

**APLIKASI
BP BATAM SEAPORT INFORMATION
MANAGEMENT SYSTEM (B-SIMS)
UNTUK PERIZINAN KEGIATAN KEPELABUHANAN
SECARA ONLINE KOTA BATAM**

PROPOSAL TUGAS AKHIR

Oleh:

MAILANY

4311931003

Disusun untuk pengajuan proposal Tugas Akhir Program Diploma IV



**PROGRAM STUDI TEKNIK MULTIMEDIA JARINGAN
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2021**

HALAMAN PENGESAHAN PROPOSAL

**APLIKASI
BP BATAM SEAPORT INFORMATION
MANAGEMENT SYSTEM (B-SIMS)
UNTUK PERIZINAN KEGIATAN KEPELABUHANAN
SECARA ONLINE KOTA BATAM**

Oleh:

MAILANY

4311931003

Proposal ini telah dikonsultasikan dengan dosen pembimbing
sebagai persyaratan untuk melaksanakan sidang proposal
di

**PROGRAM DIPLOMA IV
PROGRAM STUDI TEKNIK MULTIMEDIA JARINGAN
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM**

Batam, 07 Juli 2021

Disetujui oleh:

Pembimbing I,



Supardianto, S.S.T., M. Eng.

NIK. 113105

BAB I

PENDAHULUAN

1.1 Latar Belakang

Badan Pengusahaan (BP) Batam adalah Lembaga/instansi pemerintah pusat yang dibentuk berdasarkan Peraturan Pemerintah Republik Indonesia Nomor 46 Tahun 2007 dengan tugas dan wewenang melaksanakan pengelolaan, pengembangan dan pembangunan kawasan sesuai dengan fungsi-fungsi kawasan. Salah satu tugas dan fungsi dari BP Batam adalah pelaksanaan kegiatan pengelolaan sarana perhubungan laut. Badan Usaha Pelabuhan merupakan salah satu unit usaha pengelola Pelabuhan di Batam yang bersifat khusus karena dikelola langsung oleh Badan Pengusahaan (BP) Batam itu sendiri. Melihat dari sisi geografis Kota Batam yang strategis memberikan keunggulan dalam berinvestasi, terutama dalam sektor laut yang berdekatan dengan jalur laut internasional yaitu Selat Malaka dan Selat Singapura.

Untuk meningkatkan kegiatan berinvestasi itu sendiri perlu tata tertib kelola pelabuhan yang menegaskan bahwa Badan Usaha Pelabuhan memegang peranan penting untuk cepat atau tidaknya proses perizinan, dikarenakan pemenuhan persyaratan dan prosesnya bergantung pada kecepatan Badan Usaha Pelabuhan dan menyiapkan berbagai pemenuhan persyaratan yang dibutuhkan para investor. Tata tertib perizinan kepelabuhanan itu sendiri tidak lepas dari etika pelayanan publik dalam birokrasi seperti: kemudahan akses, akuntabel, efisien, efektif, dan transparansi yang diperlukan akan semakin mudah dan cepat dalam upaya menyederhanakan perizinan di dalam Badan Usaha Pelabuhan.

Berdasarkan hasil wawancara dengan pimpinan departemen operasional, bahwa belum ada sistem yang dapat mengakomodir kegiatan perizinan secara online di Badan Usaha Pelabuhan BP Batam untuk para pelaku usaha. Selama ini sistem yang ada hanya dapat digunakan di internal oleh pegawai Badan Usaha Pelabuhan BP Batam yaitu Sistem Seaport BP Batam, sehingga dalam proses perizinan kegiatan kepelabuhanan di lapangan masih terdapat interaksi tatap muka antara pelaku usaha dan pegawai. Sehingga diharapkan dengan adanya sistem perizinan secara online ini dapat memudahkan dalam kegiatan perizinan

kepelabuhanan di Badan Usaha Pelabuhan BP Batam secara online dan dengan adanya sistem tersebut mengurangi interaksi tatap muka antara pelaku usaha dan pegawai. Berdasarkan hasil wawancara juga didapati dalam proses kegiatan kepelabuhanan, Badan Usaha Pelabuhan BP Batam melayani proses kegiatan perizinan kepelabuhanan dari 60-80 agen kapal /hari dan melayani transaksional jasa kepelabuhanan 400-500 transaksi /hari, dimana setiap agen kapal dapat melakukan lebih dari 1 transaksi nota jasa kepelabuhanan. Sehingga dengan jumlah transaksi berskala besar ini, maka diperlukan sebuah sistem yang dapat mengakomodir kegiatan kepelabuhanan sebagai upaya pencatatan, pemeliharaan, dan penyediaan data secara transparan, mudah diakses, dan online.

BP Batam dalam upaya mewujudkan semua itu maka dibuatlah suatu perizinan yang terintegrasi secara elektronik atau yang disebut perizinan online dengan menggunakan suatu aplikasi “BP Batam Seaport Informations Management System” atau yang disingkat menjadi B-SIMS untuk menjadi sistem perizinan online dalam kegiatan kepelabuhanan di Kota Batam. Aplikasi tersebut diharapkan dapat memberikan kemudahan terkait perizinan dalam berinvestasi seperti: kelola agen kapal, manajemen jasa kepelabuhanan, manajemen dokumen kepelabuhanan, manajemen kedatangan kapal, dan *user access management*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka perumusan masalah pada laporan ini ialah sebagai berikut.

1. Bagaimana merancang dan membangun perangkat lunak perizinan online pada Badan Usaha Pelabuhan BP Batam?
2. Bagaimana memudahkan Badan Usaha Pelabuhan BP Batam dan investor dalam kemudahan terkait perizinan kegiatan kepelabuhanan?

1.3 Batasan Masalah

Berikut merupakan batasan-batasan masalah dalam laporan pembuatan Aplikasi B-SIMS antara lain :

1. Aplikasi B-SIMS yang dirancang nantinya dapat diakses oleh agen perusahaan kapal dan admin sistem.
2. Jenis *framework* yang digunakan pada aplikasi B-SIMS ialah *framework* Laravel dan VueJS.

1.4 Tujuan

Adapun tujuan dari pembuatan aplikasi B-SIMS ini adalah:

1. Menghasilkan aplikasi sebagai sistem perizinan online pada Badan Usaha Pelabuhan BP Batam.
2. Memudahkan dalam mengontrol data masukan dan keluaran dalam penyajian informasi tentang perizinan online, sehingga tersedianya laporan secara cepat, akurat, dan transparan.

1.5 Manfaat

Adapun manfaat aplikasi B-SIMS adalah:

1. Memudahkan investor dan stakeholder dalam pengurusan kegiatan perizinan kepelabuhanan pada Badan Usaha Pelabuhan BP Batam secara online.
2. Memudahkan user dalam mendapatkan informasi serta administrasi dari kegiatan perizinan online kepelabuhanan pada Badan Usaha Pelabuhan BP Batam.
3. Efektivitas dan Efisiensi kinerja pegawai Badan Usaha Pelabuhan BP Batam untuk mendata dan mengelola data kegiatan perizinan online.
4. Pencegahan praktik korupsi dengan menggunakan aplikasi pada pelayanan publik di sektor perizinan pada Badan Usaha Pelabuhan BP Batam.
5. Menjadi referensi penelitian selanjutnya dalam mengembangkan aplikasi perizinan online pada Badan Usaha Pelabuhan.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisikan tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika laporan.

BAB II LANDASAN TEORI

Bab ini berisikan teori yang berhubungan dengan penelitian. Selain itu juga dapat diulas penelitian-penelitian bidang sejenis yang pernah dilakukan serta posisi penelitian tersebut terhadap penelitian sebelumnya.

BAB III ANALISIS DAN PERANCANGAN

Bab ini terdapat uraian rinci tentang langkah-langkah dan metodologi penyelesaian masalah, bahan atau materi, alat yang dipergunakan, metode pengambilan data atau metode analisis hasil, proses pengerjaan serta masalah yang dihadapi disertai cara penyelesaiannya guna menjawab masalah yang timbul pada BAB I dan didukung oleh landasan teori di BAB II.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi implementasi hasil rancangan, termasuk hasil proses produksi dan pasca-produksi, pengujian sistem serta pembahasan. Bagian ini merupakan bagian yang paling penting, karena bagian ini memuat semua temuan ilmiah yang diperoleh. Bagian ini diharapkan dapat memberikan penjelasan ilmiah, yang secara logis dapat menerangkan alasan diperolehnya hasil-hasil tersebut.

BAB V KESIMPULAN DAN SARAN

Bagian ini memuat simpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya serta saran pengembangan dari penelitian yang telah dibuat dan mengungkap aspek-aspek apa saja yang belum terselesaikan. Sehingga dapat menjadi rekomendasi untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Penelitian Sebelumnya

Pada penyusunan tugas akhir ini, terdapat beberapa jenis aplikasi web, dengan menggunakan topik yang hampir sama dengan metode/ide pokok dan implementasi yang berbeda. Penelitian pada tugas akhir ini dibuat dengan mengacu kepada beberapa jurnal penelitian yang berjudul “Pembangunan Sistem Informasi Manajemen Perizinan di Dinas KOMINFO Kota Pasuruan” yang dilakukan oleh Anisah Evi Nur Rohmah, Denny Sagita Rusdianto, dan Nurudin Santoso di tahun 2019. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi berbasis website yang bisa diakses secara online sehingga dapat digunakan untuk melakukan pendaftaran perizinan, melihat status perizinan, dan memvalidasi berkas perizinan untuk SIUP, TDP, TDUP, dan reklame di dinas Kominfo Kota Pasuruan. Adapun metode yang digunakan adalah metode *waterfall*. Penelitian selanjutnya dilakukan oleh Putra Prima Arhandi di tahun 2016 dengan judul “Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan dengan Menggunakan Metode *Back End* dan *Front End*”. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi perijinan terkait tenaga Kesehatan seperti permohonan pemohon perizinan, validasi surat perijinan, dan pembuatan kode nomor surat terkait file surat izin sesuai dengan template yang ada.

