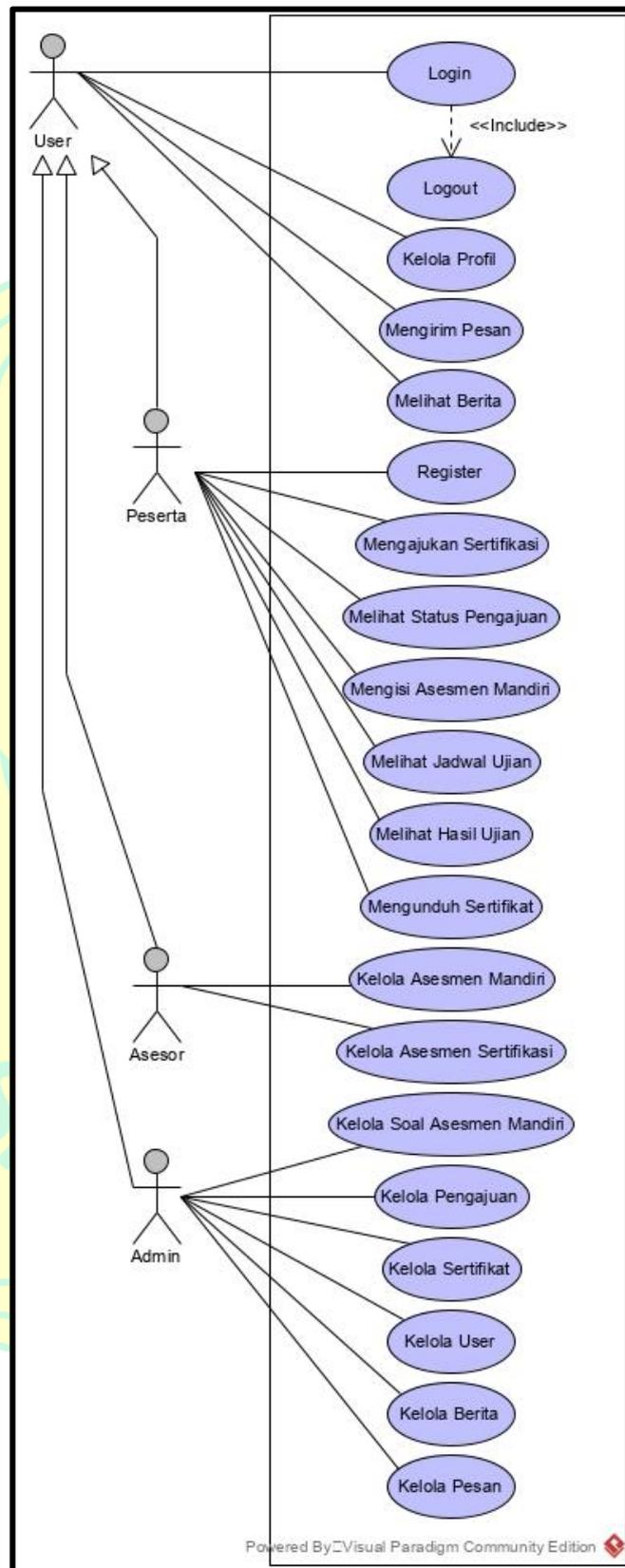
Di dalam *use case diagram* penulis menjelaskan interaksi antara pengguna dari sistem atau aktor, dengan sistem. Setiap aktor memiliki perannya masing-masing di dalam sistem yang sedang dikembangkan oleh penulis. Pada sistem informasi sertifikasi ini terdapat 3 aktor yang digeneralisasi dari *user*. 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 Perpusnas sebagai pengawas dan penilai ujian dengan syarat tertentu dan untuk jangka waktu tertentu. Aktor admin di dalam sistem adalah pegawai Lembaga Sertifikasi Pustakawan Perpusnas.

Peserta memiliki beberapa tugas, yaitu melakukan pengajuan sertifikasi dengan mengisi data, memasukkan bukti kompetensi, memasukkan dokumen pendukung atau dokumen yang telah disyaratkan oleh pihak perpusnas, 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 perpusnas.

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.



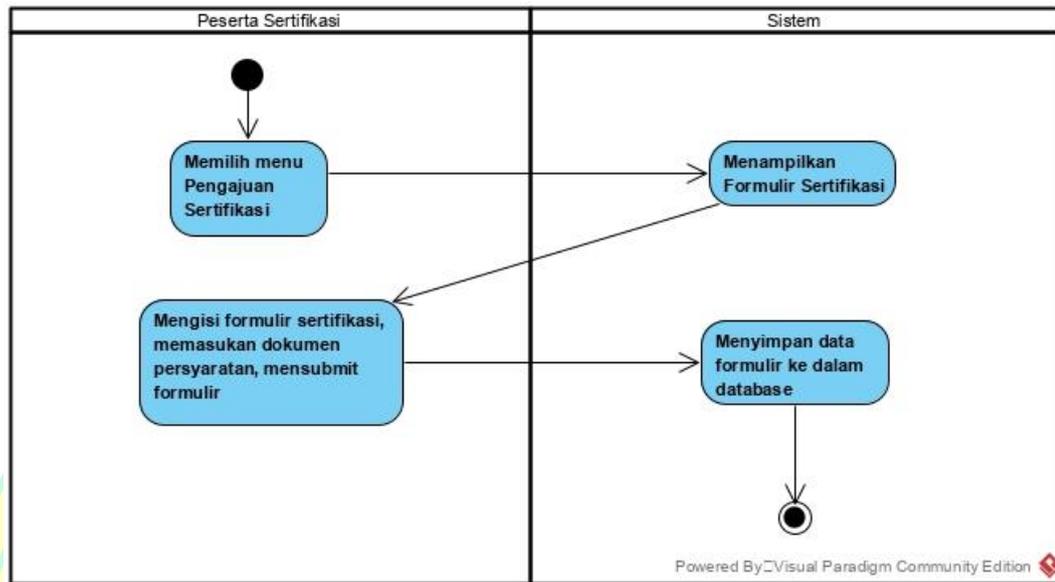
Gambar 3.2: Gambar *Use Case Diagram* Sertifikasi

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*.

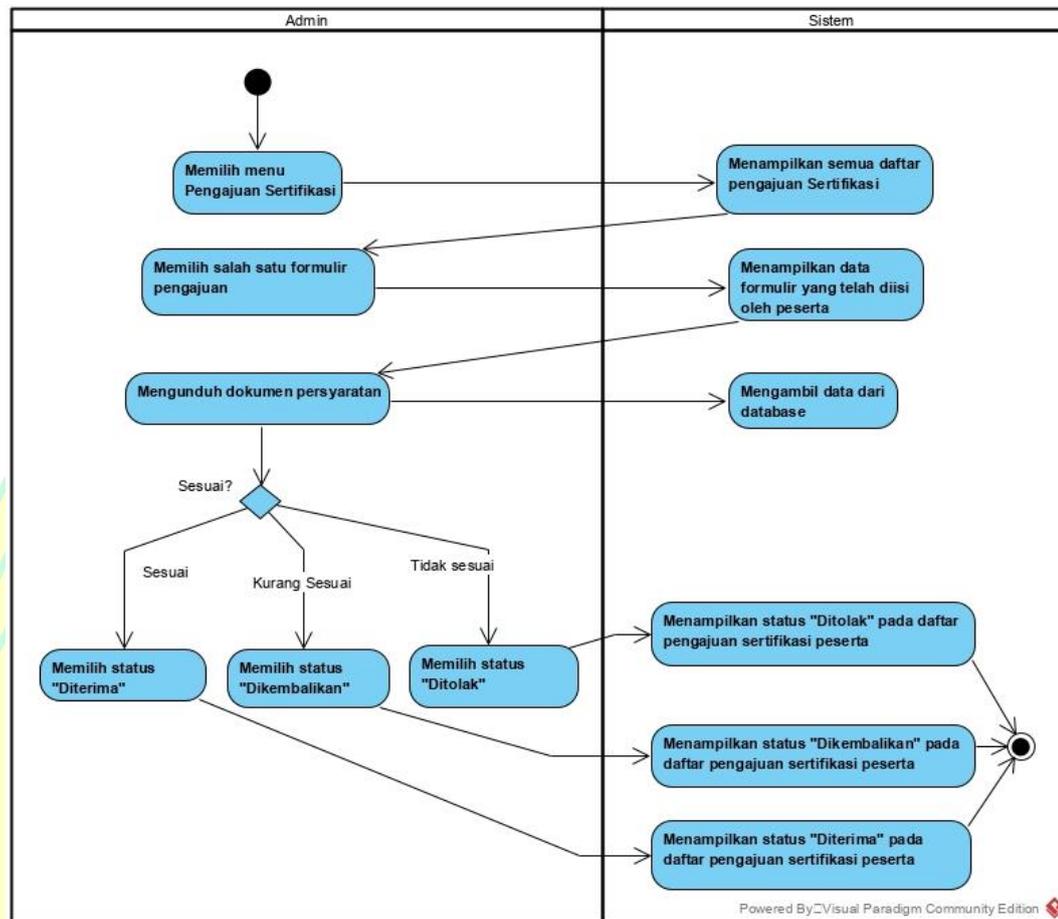
Saat peserta masuk ke dalam sistem, peserta harus melengkapi terlebih dahulu data diri pada sistem, seperti mengisi nomer telepon untuk dapat melakukan pengajuan sertifikasi. Pengajuan dilakukan dengan mengisi formulir data diri dan mengunggah dokumen yang diminta oleh sistem. Peserta juga dapat melihat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



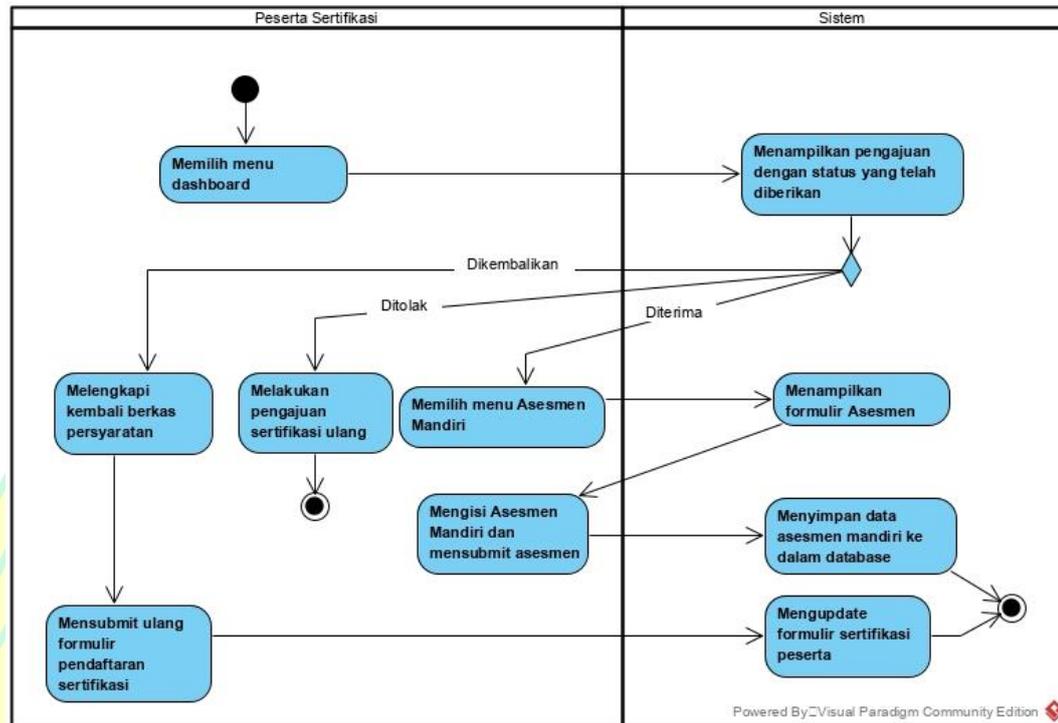
Gambar 3.3: Activity Diagram Pengajuan Sertifikasi

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.



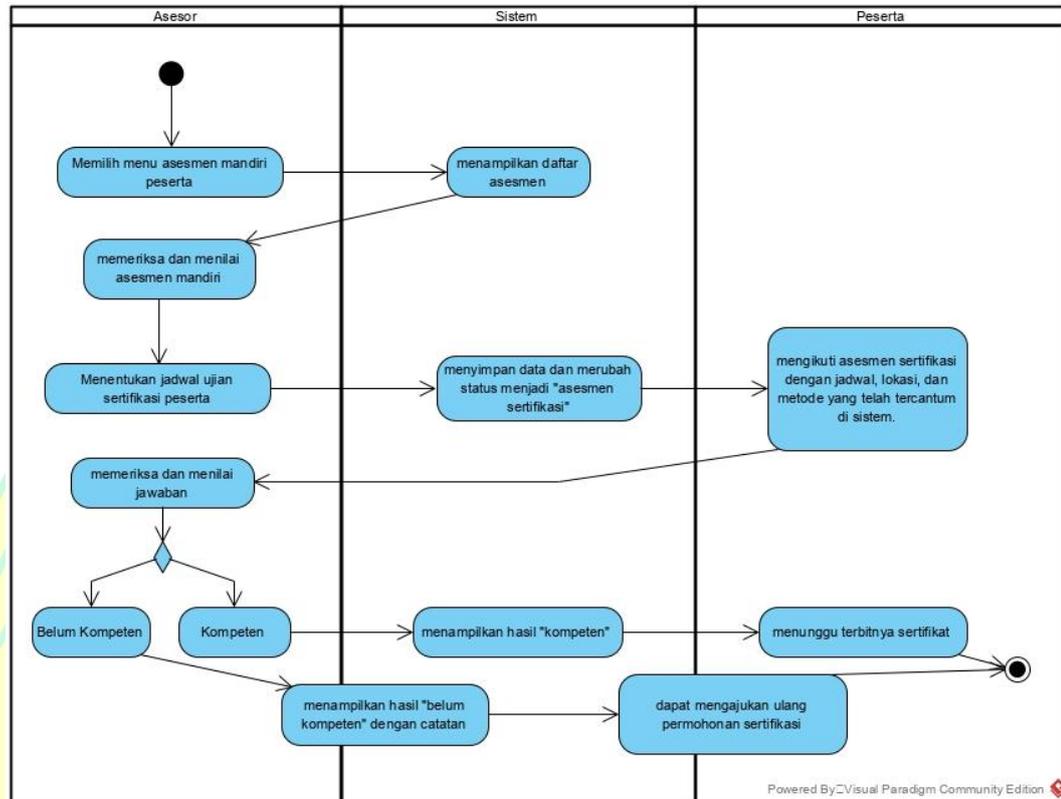
Gambar 3.4: Activity Diagram Verifikasi Data dan Dokumen

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.