Berikut merupakan perbandingan penelitian sesuai dengan penelitian yang penulis sebutkan pada pembahasan sebelumnya pada Tabel 6.1

Tabel 3.1 Perbandingan Penelitian

No	Judul	Metode	Hasil	Penulis & Tahun
1	Pembangunan Sistem Informasi Manajemen Perizinan di Dinas KOMINFO Kota Pasuruan	<i>Waterfall</i>	Sistem Informasi DPMPTSP Berbasis Web	Anisah Evi Nur Rohmah, Denny Sagita Rusdianto, Nurudin Santoso, 2019

2	Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode <i>Back End</i> dan <i>Front End</i>	<i>Waterfall</i>	Aplikasi Perijinan Tenaga Kesehatan Berbasis Web	Putra Prima Arhandi, 2016
---	---	------------------	--	---------------------------

2.2 Definisi Perizinan Online

Salah satu bentuk kewenangan yang dimiliki oleh pemerintah daerah adalah perizinan yang bertujuan untuk mengendalikan perilaku atau kegiatan yang dilakukan oleh golongan maupun individu. Perizinan dapat diartikan sebagai salah satu bentuk pelaksanaan fungsi pengaturan dan bersifat pengendalian yang dimiliki oleh pemerintah terhadap kegiatan-kegiatan yang dilakukan oleh masyarakat. Bentuk perizinan antara lain: pendaftaran, rekomendasi, sertifikasi, penentuan kuota dan izin untuk melakukan sesuatu usaha yang biasanya harus memiliki atau diperoleh suatu organisasi perusahaan atau seseorang sebelum yang bersangkutan dapat melaksanakan suatu kegiatan atau tindakan. Perizinan dapat berupa pendaftaran, rekomendasi, penentuan kuota, dan izin untuk melakukan suatu usaha (Sutedi, 2010).

Penggunaan aplikasi yang terintegrasi secara elektronik pada sektor perizinan disebut pertama kali dalam Peraturan Pemerintah Nomor 24 Tahun 2018 tentang aturan pelaksanaan Online Single Submission (OSS). Sehingga aplikasi perizinan yang terintegrasi dengan elektronik dan berbasis online, dapat menyederhanakan dan memudahkan para pelaku investor atau stakeholder dalam hal kegiatan perizinan dan dalam prosesnya menjadi transparan.

2.3 Definisi Aplikasi

Aplikasi adalah perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan oleh

pengguna. Secara istilah aplikasi berasal dari bahasa Inggris “application” yang berarti penerapan, lamaran ataupun penggunaan. Secara istilah juga pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Jadi, Aplikasi adalah program yang dirancang untuk membuat pengguna lebih produktif dan dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Aplikasi adalah kumpulan dari sub bagian atau komponen apapun baik fisik atau pun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu (Jogiyanto, 2009). Sehingga, aplikasi dimaksudkan untuk memberikan solusi spesifik untuk masalah dan persyaratan pengguna, dengan fokusnya adalah pada penyelesaian masalah untuk kebutuhan yang jelas.

2.4 Definisi Aplikasi Berbasis Web

Web adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (link) satu dokumen dengan dokumen lainnya (hypertext) yang dapat diakses melalui sebuah browser (Yuhefizar, 2012). Gabungan dari semua informasi tersebut disediakan melalui internet dan dapat diakses oleh banyak orang di seluruh dunia selama memiliki koneksi internet.

Aplikasi berbasis web adalah sistem perangkat lunak yang berdasarkan pada teknologi dan standar *World Wide Web Consortium* (W3C), dimana menyediakan sumber daya web spesifik seperti konten dan layanan melalui sebuah antarmuka pengguna dan *browser web* (Simarmata, 2010). Aplikasi berbasis web mulai populer seiring dengan jumlah pengguna internet aktif di seluruh dunia. Selain kemudahan akses melalui website, juga tidak membutuhkan sumber daya yang besar dari sisi perangkat keras atau lunak.

2.5 Framework Laravel

Laravel yang diluncurkan tahun 2011 ini merupakan salah satu framework handal dalam pengembangan aplikasi web menjadi lebih elegan, ekspresif, dan menyenangkan, sehingga framework ini menjadi salah satu framework yang populer di dunia bahkan di Indonesia. Laravel merupakan framework bahasa pemrograman *Hypertext Preprocessor* (PHP) yang ditujukan untuk pengembangan aplikasi berbasis web dengan menerapkan konsep *Model View Controller* (MVC) yang dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, yang artinya bebas digunakan tanpa harus melakukan pembayaran (Aminudin, 2015).

Adapun fitur-fitur keunggulan dari framework Laravel antara lain :

1. Dependency Management

Pada Laravel terdapat sebuah tool yang dinamakan “Composer” yang merupakan “Dependency Manager” dari PHP. Tool ini mengatur ketergantungan antar kumpulan *script/library* yang memungkinkan untuk menggunakan *library* milik pengguna lain untuk diaplikasikan pada proyek Laravel milik sendiri.

2. Modularity

Laravel dirancang untuk membuat aplikasi dari kumpulan modul dan *library* yang terkait dengan composer. Fitur ini membantu untuk menyempurnakan dan meningkatkan fungsionalitas dari aplikasi web yang dibangun, bahkan mengembangkan aplikasi perusahaan dalam skala besar dengan mudah, serta fitur ini juga memudahkan dalam proses *update*.

3. Authentication

Sistem otentikasi pada Laravel sudah difungsikan secara penuh dan prosesnya dilakukan secara otomatis, dimana file konfigurasi ini dapat ditemukan di file ‘config/auth.php’. Di dalam file ini sudah terdapat beberapa opsi otentikasi yang sudah terdokumentasi dengan baik dan dapat digunakan sesuai kebutuhan dengan menjalankan perintah sederhana.

4. Caching

Laravel hampir menyimpan semua data dari tampilan ke rute, dimana membantu Laravel mengurangi waktu pemrosesan sehingga meningkatkan kinerja aplikasi web itu sendiri.

5. Routing

Dalam Laravel semua *request* dipetakan dengan bantuan rute. Rute ini dapat kita kelompokkan, menamainya, menerapkan filter, dan mengikat data model ke rute tersebut. Terdapat tiga kategori routing di Laravel, yaitu: *basic routing*, *route parameters*, dan *named routes*.

6. Restfull Controller

Fitur ini memungkinkan untuk memisahkan logika dalam menyajikan perintah seperti *HTTP GET* dan *POST*. *Restfull Controller* memungkinkan untuk membuat backend *API (Application Programming Interface)* dengan metode *REST (Representative State Transfer)* seperti: *controller API*, *passport* untuk otentikasi, dan *API routes* dalam membuat *CRUD (Create, Update, dan Delete)*.

7. Testing and Debugging

Pada Laravel memiliki metode pengujian aplikasi web secara ekspresif. Framework Laravel mendukung proses pengecekan dengan *PHPUnit* dan file *phpunit.xml* yang dapat disesuaikan dengan aplikasi yang dibangun. Pengujian aplikasi web pada Laravel dengan menerapkan dua pengujian yaitu: Pengujian Unit untuk menguji pengontrol, kelas, dan komponen lain, yang kedua yaitu Pengujian Fitur untuk menguji basis kode sementara pada aplikasi web yang dibangun.

8. Template Engine

Pada Laravel memiliki *template engine* yaitu *Blade*. *Blade* memiliki fungsi pembantu untuk memformat data ke dalam bentuk desain *layout* yang unik. Tidak seperti fitur *templating* PHP lainnya, *Blade* tidak membatasi pemrogram untuk menggunakan kode PHP biasa dalam membuat kode untuk tampilan. *Layout* yang di desain dapat digunakan di tampilan lain sehingga menyediakan konsistensi desain dan struktur selama proses pengembangan. Semua *template Blade* menggunakan

ekstensi file ‘.blade.php’.

9. Configuration Management Features

Pada framework Laravel semua file konfigurasi tersimpan di dalam direktori *config* yang terdokumentasi dengan baik dan dapat diubah sesuai konfigurasi yang tersedia.

10. Eloquent ORM

Dapat disebut juga dengan *query builder*, dimana Laravel database menyediakan antarmuka yang lancar untuk menjalankan *database query*. Laravel berbasis *Eloquent ORM (Object Relation Mapping)* menyediakan dukungan untuk hampir semua mesin basis data, sehingga dalam pengimplementasikan kueri kompleks menggunakan Laravel menjadi sangat mudah.