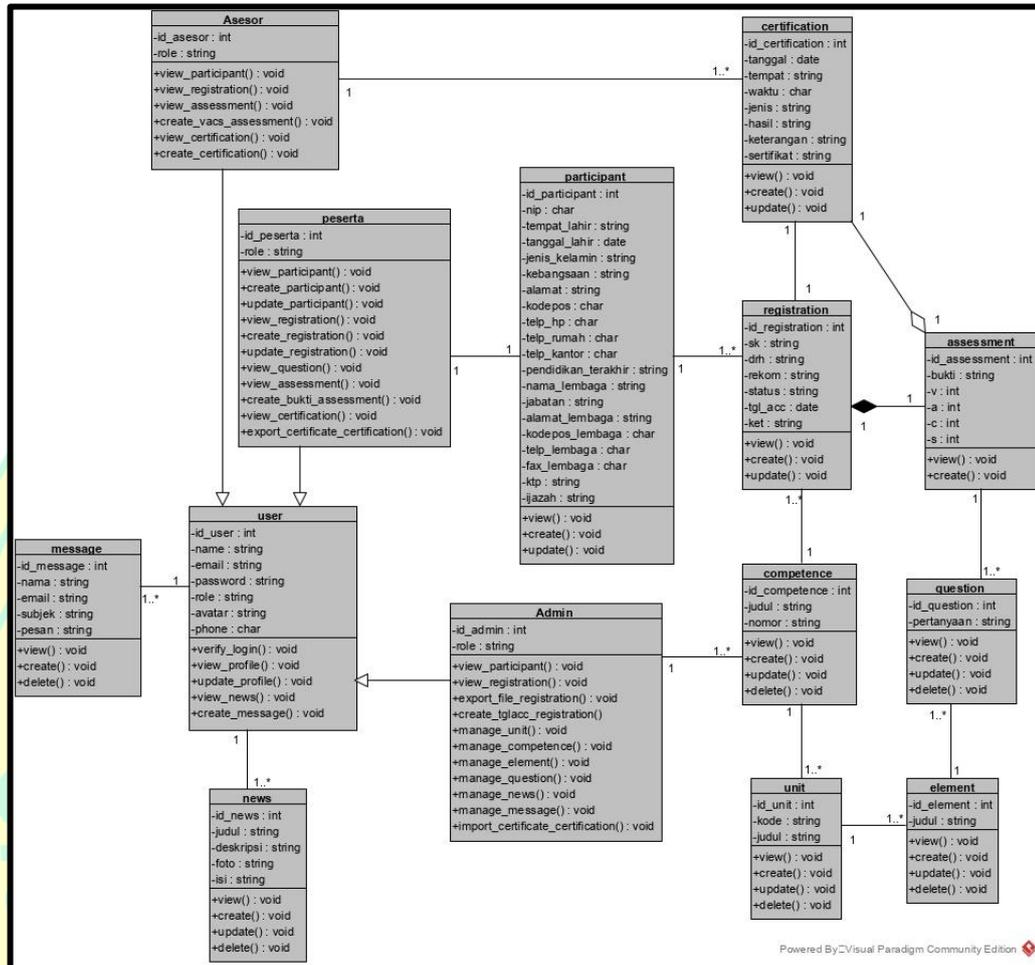
3.2.3 Class Diagram

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)

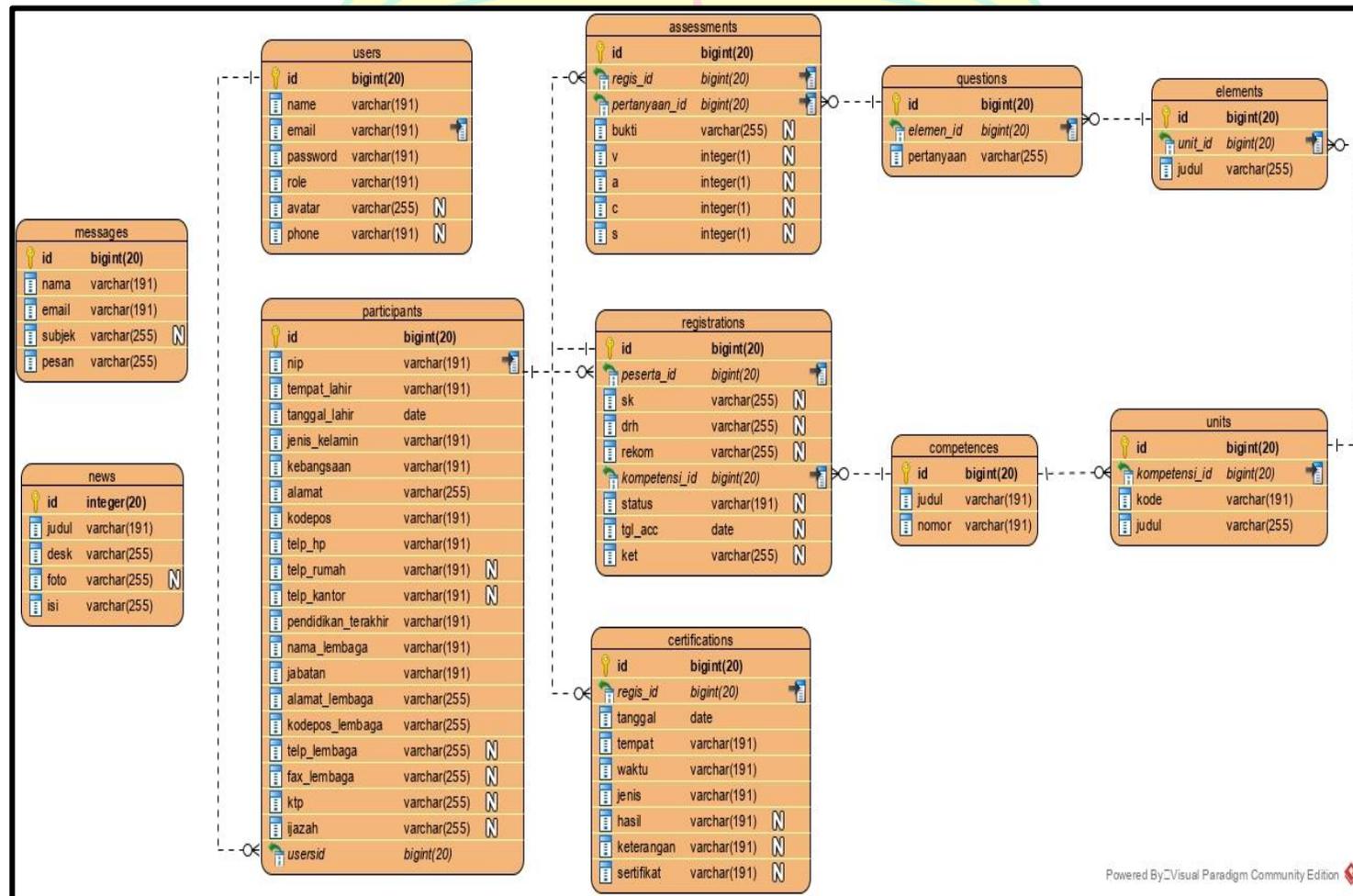
Desain Entity Relationship Diagram (ERD) merupakan diagram yang menggambarkan alur penyimpanan data pada sistem yang dikembangkan. ERD sistem informasi sertifikasi ini memiliki 11 entitas yaitu *users*, *news*, *messages*, *participants*, *registrations*, *certifications*, *assessments*, *questions*,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

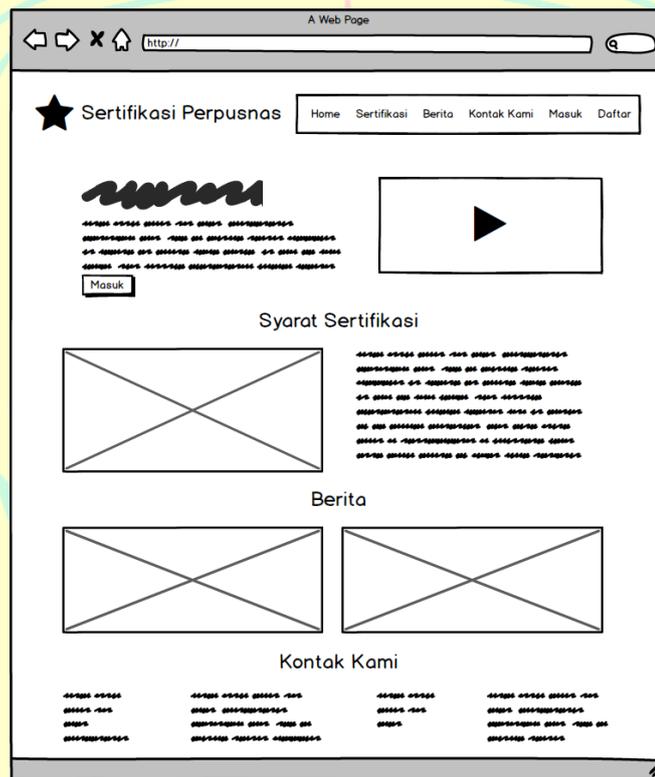
Berikut adalah desain ERD dari sistem informasi sertifikasi pustakawan:



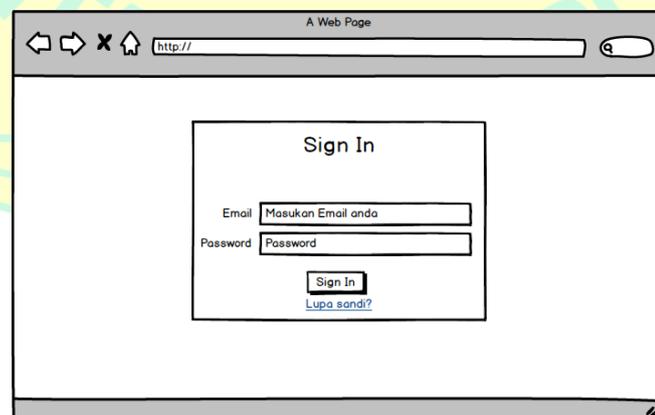
Gambar 3.8: Entity Relationship Diagram Sistem Informasi Sertifikasi

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.

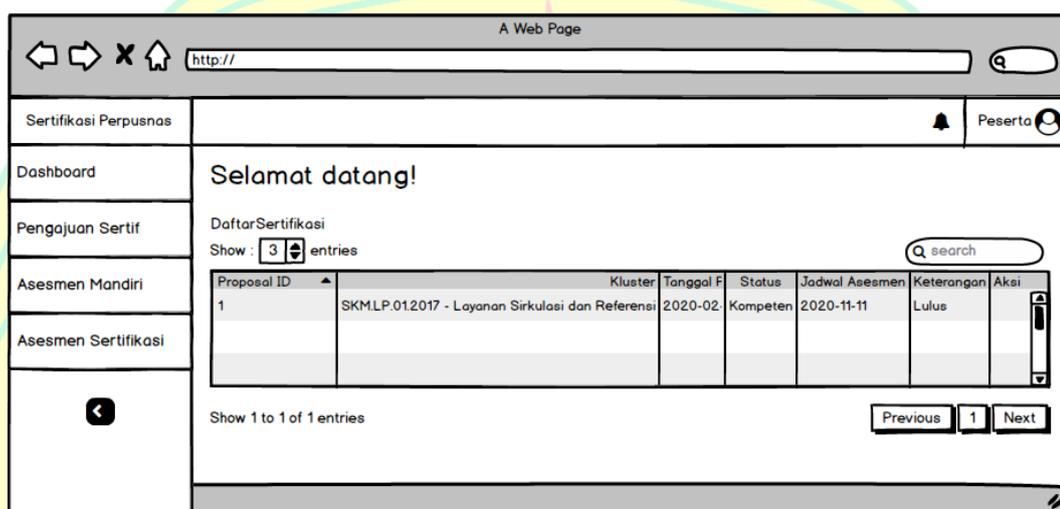


Gambar 3.9: Desain Tampilan Luar Sertifikasi



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 kluster 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.

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

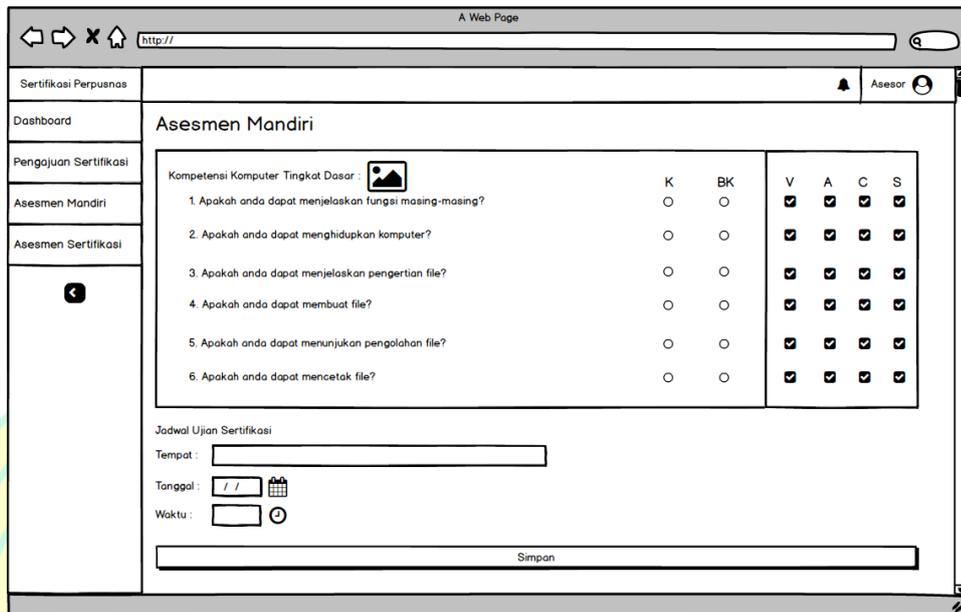
Setelah peserta mengajukan pengajuan. Admin akan memeriksa apakah berkas yang diajukan apakah sudah sesuai dan benar menurut persyaratan atau belum. Kemudian admin akan menentukan diterima, dikembalikan atau ditolaknya suatu pengajuan. Peserta baru bisa mengisi asesmen mandiri jika pemberkasannya telah 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 menunjukkan pengolahan file?	<input type="radio"/>	<input type="radio"/>	
6. Apakah anda dapat mencetak file?	<input type="radio"/>	<input type="radio"/>	