Selain beberapa keunggulan yang sudah disebutkan diatas, Laravel juga memiliki dua tools andalan yang jarang dimiliki oleh framework lain antara lain:

1. Composer

Composer merupakan tool yang di dalamnya terdapat *dependencies* (ketergantungan) dan kumpulan *library* yang dapat digunakan hanya dengan menjalankan perintah yang sederhana. *Tool* ini memudahkan seorang pemrogram untuk membuat atau mengambil suatu library, sehingga dalam pembuatan aplikasi web menjadi lebih mudah, cepat, dan terstruktur.

2. Artisan

Artisan merupakan *command line interface* Laravel, dimana mencakup sekumpulan perintah dalam pembuatan aplikasi web. Dengan adanya *tool* ini, seorang pemrogram dapat menambahkan berbagai macam fitur baru ke dalam projek Laravel.

2.6 Framework VueJS

VueJS adalah sebuah framework Javascript yang menyebut dirinya “progressive framework”, dikarenakan dapat diimplementasikan dalam sebuah

sistem dengan menggunakan komponen-komponen yang sama untuk mendapatkan efisiensi (*reuse-components*). Pada arsitektur MVC (*Model, View, dan Controller*), VueJS hanya akan mengambil peran pada layer *View* saja, sehingga VueJS dapat digunakan dalam aplikasi skala besar dimana membutuhkan bagian *frontend* yang meningkat.

Adapun fitur-fitur yang terdapat pada VueJS antara lain:

1. Template Syntax
2. Class & Style Binding
3. Conditional & List Rendering
4. Event Handling
5. Form Input Bindings
6. Components Basics
7. Custom Events
8. Transition & Animation
9. Reusability & Composition

2.7 Database Oracle

Database *oracle* memiliki kemampuan untuk menyimpan data dalam ukuran yang sangat besar, sehingga database *oracle* banyak digunakan oleh perusahaan yang sudah maju dan berkembang. Oracle yang merupakan *software* untuk sistem manajemen database relasional (RDBMS) memiliki fungsi memvisualisasikan data dalam bentuk baris dan kolom di dalam tabel digital, dimana menjadi alat manajemen suatu perusahaan untuk membangun sistem manajemen informasi handal demi meningkatkan kinerja perusahaan. Oracle juga dapat menjadi pengontrol akses bagi user yang mengakses data ke sistem database dan perlindungan data perusahaan dengan menggunakan *Real Application Cluster* (RAP). RAP ini berfungsi agar tidak mempengaruhi kinerja server database saat terjadi kegagalan sistem, sehingga mampu mengelola data yang selalu bertambah besar dengan menambahkan kapasitasnya menjadi lebih besar dengan aman, dan dapat melakukan manajemen user terhadap pengaturan

hak akses ke dalam database.

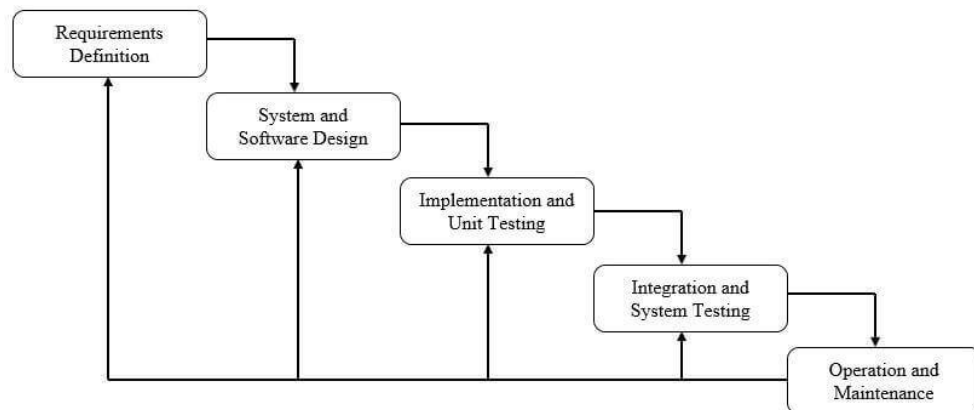
2.8 MySQL (*My Structured Query Language*)

Menurut (Kristanto A. , 2010), “MySQL merupakan sebuah program pembuat database yang bersifat open source. Selain itu MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FireBSD, Mac OS X Server, Solaris, Amiga, dan masih banyak lagi.

Modul yang sudah termasuk yaitu command-line (perintah: mysql dan mysql admin). Juga dapat diunduh dari situs MySQL yaitu sebuah modul berbasis grafik (GUI): MySQL Administrator dan MySQL Query. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu php MyAdmin. Untuk perangkat lunak untuk administrasi basis data MySQL yang dijual secara komersial antara lain: MySQL front, Navicat dan EMS SQL Manager for MySQL.

2.9 Metode *Waterfall*

Pembangunan pada suatu sistem secara keseluruhan dilakukan dengan beberapa tahapan di dalam prosesnya. Metode pengembangan perangkat lunak dikenal juga dengan istilah *Software Development Life Cycle* (SDLC). Metode *waterfall* (air terjun) merupakan metode tertua dan pendekatan SDLC paling awal yang digunakan dalam pengembangan perangkat lunak dikarenakan sifatnya yang natural. Tahapan utama dari metode *waterfall* menurut (Sommerville & Ian, 2011), yaitu: *Requirements Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing*, dan *Operational and Maintenance*.



Gambar 3.1 Tahapan Metode Waterfall

Tahap-tahap yang dilakukan dalam penggunaan metode *waterfall* adalah:

1. *Requirements Analysis and Definition*

Seorang pengembang melakukan pengumpulan informasi sebelum memulai melakukan pengembangan perangkat lunak. Pengumpulan informasi dapat diperoleh dengan berbagai cara, seperti: diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh secara lengkap kemudian diolah dan dilakukan analisa terhadap kebutuhan pengguna akan perangkat lunak yang dibutuhkan.

2. *System and Software Design*

Informasi mengenai kebutuhan spesifikasi sistem dari tahapan sebelumnya yang telah dianalisa, selanjutnya kemudian diimplementasikan pada desain pengembangan. Proses ini berfokus pada pembangunan struktur data, arsitektur perangkat lunak, perancangan interface, perancangan fungsi internal dan eksternal serta detail dari seluruh algoritma prosedural.

3. *Implementation and Unit Testing*

Pada tahapan ini adalah tahapan pembuatan aplikasi oleh para pengembang dengan kode-kode bahasa pemrograman tertentu. Penulisan koding mengacu pada dokumen-dokumen yang telah dibuat sebelumnya, dimana pembuatan perangkat lunak dibagi

menjadi modul-modul kecil yang akan digabungkan dalam tahap berikutnya. Pada fase ini juga dilakukan pengujian terhadap fungsionalitas modul apakah sudah sesuai dengan spesifikasi yang diinginkan.

4. *Integration and System Testing*

Tahapan ini adalah dimana seluruh unit atau modul perangkat lunak diintegrasikan ke dalam sistem secara keseluruhan. Setelah itu dilakukan pemeriksaan dan pengujian sistem secara keseluruhan jika terdapat kesalahan atau kegagalan (*error*).

5. *Operational and Maintenance*

Tahapan terakhir dari metode *waterfall* ini, dimana dilakukan tahapan penginstalan sistem kepada pengguna dan dilakukan pemeliharaan. Pemeliharaan oleh pengembang adalah sebagai bentuk tanggung jawab meliputi perbaikan atas kesalahan yang belum terdeteksi sebelumnya, dan penambahan modul-modul baru untuk pengembangan aplikasi.

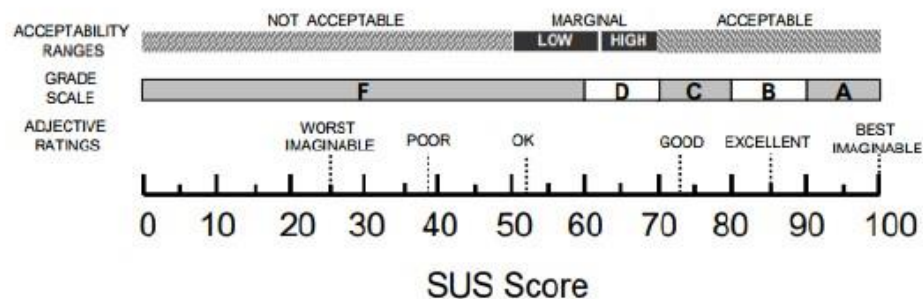
2.10 SUS (*System Usability Scale*)

SUS ini merupakan salah satu alat pengujian usability yang paling populer, yang dikembangkan oleh Jhon Brooke pada tahun 1986. SUS ini merupakan skala *usability* yang populer, efektif, dan murah. Pengujian ini dilakukan dengan mengukur interaksi antara sistem dan pengguna sistem menggunakan serangkaian kuisisioner yang dapat mengolah data terhadap tingkat *usability* yang baik. SUS memiliki 10 pertanyaan dengan 5 pilihan jawaban, dengan pilihan jawaban terdiri dari sangat tidak setuju sampai sangat setuju. SUS memiliki skor 0 dan skor maksimal 100, dan sebuah sistem dikatakan memiliki tingkat *usability* yang baik apabila memiliki nilai SUS di atas 70. Berikut adalah gambar kategori

pengujian skor SUS.

Menurut penelitian (Z & H, 2016,2017), terdapat 10 pertanyaan standar kuisioner dari *System Usability Scale* (SUS) dengan skor masing-masing jawaban mulai dari 1 sampai 5. Berikut adalah standar kuisioner SUS.

Tabel 3.2 Standar Kuisioner SUS



Gambar 3.2 Kategori Skor SUS

No	Item Pernyataan	1= Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Ragu-Ragu, 4 = Setuju, 5 = Sangat Setuju				
1	Saya berpikir akan menggunakan sistem ini lagi	1	2	3	4	5
2	Saya merasa sistem ini rumit untuk digunakan (memuat banyak hal yang tidak perlu)	1	2	3	4	5
3	Saya merasa sistem ini mudah digunakan	1	2	3	4	5
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini	1	2	3	4	5
5	Saya merasa fitur-fitur yang disediakan pada situs ini dirancang dan disiapkan dengan baik	1	2	3	4	5

6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)	1	2	3	4	5
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat	1	2	3	4	5
8	Saya merasa sistem ini membingungkan	1	2	3	4	5
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini	1	2	3	4	5
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini	1	2	3	4	5

Setelah melakukan pengumpulan data dari responden berdasarkan standar kuisioner diatas, kemudian data tersebut dihitung. Ada beberapa aturan dalam perhitungan skor SUS seperti dibawah ini.

1. Setiap pernyataan bernomor ganjil, skor setiap pernyataan yang didapat dari skor pengguna akan dikurangi 1.
2. Setiap pernyataan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pernyataan yang didapat dari pengguna.
3. Skor SUS didapat dari hasil penjumlahan skor setiap pernyataan yang kemudian dikali 2.5.

Aturan perhitungan skor hanya berlaku untuk 1 responden. Skor SUS dari masing-masing responden dicari skor rata-ratanya dengan menjumlahkan semua skor dan dibagi dengan jumlah responden. Berikut rumus menghitung skor SUS.

$$\bar{x} = \frac{\sum x}{n}$$

$$\begin{array}{lcl} \bar{x} & = & \text{skor rata-rata} \\ \sum x & = & \text{jumlah skor SUS} \\ n & = & \text{jumlah responden} \end{array}$$

2.11 Skala Likert

Skala *likert* merupakan salah satu bentuk skala yang dilakukan untuk mengumpulkan data demi mengetahui atau mengukur data yang bersifat

kualitatif. Dengan skala likert ini dapat digunakan untuk mengukur sikap dan pendapat, atau persepsi seseorang (Situmorang, 2010) dengan cara meminta responden untuk melengkapi kuisioner yang mengharuskan mereka untuk menunjukkan tingkat persetujuannya terhadap serangkaian pertanyaan. Serangkaian pertanyaan ini menjadi suatu yang disebut variabel penelitian yang ditetapkan spesifik oleh peneliti.

Dalam menentukan nilai dari skala likert, responden dapat menentukan jawaban, yang dimulai dari tingkat persetujuan dengan pilihan seperti format dibawah ini.

Tabel 3.3 Skala Likert

Pilihan Jawaban	Skor
Sangat setuju/Sangat baik/Sangat puas	5
Setuju/Puas/Baik	4
Ragu-ragu/Netral/Cukup	3
Tidak setuju/Tidak puas/Kurang baik	2
Sangat tidak setuju/Sangat tidak puas/Sangat tidak baik	1

Untuk menghitung total skor berdasarkan pilihan tersebut, dapat dilakukan dengan rumus berikut:

$$\text{Rumus Pilihan Skor} = T \times Pn$$

Dimana T adalah jumlah responden yang memilih dan Pn adalah pilihan angka skor *likert*.

Total skor akan didapatkan dari penjumlahan masing-masing pilihan skor yang telah didapatkan hasilnya menggunakan rumus pilihan skor. Adapun untuk mengetahui hasil interpretasi, maka harus didapatkan terlebih dahulu skor tertinggi (Y) dan skor terendah (X) menggunakan rumus dibawah ini :

$$Y = \text{Skor tertinggi Likert} \times \text{jumlah panelis}$$

$$X = \text{Skor terendah likert} \times \text{jumlah panelis}$$

Hasil total skor dan nilai Y yang telah didapatkan dari proses sebelumnya, akan diproses kembali untuk mendapatkan hasil dalam satuan persen. Adapun

rumus index dalam persen adalah sebagai berikut :

$$\text{Rumus Index \%} = \frac{\text{Total Skor}}{Y} \times 100$$

Untuk mengetahui tingkat penilaian dari hasil perhitungan rumus index persen, maka dilakukan interpretasi skor terlebih dahulu menggunakan rumus interval (I) berikut :

$$I = \frac{100}{\text{Jumlah Skor (Likert)}}$$

Interval didapatkan dari perhitungan 100 dibagi dengan jumlah skor *likert* yang berjumlah 5 pilihan yaitu sangat setuju, setuju, netral, tidak setuju, sangat tidak setuju. Maka akan didapatkan hasil interval yaitu 20. Adapun rinciannya adalah sebagai berikut :

- Angka 80% - 100% = Sangat Setuju (SS)
- Angka 60% - 79.99% = Setuju (S)
- Angka 40% - 59.99% = Netral (N)
- Angka 20% - 39.99% = Tidak Setuju (TS)
- Angka 0% - 19.99% = Sangat Tidak Setuju (STS)

Interval diatas merupakan penentu sebagai hasil akhir penilaian dari interpretasi responden terhadap sebuah pernyataan apakah masuk dalam kategori SS, S, N, TS, dan STS dengan menggunakan rumus index % (Darmadi, 2011).

2.12 Black-Box Testing

Black-Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Black-box testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Jaya, 2018). Setelah dilakukan pengujian, sistem akan dievaluasi apakah hasilnya telah sesuai dengan proses bisnis yang diharapkan atau tidak (Nusantara Halim & Amirul Nashrullah).

BAB III

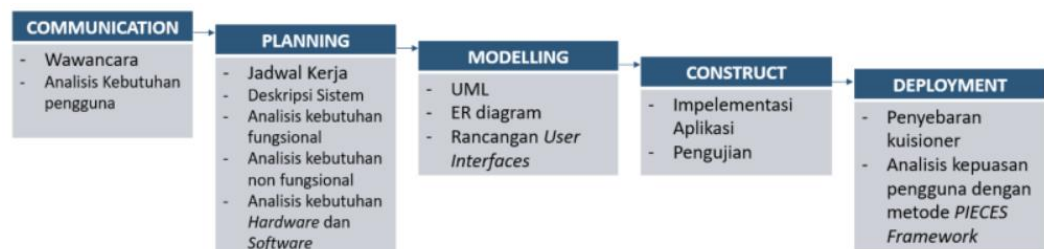
ANALISIS DAN PERANCANGAN

3. Metode Penyelesaian Masalah

Metode yang digunakan untuk mengembangkan sistem perangkat lunak ini adalah metode *waterfall*. Adapun 5 fase dalam metode *waterfall* ini yaitu :

1. *Communication (Project Initiation & Requirements Gathering)*
2. *Planning (Estimating, Scheduling, Tracking)*
3. *Modelling (Analysis & Design)*
4. *Consctruction (Code & Test)*
5. *Deployment (Delivery, Support, Feedback)*

Pada bab ini fase yang akan dibahas meliputi tiga fase, yaitu *communication*, *planning*, dan *modelling*.



Gambar 3.1 Fase-Fase Metode Waterfall

3.1 Communication

Metode yang digunakan untuk mengembangkan sistem perangkat lunak ini adalah tahap ini dilakukan komunikasi berupa wawancara tidak terstruktur kepada pimpinan departemen Operasional TIK di BP Batam. Adapun hasil wawancara yang dilakukan dengan Bapak Rizal Safani selaku *Asisten Manager* Solusi dan Infrastruktur TIK adalah sebagai berikut:

- Apa masalah yang sedang dihadapi oleh Badan Usaha Pelabuhan?
Jawab: Belum adanya suatu sistem terkait perizinan kegiatan kepelabuhanan untuk para pelaku usaha secara online.
- Apakah sudah ada aplikasi untuk kegiatan perizinan online di Badan Usaha Pelabuhan?

Jawab: Belum ada aplikasi untuk para pelaku usaha, selama ini aplikasi terkait perizinan hanya digunakan di internal oleh pegawai Badan Usaha Pelabuhan.

- Apakah ada rencana dari Operasional TIK & Badan Usaha Pelabuhan untuk membuat aplikasi perizinan online untuk kegiatan kepelabuhanan?

Jawab: Ada, dan diharapkan aplikasi ini segera dilakukan agar memudahkan kegiatan perizinan kepelabuhanan.

- Hasil apa yang diharapkan dengan adanya aplikasi perizinan online tersebut?

Jawab: Hasilnya agar memudahkan dalam kegiatan perizinan kepelabuhanan dan menghindari interaksi tatap muka antara pelaku usaha dan pegawai.

- Aplikasi seperti apa yang diinginkan?

Jawab: Hasilnya dimana sistem mudah dalam penggunaannya, sistem dilakukan secara online dan *responsive*.