Submit

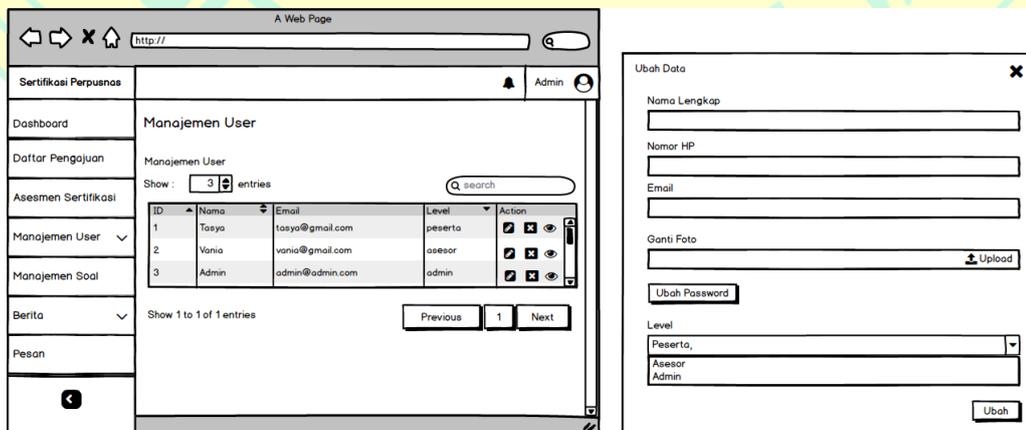
Gambar 3.13: Desain Tampilan Asesmen Mandiri Peserta

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

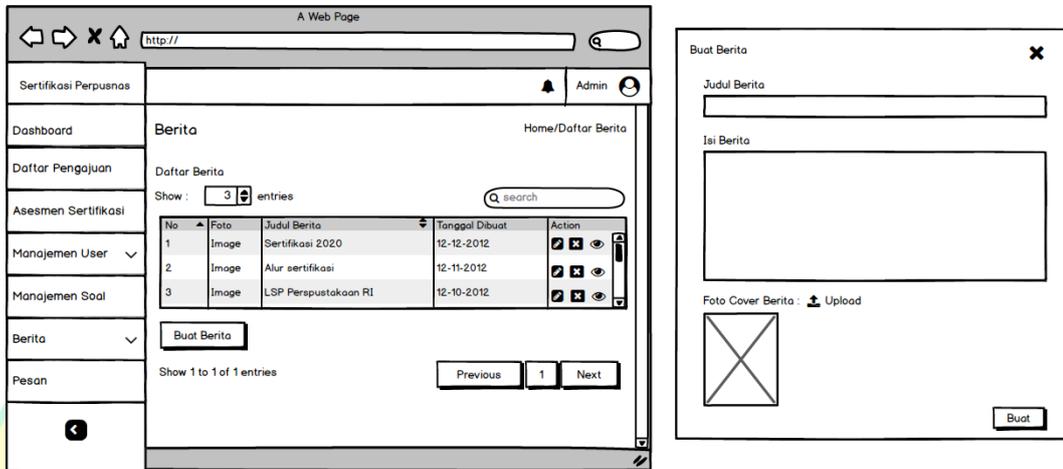


Gambar 3.14: Desain Tampilan Penilaian Asesmen Mandiri

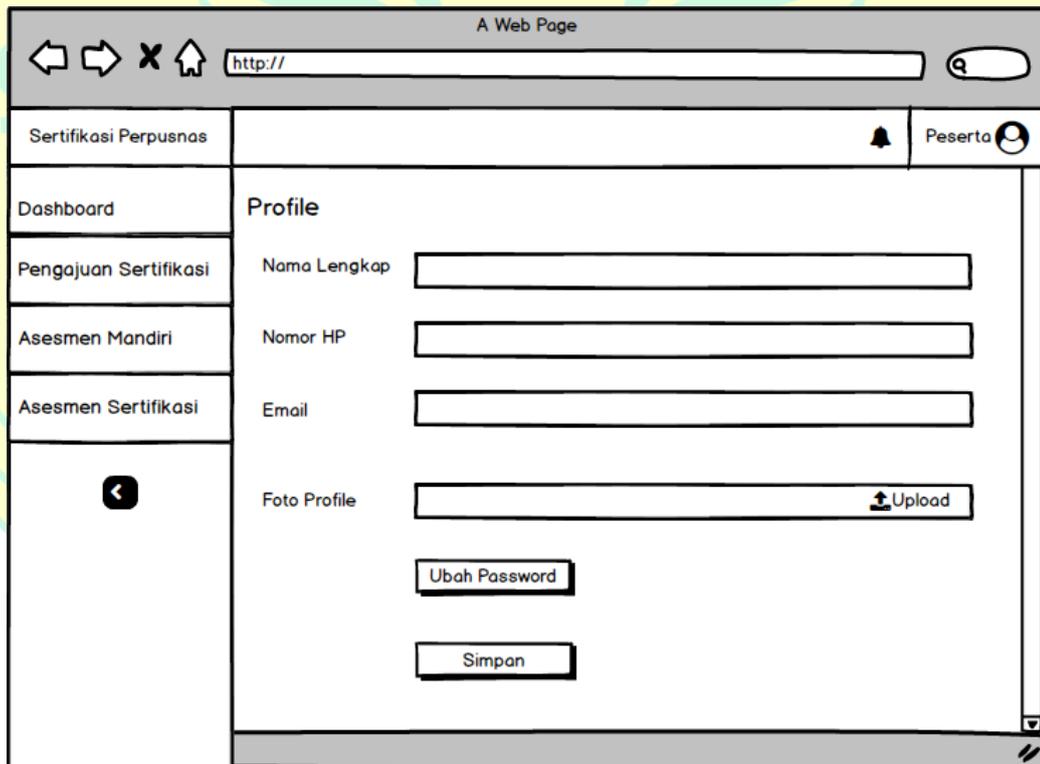
Pada gambar 3.14, selain melakukan penilaian, asesor akan menentukan jadwal ujian sertifikasi peserta. Setelah itu peserta akan mengikuti ujian secara *offline*. Setelah peserta mengikuti ujian sertifikasi pada jadwal yang telah ditentukan, asesor akan memasukkan nilai peserta berupa kompeten atau belum 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*.