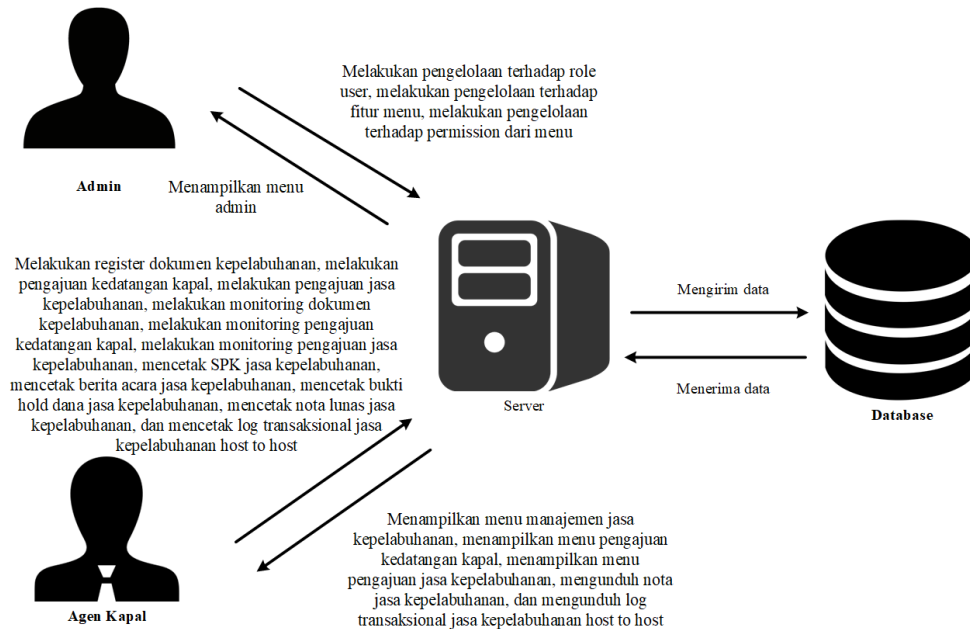
- Siapa yang akan menggunakan aplikasi ini?

Jawab: Operator Badan Usaha Pelabuhan, agen kapal, dan admin sistem.

3.2 *Planning*

Pada tahap ini menjabarkan lebih rinci kebutuhan pengguna (*user requirement*) menjadi kebutuhan sistem yang di dalamnya telah tercakup seluruh kebutuhan pengguna. Adapun kebutuhan pengguna yang dianalisis meliputi: deskripsi sistem, analisis kebutuhan sistem yang terdiri dari analisis kebutuhan fungsional dan non fungsional, analisis kebutuhan perangkat keras, dan analisis kebutuhan perangkat lunak. Selain itu pada tahapan ini juga menjelaskan penjadwalan waktu yang diperlukan selama mengembangkan aplikasi.

3.2.1 Deskripsi Sistem



Gambar 3.2 Deskripsi Sistem B-SIMS

Aplikasi yang dikembangkan merupakan aplikasi berbasis web yang akan dibangun dan dioperasikan pada *server*. Aplikasi bertujuan untuk melakukan kelola kegiatan kapal lalu akan disimpan di dalam database.

Pengguna aplikasi terdiri dari 2 kategori, yaitu *Admin* dan *Agan Kapal*. Adapun penjelasan ruang lingkup pengguna adalah sebagai berikut:

1. *Admin*

Admin merupakan seseorang yang memiliki hak akses penuh sebagai pengelola aplikasi meliputi: pengelolaan terhadap *role user*, pengelolaan terhadap fitur menu aplikasi, dan pengelolaan terhadap *permission* dari fitur menu aplikasi.

2. *Agan Kapal*

Agan Kapal merupakan seseorang (*user*) yang diberikan hak akses oleh *Admin* untuk menggunakan fitur aplikasi kelola data agen perusahaan kapal meliputi: pengelolaan *register* dokumen kepelabuhanan, pengelolaan pengajuan kedatangan kapal, pengelolaan pengajuan jasa kepelabuhanan, melakukan

monitoring dokumen kepelabuhanan, melakukan monitoring pengajuan kedatangan kapal, melakukan monitoring pengajuan jasa kepelabuhanan, mencetak SPK (Surat Perintah Kerja) jasa kepelabuhanan, mencetak berita acara jasa kepelabuhanan, mencetak bukti hold dana jasa kepelabuhanan, mencetak nota lunas jasa kepelabuhanan, dan mencetak *log* transaksional jasa kepelabuhanan *host to host*.

3.2.2 Analisis Sistem

1. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah analisis yang dibutuhkan untuk menentukan kebutuhan yang memiliki keterkaitan langsung dengan sistem meliputi: kebutuhan *user*/pengunjung dan kebutuhan *admin*/pengelola. Adapun kebutuhan fungsional pada aplikasi yang akan dibangun dijabarkan pada Tabel 7.1 berikut:

Tabel 3.1 Kebutuhan Fungsional

ID	Fungsi
F001	Pengguna dapat melakukan <i>login</i>
F002	Pengguna dapat menambah <i>role user</i>
F003	Pengguna dapat menghapus <i>role user</i>
F004	Pengguna dapat merubah hak akses role user terhadap fitur menu
F005	Pengguna dapat menambah fitur menu
F006	Pengguna dapat menghapus fitur menu
F007	Pengguna dapat menambah <i>permission</i> dari menu
F008	Pengguna dapat menghapus <i>permission</i> dari menu
F009	Pengguna dapat menambah dokumen kepelabuhanan
F010	Pengguna dapat melihat daftar dokumen kepelabuhanan
F011	Pengguna dapat melakukan pencarian daftar dokumen kepelabuhanan
F012	Pengguna dapat melihat detail daftar dokumen kepelabuhanan

F013	Pengguna dapat menghapus dokumen kepelabuhanan
F014	Pengguna dapat mengunduh dokumen kepelabuhanan
F015	Pengguna dapat menambah permohonan pengajuan kedatangan kapal
F016	Pengguna dapat melihat daftar permohonan pengajuan kedatangan kapal
F017	Pengguna dapat melakukan pencarian daftar permohonan pengajuan kedatangan kapal
F018	Pengguna dapat melihat detail daftar permohonan pengajuan kedatangan kapal
F019	Pengguna dapat melakukan <i>submit</i> persetujuan permohonan pengajuan kedatangan kapal
F020	Pengguna dapat menambah permohonan pengajuan jasa kepelabuhanan
F021	Pengguna dapat melihat daftar permohonan pengajuan jasa kepelabuhanan
F022	Pengguna dapat melakukan pencarian daftar permohonan pengajuan jasa kepelabuhanan
F023	Pengguna dapat melihat detail daftar permohonan pengajuan jasa kepelabuhanan
F024	Pengguna dapat melakukan <i>submit</i> persetujuan daftar permohonan pengajuan jasa kepelabuhanan
F025	Pengguna dapat mencetak Surat Perintah Kerja (SPK) Jasa Kepelabuhanan
F026	Pengguna dapat mencetak berita acara jasa kepelabuhanan
F027	Pengguna dapat mencetak bukti <i>hold</i> dana jasa kepelabuhanan
F028	Pengguna dapat mencetak nota lunas jasa kepelabuhanan
F029	Pengguna dapat mencetak <i>log</i> transaksional jasa kepelabuhanan <i>host to host</i>

2. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan yang berkaitan dengan fitur tertentu di dalam sistem. Adapun kebutuhannya sebagai berikut:

- a. Sistem dapat dijalankan pada *PC* dengan sistem operasi *Windows 7* dan versi di atasnya, serta sudah ter-*install NET.Framework 3.0*
- b. Sistem memiliki antarmuka yang dapat dipahami

3. Analisis Kebutuhan Perangkat Keras

Adapun perangkat keras yang akan digunakan dalam pembuatan Tugas Akhir dijabarkan pada Tabel 7.2 sebagai berikut:

Tabel 3.2 Analisis Perangkat Keras

Perangkat Keras	Spesifikasi
Laptop HP Pavilion Power Laptop 15-cb0xx	<ul style="list-style-type: none"> - Processor Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz - RAM 12,00 GB - Display Card NVIDIA GeForce GTX 1050 - 1000 GB Harddisk

4. Analisis Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam pembuatan aplikasi diantaranya:

- a. Sistem Operasi *Windows 10 Home 64-bit* yang sudah ter-install .NET Framework
- b. *Ms Visio 2019* untuk mendesain *Use Case Diagram*, *Activity Diagram*, *ER Diagram*, dan *Sequence Diagram*
- c. *Microsoft Visual Studio 2017* digunakan untuk mendesain antarmuka
- d. *MySQL* dan *Oracle* digunakan untuk *database server*
- e. *Navicat 15* digunakan untuk menjalankan perintah-perintah *SQL*.

3.3 Modelling

Pada tahap ini dilakukan perancangan sistem, dimana pada tahap ini dilakukan penentuan proses dan data yang diperlukan oleh sistem. Adapun rancangan yang digunakan dalam proses ini menggunakan metode UML (*Unified Modeling Language*). Adapun model UML terdiri dari *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

3.3.1 Use Case Diagram

Use Case diagram adalah diagram yang menggambarkan kebutuhan

sistem dari sudut pandang *user*, yang memperlihatkan hubungan-hubungan yang terjadi antara *actors* dengan *use case* dalam sistem (Munawar, 2005). Pada sistem terdiri dari 2 (aktor), yaitu admin dan agen kapal. Aktor berinteraksi dengan aplikasi berbasis website. Untuk dapat mengakses dan menggunakan aplikasi, aktor diharuskan melakukan login terlebih dahulu. Adapun kebutuhan fitur tampilan antarmuka antara lain:

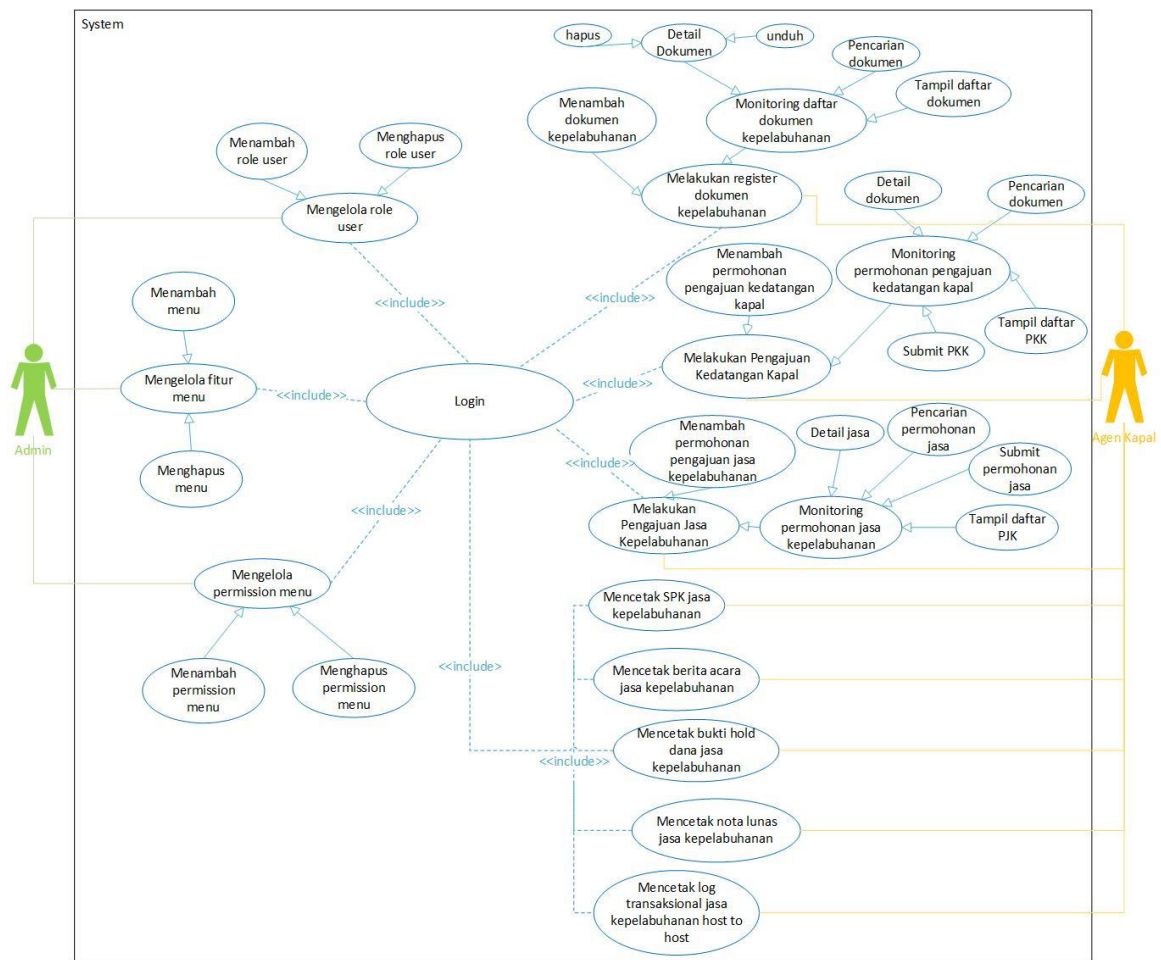
1. Admin dan Agen Kapal dapat menggunakan fitur *Login* (masuk kedalam sistem dengan memasukkan email dan password)
2. Admin dapat menggunakan fitur Mengelola Role User (melakukan penambahan user berdasarkan *role user* yang sudah ada dan melakukan penghapusan *user*)
3. Admin dapat menggunakan fitur Mengelola Fitur Menu (melakukan penambahan menu dan melakukan penghapusan menu sesuai *role user* yang ada, seperti: menu PKK (Pengajuan Kedatangan Kapal) hanya bisa diakses oleh Agen Kapal saja)
4. Admin dapat menggunakan fitur Mengelola *Permission* Menu (melakukan penambahan *permission* menu dan melakukan penghapusan *permission* menu sesuai *role user* yang ada, seperti: menu PKK (Pengajuan Kedatangan Kapal) untuk *user* Agen Kapal hanya memiliki *permission* menu yaitu: *View*, *Create*, *Update*, dan *Delete*)
5. Agen Kapal dapat menggunakan fitur *Register* Dokumen Kepelabuhanan yang dapat dilakukan berulang-ulang sesuai jenis file dokumen yang ada (melakukan penambahan dokumen kepelabuhanan berdasarkan lokasi terminal yang ada dan melakukan penghapusan dokumen kepelabuhanan, seperti: Perusahaan Agen Kapal yang bernama PT A melalui Terminal B ingin meng-*upload* dokumen Form 1A (Form rencana labuh/tambat/pandu/tunda/air)
6. Agen Kapal dapat menggunakan fitur Pengajuan Kedatangan Kapal/PKK (melakukan pengajuan kedatangan kapal sesuai nama kapal, jenis kegiatan, jenis detail kegiatan, jenis trayek, dan rencana

pelayaran, seperti: Perusahaan Agen Kapal yang bernama PT A melakukan kegiatan pengisian bahan bakar dengan jenis bahan bakar MDO (*Marine Diesel Oil*) dengan jenis trayek Tongkang Reguler dan rencana pelayaran meliputi: lokasi terminal, tanggal kedatangan/keberangkatan, jam kedatangan/keberangkatan, negara asal/tujuan, jumlah awak/penumpang, dan sebagainya)

7. Agen Kapal dapat menggunakan fitur Pengajuan Jasa Kepelabuhanan/PJK (setelah mengajukan PKK, agen kapal dan melakukan pengajuan jasa kepelabuhanan meliputi: Jasa Bongkar Muat, Jasa Penumpukan, Jasa Sewa Chasis, Jasa Labuh Tambat, dan Jasa Pengisian Air dengan mengisi detail masing-masing jasa seperti: jenis trayek, jenis jasa, jenis kegiatan, lokasi terminal, dan sebagainya dengan meng-*upload* surat-surat yang diperlukan).
8. Agen Kapal dapat menggunakan fitur *Monitoring* Dokumen Kepelabuhanan (melakukan *monitoring* seperti: melihat tabel daftar dokumen kepelabuhanan yang sudah di *registrasi*, melakukan pencarian daftar dokumen kepelabuhanan, menghapus dokumen yang sudah di *registrasi*, dan mengunduh dokumen yang ter-*upload*)
9. Agen Kapal dapat menggunakan fitur *Monitoring* Pengajuan Kedatangan Kapal (melakukan *monitoring* meliputi: melihat tabel daftar permohonan pengajuan kedatangan kapal yang sudah diajukan, melakukan pencarian permohonan pengajuan kedatangan kapal, melihat detail dari permohonan pengajuan kedatangan kapal, melakukan *submit* persetujuan dokumen pengajuan kedatangan kapal, dan melihat status *progress* dari permohonan pengajuan kedatangan kapal)
10. Agen Kapal dapat menggunakan fitur *Monitoring* Pengajuan Jasa Kepelabuhanan (melakukan *monitoring* meliputi: melihat tabel daftar permohonan pengajuan jasa kepelabuhanan yang sudah diajukan, melakukan pencarian permohonan pengajuan jasa kepelabuhanan, melihat detail dari permohonan pengajuan jasa kepelabuhanan,

melakukan *submit* persetujuan dokumen pengajuan jasa kepelabuhanan, dan melihat status *progress* dari permohonan pengajuan jasa kepelabuhanan)

11. Agen Kapal dapat menggunakan fitur Mencetak SPK Jasa Kepelabuhanan (seperti melakukan unduh dan cetak Surat Perintah kerja (SPK) untuk jasa pandu)
12. Agen Kapal dapat menggunakan fitur Mencetak Berita Acara Jasa Kepelabuhanan (seperti melakukan unduh dan cetak berita acara untuk jasa pandu)
13. Agen Kapal dapat menggunakan fitur Mencetak Nota Lunas Jasa Kepelabuhanan (seperti melakukan unduh dan cetak nota lunas untuk jasa pandu)
14. Agen Kapal dapat menggunakan fitur Mencetak *Log* Transaksional Jasa Kepelabuhanan *Host to Host* (seperti melakukan unduh dan cetak *history* transaksi dari permohonan jasa kepelabuhanan yang diajukan)



Gambar 3.3 Use Case Diagram

3.3.2 Skenario Use Case

1. Skenario Use Case Login Admin

Tabel 3.3 Skenario Use Case Login Admin

Aktor	: admin
Kondisi awal	: admin belum masuk sistem
Kondisi akhir	: admin berhasil masuk sistem
Skenario	1. Admin memilih menu <i>login</i> 2. Admin memasukkan <i>email</i> dan <i>password</i> 3. Sistem melakukan validasi akun berdasarkan <i>role user</i> yang ada