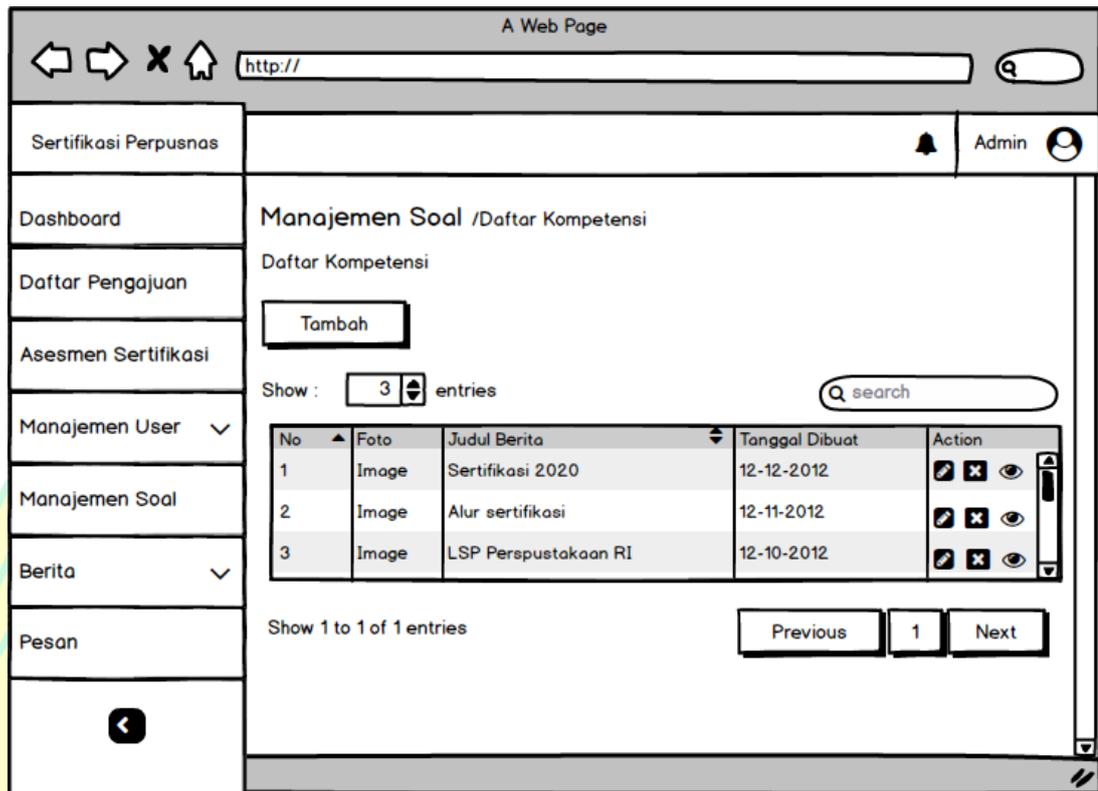
Gambar 3.15: Desain Tampilan Manajemen user



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 terancang 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

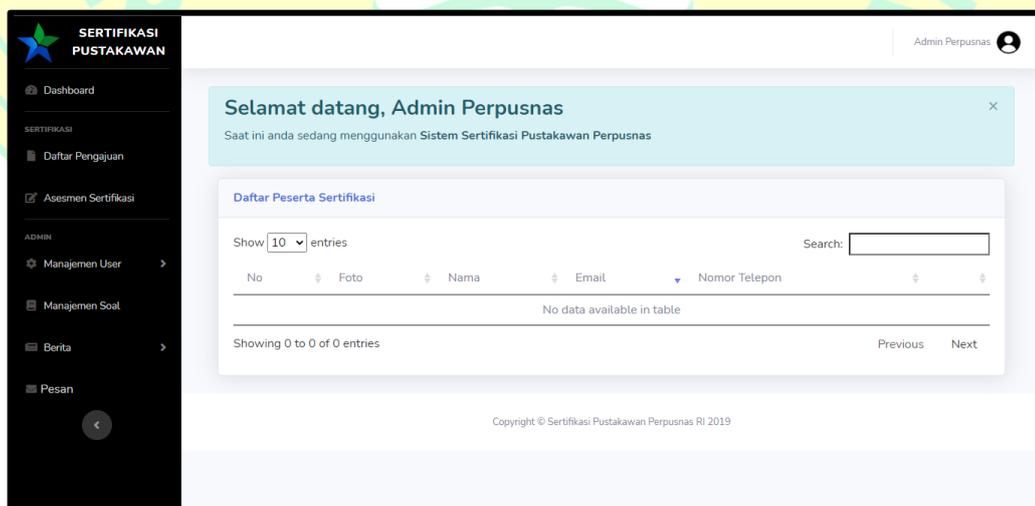
3.3.2 Implementasi Desain Tampilan

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.



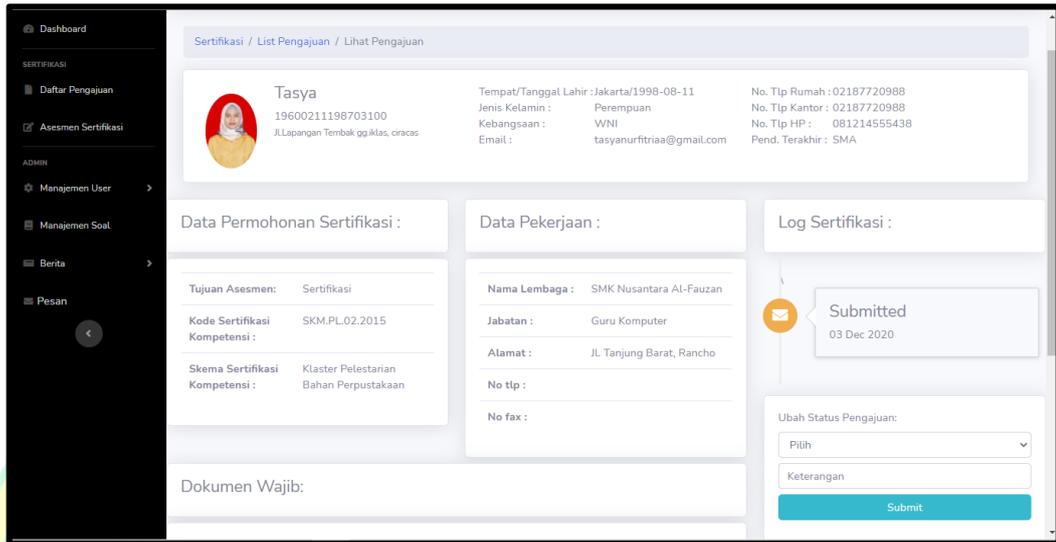
Gambar 3.21: Tampilan Halaman Untuk Beranda

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.