	4. Admin mengakses halaman utama
Skenario alternatif	: jika <i>email</i> dan <i>password</i> salah, maka sistem akan menampilkan pesan kesalahan dan mengembalikan ke halaman <i>login</i>

2. Skenario *Use Case* Login Agen Kapal

Tabel 3.4 Skenario *Use Case Login Agen Kapal*

Aktor	: agen kapal
Kondisi awal	: agen kapal belum masuk sistem
Kondisi akhir	: agen kapal berhasil masuk sistem
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu <i>registrasi</i> pada sistem 2. Agen kapal memasukkan data berupa <i>email</i> dan <i>password</i> ke dalam <i>form</i> 3. Agen kapal melakukan konfirmasi <i>email</i> 4. Agen kapal memilih menu <i>login</i> 5. Agen kapal memasukkan <i>email</i> dan <i>password</i> 6. Sistem melakukan validasi akun berdasarkan <i>role user</i> yang ada 7. Agen kapal mengakses halaman utama
Skenario alternatif	: jika <i>email</i> dan <i>password</i> salah, maka sistem akan menampilkan pesan kesalahan dan mengembalikan ke halaman <i>login</i>

3. Skenario *Use Case* Menambah *Role User*

Tabel 3.5 Skenario *Use Case Menambah Role User*

Aktor	: admin
Kondisi awal	: data <i>role user</i> kosong
Kondisi akhir	: data <i>role user</i> disimpan di dalam <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Admin memilih menu <i>permission</i>

	2. Admin memilih menu tambah <i>role user</i> 3. Admin mengisi <i>role user</i> baru 4. Admin menekan <i>save</i> 5. Sistem melakukan operasi simpan ke <i>database</i> 6. Sistem menampilkan halaman admin
Skenario alternatif	: jika data <i>role user</i> salah, maka sistem akan menampilkan pesan kesalahan dan mengembalikan ke halaman admin

4. Skenario *Use Case* Merubah Hak Akses *Role User*

Tabel 3.6 Skenario Use Case Merubah Hak Akses Role User

Aktor	: admin
Kondisi awal	: data hak akses <i>role user</i> kosong
Kondisi akhir	: data hak akses <i>role user</i> tersedia
Skenario	1. Admin memilih data <i>role user</i> 2. Admin memilih data menu 3. Admin memberi tanda <i>checklist</i> hak akses pada menu
Skenario alternatif	: -

5. Skenario *Use Case* Menghapus *Role User*

Tabel 3.7 Skenario Use Case Menghapus Role User

Aktor	: admin
Kondisi awal	: data <i>role user</i> tersedia
Kondisi akhir	: data <i>role user</i> dihapus dari <i>database</i>
Skenario	1. Admin memilih data <i>role user</i> 2. Admin menekan <i>delete</i> 3. Sistem menampilkan pesan konfirmasi 4. Sistem melakukan operasi hapus ke <i>database</i>

	5. Sistem menampilkan halaman admin
Skenario alternatif	: -

6. Skenario *Use Case* Menambah Menu

Tabel 3.8 Skenario *Use Case* Menambah Menu

Aktor	: admin
Kondisi awal	: data menu kosong
Kondisi akhir	: data menu disimpan di dalam <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Admin memilih menu <i>permission</i> 2. Admin memilih tambah menu 3. Admin mengisi data menu 4. Admin menekan <i>save</i> 5. Sistem melakukan operasi simpan ke <i>database</i> 6. Sistem menampilkan pesan data tersimpan 7. Sistem menampilkan halaman menu
Skenario alternatif	: -

7. Skenario *Use Case* Menghapus Menu

Tabel 3.9 Skenario *Use Case* Menghapus Menu

Aktor	: admin
Kondisi awal	: data menu tersedia
Kondisi akhir	: data menu dihapus dari <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Admin memilih data menu 2. Admin menekan <i>delete</i> 3. Sistem menampilkan pesan konfirmasi 4. Sistem melakukan operasi hapus ke <i>database</i> 5. Sistem menampilkan halaman admin
Skenario alternatif	: -

8. Skenario *Use Case* Menambah *Permission* Menu

Tabel 3.10 Skenario Use Case Menambah Permission Menu

Aktor	: admin
Kondisi awal	: data <i>permission</i> menu kosong
Kondisi akhir	: data <i>permission</i> menu tersimpan di <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Admin memilih menu <i>permission</i> 2. Admin memilih tambah menu <i>permission</i> pada menu yang tersedia 3. Admin mengisi data menu <i>permission</i> 4. Admin menekan <i>save</i> 5. Sistem melakukan operasi simpan ke <i>database</i> 6. Sistem menampilkan pesan data tersimpan 7. Sistem menampilkan halaman menu
Skenario alternatif	: -

9. Skenario Use Case Menghapus Permission Menu

Tabel 3.11 Skenario Use Case Menghapus Permission Menu

Aktor	: admin
Kondisi awal	: data <i>permission</i> menu tersedia
Kondisi akhir	: data <i>permission</i> menu dihapus dari <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Admin memilih data menu <i>permission</i> 2. Admin menekan <i>delete</i> 3. Sistem menampilkan pesan konfirmasi 4. Sistem melakukan operasi hapus ke <i>database</i> 5. Sistem menampilkan halaman menu
Skenario alternatif	: -

10. Skenario Use Case Menambah Dokumen Kepelabuhanan

Tabel 3.12 Skenario Use Case Menambah Dokumen Kepelabuhanan

Aktor	: agen kapal
-------	--------------

Kondisi awal	: data dokumen kosong
Kondisi akhir	: data dokumen disimpan di <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu <i>register</i> dokumen kepelabuhanan 3. Agen kapal mengisi data dokumen sesuai jenis file yang ada 4. Agen kapal menekan tambah 5. Sistem menampilkan data di tabel <i>register</i> dokumen 6. Agen kapal menekan simpan 7. Sistem melakukan operasi simpan ke <i>database</i> 8. Sistem menampilkan pesan data tersimpan 9. Sistem menampilkan halaman daftar dokumen kepelabuhanan
Skenario alternatif	: -

11. Skenario *Use Case* Tampil Daftar Dokumen Kepelabuhanan

Tabel 3.13 Skenario Use Case Tampil Daftar Dokumen Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: daftar dokumen kosong
Kondisi akhir	: daftar dokumen disimpan di <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu daftar dokumen kepelabuhanan 3. Sistem mengambil data dari <i>database</i> 4. Sistem menampilkan daftar dokumen kepelabuhanan yang sudah di <i>register</i>

Skenario alternatif	: -
---------------------	-----

12. Skenario *Use Case* Melakukan Pencarian Daftar Dokumen
Kepelabuhanan

**Tabel 3.14 Skenario Use Case Melakukan Pencarian Daftar Dokumen
Kepelabuhanan**

Aktor	: agen kapal
Kondisi awal	: data pencarian dokumen kepelabuhanan kosong
Kondisi akhir	: data pencarian dokumen kepelabuhanan tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu daftar dokumen kepelabuhanan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal memasukkan kata kunci dokumen 5. Sistem menampilkan data dokumen kepelabuhanan spesifik
Skenario alternatif	: -

13. Skenario *Use Case* Melihat Detail Dokumen Kepelabuhanan

**Tabel 3.15 Skenario Use Case Melihat Detail Dokumen
Kepelabuhanan**

Aktor	: agen kapal
Kondisi awal	: detail dokumen kepelabuhanan kosong
Kondisi akhir	: detail dokumen kepelabuhanan tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu daftar dokumen kepelabuhanan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal melakukan klik 2x pada data

	dokumen kepelabuhanan 5. Sistem mengambil data dari <i>database</i> 6. Sistem menampilkan detail data dokumen kepelabuhanan
Skenario alternatif	: -

14. Skenario *Use Case* Mengunduh Dokumen Kepelabuhanan

Tabel 3.16 Skenario Use Case Mengunduh Dokumen Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: hasil unduhan dokumen kepelabuhanan kosong
Kondisi akhir	: hasil unduhan dokumen kepelabuhanan tersedia
Skenario	1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu daftar dokumen kepelabuhanan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal melakukan klik 2x pada data dokumen kepelabuhanan 5. Sistem mengambil data dari <i>database</i> 6. Agen kapal menekan <i>download</i> 7. Sistem mengambil data dari <i>database</i> 8. Agen kapal menyimpan data ke komputer
Skenario alternatif	: -

15. Skenario *Use Case* Menghapus Dokumen Kepelabuhanan

Tabel 3.17 Skenario Use Case Menghapus Dokumen Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: data dokumen kepelabuhanan tersedia
Kondisi akhir	: data dokumen kepelabuhanan dihapus dari

	<i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu daftar dokumen kepelabuhanan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal melakukan klik 2x pada data dokumen kepelabuhanan 5. Sistem mengambil data dari <i>database</i> 6. Agen kapal menekan hapus 7. Sistem melakukan operasi hapus ke <i>database</i> 8. Sistem menampilkan halaman daftar dokumen kepelabuhanan
Skenario alternatif	: -