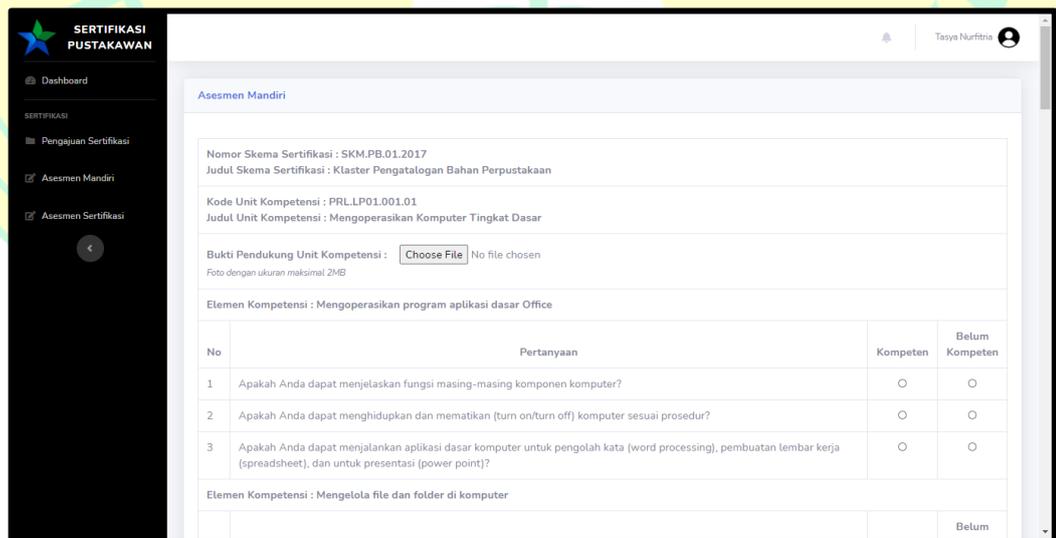
Data Pribadi		Data Pekerjaan Sekarang	
NIP *	<input type="text" value="NIP"/>	Nama Lembaga *	<input type="text" value="Masukkan nama lembaga/perusahaan"/>
Tempat lahir *	<input type="text" value="Tempat Lahir"/>	Jabatan *	<input type="text" value="Masukkan jabatan terakhir anda"/>
Tanggal Lahir *	<input 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"/>

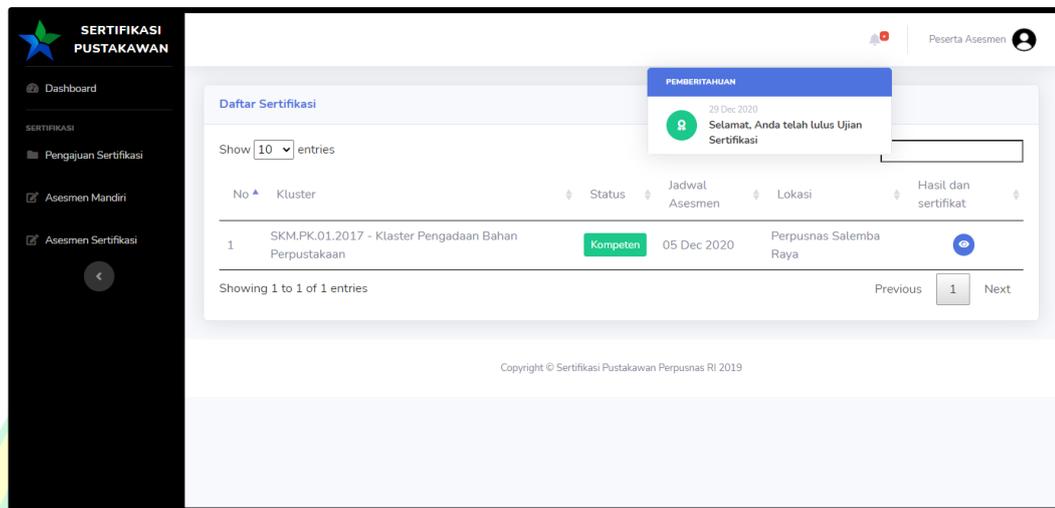
Jadwal Ujian Sertifikasi :

Tempat :

Tanggal: 

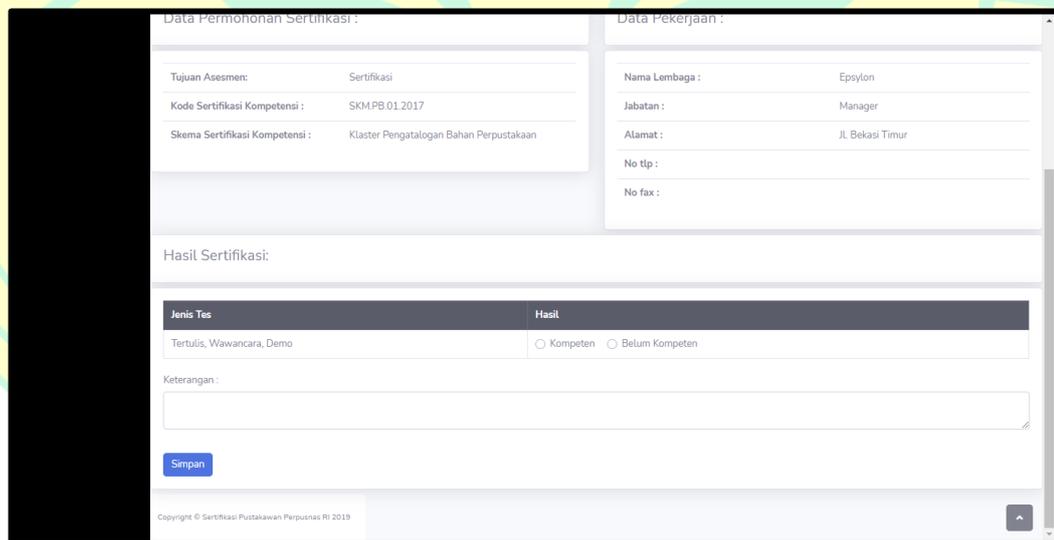
Waktu : 

Gambar 3.25: Tampilan Halaman Penilaian Asesmen Mandiri pada Asesor



Gambar 3.26: Tampilan Halaman Asesmen Sertifikasi pada Peserta

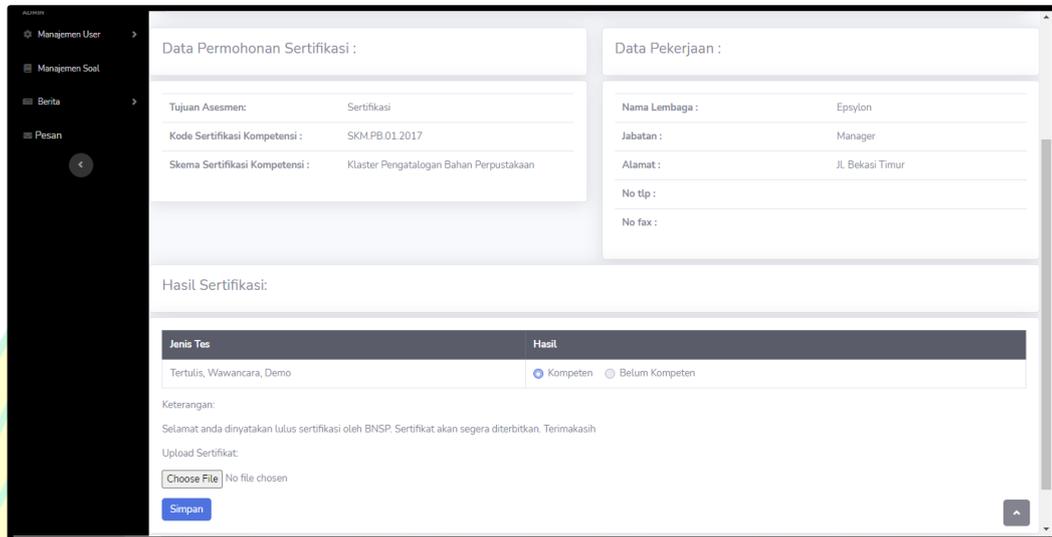
Pada gambar 3.26 peserta akan mendapatkan notifikasi apabila jadwal sertifikasi telah ditentukan oleh asesor dan pada halaman ini peserta dapat melihat hasil asesmen mandiri dengan menekan tombol biru di sebelah tabel.



Gambar 3.27: Tampilan Halaman Penilaian Asesmen Sertifikasi pada Asesor

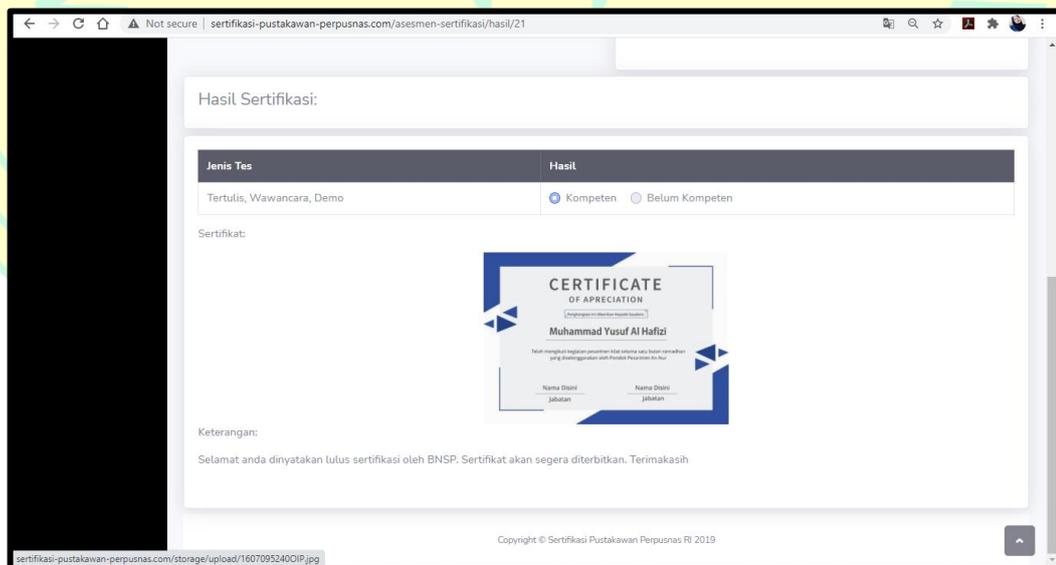
Gambar 3.27 adalah tampilan halaman pada asesor ketika asesor akan melakukan penilaian untuk asesmen sertifikasi yang telah dilakukan peserta secara *offline*. Asesor dapat memilih hasil sertifikasi berupa kompeten atau

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



Gambar 3.28: Tampilan Halaman Asesmen Sertifikasi Peserta pada Admin

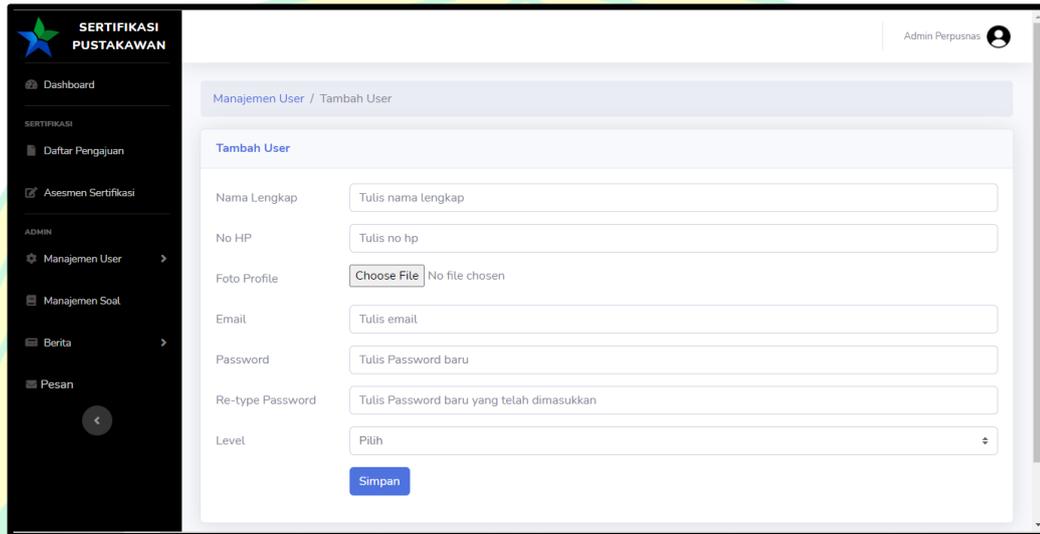
Gambar 3.28 merupakan tampilan pada halaman admin ketika akan mengunggah sertifikat untuk peserta yang telah dinyatakan kompeten oleh asesor. Sertifikat yang diunggah dapat berbentuk PDF ataupun JPEG/JPG.



Gambar 3.29: Tampilan Halaman Hasil Asesmen Sertifikasi pada Peserta

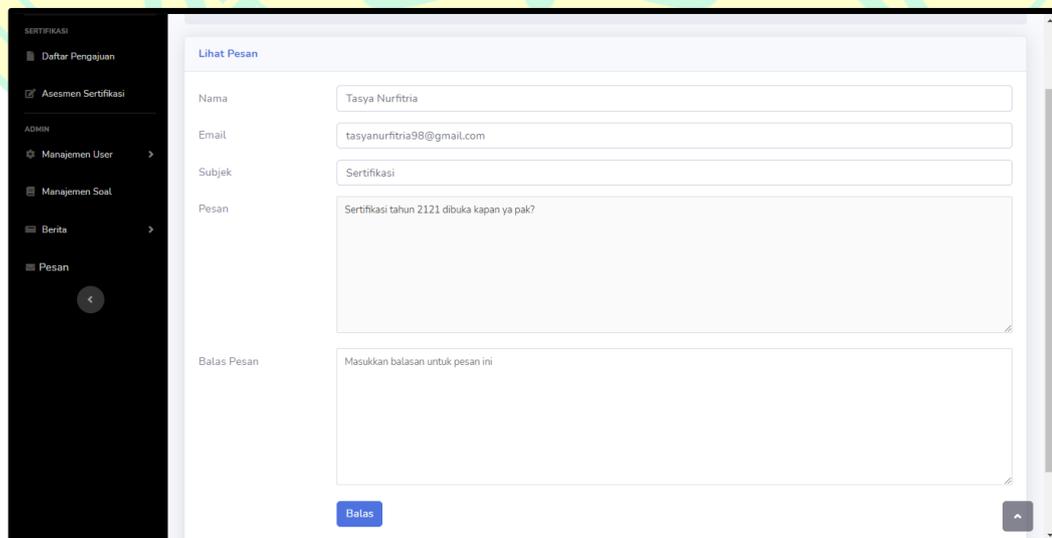
Gambar 3.31: Tampilan Halaman Berita pada Admin

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*.



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*.



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*.

A screenshot of a code editor window titled 'Registration.php'. The code is as follows:

```
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-3MI" 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_p');
52 Route::delete('manajemen-soal/kompetensi/unit/elemen/pertanyaan/{id}', 'AdminController@delete_pertanyaan')->name('d');
53 //manajemen-user
```

Gambar 3.37 : Struktur Pemrograman Routing