16. Skenario *Use Case* Menambah Permohonan Pengajuan Kedatangan Kapal

Tabel 3.18 Skenario Use Case Menambah Permohonan Pengajuan Kedatangan Kapal

Aktor	: agen kapal
Kondisi awal	: data pkk kosong
Kondisi akhir	: data pkk disimpan di <i>database</i>
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan kedatangan kapal 2. Agen kapal memilih menu <i>register</i> permohonan baru 3. Agen kapal mengisi data PKK 4. Agen kapal menekan simpan 5. Sistem melakukan operasi simpan ke <i>database</i> 6. Sistem menampilkan pesan data tersimpan 7. Sistem menampilkan halaman daftar

	permohonan PKK
Skenario alternatif	: -

17. Skenario *Use Case* Tampil Daftar Permohonan Pengajuan Kedatangan Kapal (PKK)

Tabel 3.19 Skenario Use Case Tampil Daftar Permohonan Pengajuan Kedatangan Kapal (PKK)

Aktor	: agen kapal
Kondisi awal	: daftar permohonan PKK kosong
Kondisi akhir	: daftar permohonan PKK tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan kedatangan kapal 2. Agen kapal memilih menu daftar permohonan PKK 3. Sistem mengambil data dari <i>database</i> 4. Sistem menampilkan daftar permohonan PKK
Skenario alternatif	: -

18. Skenario *Use Case* Melakukan Pencarian Daftar Permohonan Pengajuan Kedatangan Kapal

Tabel 3.20 Skenario Use Case Melakukan Pencarian Daftar Permohonan Pengajuan Kedatangan Kapal

Aktor	: agen kapal
Kondisi awal	: data pencarian daftar permohonan pkk kosong
Kondisi akhir	: data pencarian daftar permohonan pkk tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu daftar dokumen kepelabuhanan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal memasukkan kata kunci dokumen

	5. Sistem menampilkan data dokumen kepelabuhanan spesifik
Skenario alternatif	: -

19. Skenario *Use Case* Melihat Detail Daftar Permohonan Pengajuan Kedatangan Kapal

Tabel 3.21 Skenario Use Case Melihat Detail Daftar Permohonan Pengajuan Kedatangan Kapal

Aktor	: agen kapal
Kondisi awal	: detail daftar permohonan pkk kosong
Kondisi akhir	: detail daftar permohonan pkk tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan kedatangan kapal 2. Agen kapal memilih menu daftar permohonan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal menekan <i>detail</i> 5. Sistem mengambil data dari <i>database</i> 6. Sistem menampilkan detail data permohonan pkk
Skenario alternatif	: -

20. Skenario *Use Case Submit* Persetujuan Permohonan PKK (Pengajuan Kedatangan Kapal)

Tabel 3.22 Skenario Use Case Submit Persetujuan Permohonan PKK (Pengajuan Kedatangan Kapal)

Aktor	: agen kapal
Kondisi awal	: status <i>submit: registration</i> , status puk: <i>draft</i>
Kondisi akhir	: status <i>submit: approved</i> , status puk: <i>register</i>
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan kedatangan kapal 2. Agen kapal memilih menu daftar permohonan

	3. Sistem mengambil data dari <i>database</i> 4. Agen kapal menekan <i>submit</i> 5. Sistem menampilkan pesan persetujuan 6. Agen kapal menekan <i>continue</i> 7. Sistem melakukan operasi simpan ke <i>database</i> 8. Sistem menampilkan halaman daftar permohonan pkk
Skenario alternatif	: -

21. Skenario *Use Case* Menambah Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Tabel 3.23 Skenario Use Case Menambah Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Aktor	: agen kapal
Kondisi awal	: data pjk kosong
Kondisi akhir	: data pjk tersedia
Skenario	1. Agen kapal memilih menu pengajuan jasa kepelabuhanan 2. Agen kapal memilih menu <i>register</i> permohonan baru 3. Agen kapal mengisi data PJK 4. Agen kapal menekan simpan 5. Sistem melakukan operasi simpan ke <i>database</i> 6. Sistem menampilkan pesan data tersimpan 7. Sistem menampilkan halaman daftar permohonan PJK
Skenario alternatif	: -

22. Skenario *Use Case* Tampil Daftar Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Tabel 3.24 Skenario Use Case Tampil Daftar Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Aktor	: agen kapal
Kondisi awal	: daftar permohonan pjk kosong
Kondisi akhir	: daftar permohonan pjk tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan jasa kepelabuhanan 2. Agen kapal memilih menu daftar permohonan PJK 3. Sistem mengambil data dari <i>database</i> 4. Sistem menampilkan daftar permohonan PJK
Skenario alternatif	: -

23. Skenario *Use Case* Melakukan Pencarian Daftar Permohonan Pengajuan Jasa Kepelabuhanan

Tabel 3.25 Skenario Use Case Melakukan Pencarian Daftar Permohonan Pengajuan Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: data pencarian daftar permohonan pjk kosong
Kondisi akhir	: data pencarian daftar permohonan pjk tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan jasa kepelabuhanan 2. Agen kapal memilih menu daftar permohonan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal memasukkan kata kunci daftar permohonan 5. Sistem menampilkan data permohonan PJK secara spesifik
Skenario alternatif	: -

24. Skenario *Use Case* Melihat Detail Daftar Permohonan Pengajuan Jasa

Kepelabuhanan

Tabel 3.26 Skenario Use Case Melihat Detail Daftar Permohonan Pengajuan Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: detail daftar permohonan pjk kosong
Kondisi akhir	: detail daftar permohonan pjk tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan jasa kepelabuhanan 2. Agen kapal memilih menu daftar permohonan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal menekan <i>detail</i> 5. Sistem mengambil data dari <i>database</i> 6. Sistem menampilkan detail data permohonan pjk
Skenario alternatif	: -

25. Skenario *Use Case Submit* Persetujuan Permohonan Pengajuan Jasa Kepelabuhanan

Tabel 3.27 Skenario Use Case Submit Persetujuan Permohonan Pengajuan Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: status <i>submit: registration</i>
Kondisi akhir	: status <i>submit: approved</i>
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu pengajuan jasa kepelabuhanan 2. Agen kapal memilih menu daftar permohonan 3. Sistem mengambil data dari <i>database</i> 4. Agen kapal menekan <i>submit</i> 5. Sistem menampilkan pesan persetujuan 6. Agen kapal menekan <i>continue</i> 7. Sistem melakukan operasi simpan ke <i>database</i>

	8. Sistem menampilkan halaman daftar permohonan pjk
Skenario alternatif	: -

26. Skenario *Use Case* Mencetak SPK (Surat Perintah Kerja) Jasa Kepelabuhanan

Tabel 3.28 Skenario Use Case Mencetak SPK (Surat Perintah Kerja) Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: spk belum tersedia
Kondisi akhir	: spk tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu jasa pelabuhan 3. Agen kapal memilih data pkk 4. Sistem melakukan operasi pengambilan data dari <i>database</i> 5. Agen kapal menekan file <i>download</i> SPK 6. Sistem melakukan operasi pengambilan data dari <i>database</i> 7. Agen kapal menyimpan dokumen SPK 8. Agen kapal mencetak dokumen SPK
Skenario alternatif	: -

27. Skenario *Use Case* Mencetak Berita Acara Jasa Kepelabuhanan

Tabel 3.29 Skenario Use Case Mencetak Berita Acara Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: berita acara belum tersedia
Kondisi akhir	: berita acara tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa

	kepelabuhanan 2. Agen kapal memilih menu jasa pelabuhan 3. Agen kapal memilih data pkk 4. Sistem melakukan operasi pengambilan data dari <i>database</i> 5. Agen kapal menekan file <i>download</i> berita acara (ba) 6. Sistem melakukan operasi pengambilan data dari <i>database</i> 7. Agen kapal menyimpan dokumen berita acara jasa pelabuhan 8. Agen kapal mencetak dokumen berita acara jasa pelabuhan
Skenario alternatif	: -

28. Skenario *Use Case* Mencetak Bukti *Hold* Dana Jasa Kepelabuhanan

Tabel 3.30 Skenario Use Case Mencetak Bukti *Hold* Dana Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: bukti <i>hold</i> dana belum tersedia
Kondisi akhir	: bukti <i>hold</i> dana tersedia
Skenario	1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu jasa pelabuhan 3. Agen kapal memilih data pkk 4. Sistem melakukan operasi pengambilan data dari <i>database</i> 5. Agen kapal menekan file <i>download</i> bukti hold dana 6. Sistem melakukan operasi pengambilan data dari <i>database</i>

	7. Agen kapal menyimpan dokumen bukti <i>hold</i> dana jasa pelabuhan 8. Agen kapal mencetak dokumen bukti <i>hold</i> dana jasa pelabuhan
Skenario alternatif	: -

29. Skenario *Use Case* Mencetak Nota Lunas Jasa Kepelabuhanan

Tabel 3.31 Skenario Use Case Mencetak Nota Lunas Jasa Kepelabuhanan

Aktor	: agen kapal
Kondisi awal	: nota lunas belum tersedia
Kondisi akhir	: nota lunas tersedia
Skenario	1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu jasa pelabuhan 3. Agen kapal memilih data pkk 4. Sistem melakukan operasi pengambilan data dari <i>database</i> 5. Agen kapal menekan file <i>download</i> nota lunas 6. Sistem melakukan operasi pengambilan data dari <i>database</i> 7. Agen kapal menyimpan dokumen nota lunas jasa pelabuhan 8. Agen kapal mencetak dokumen nota lunas jasa pelabuhan
Skenario alternatif	: -

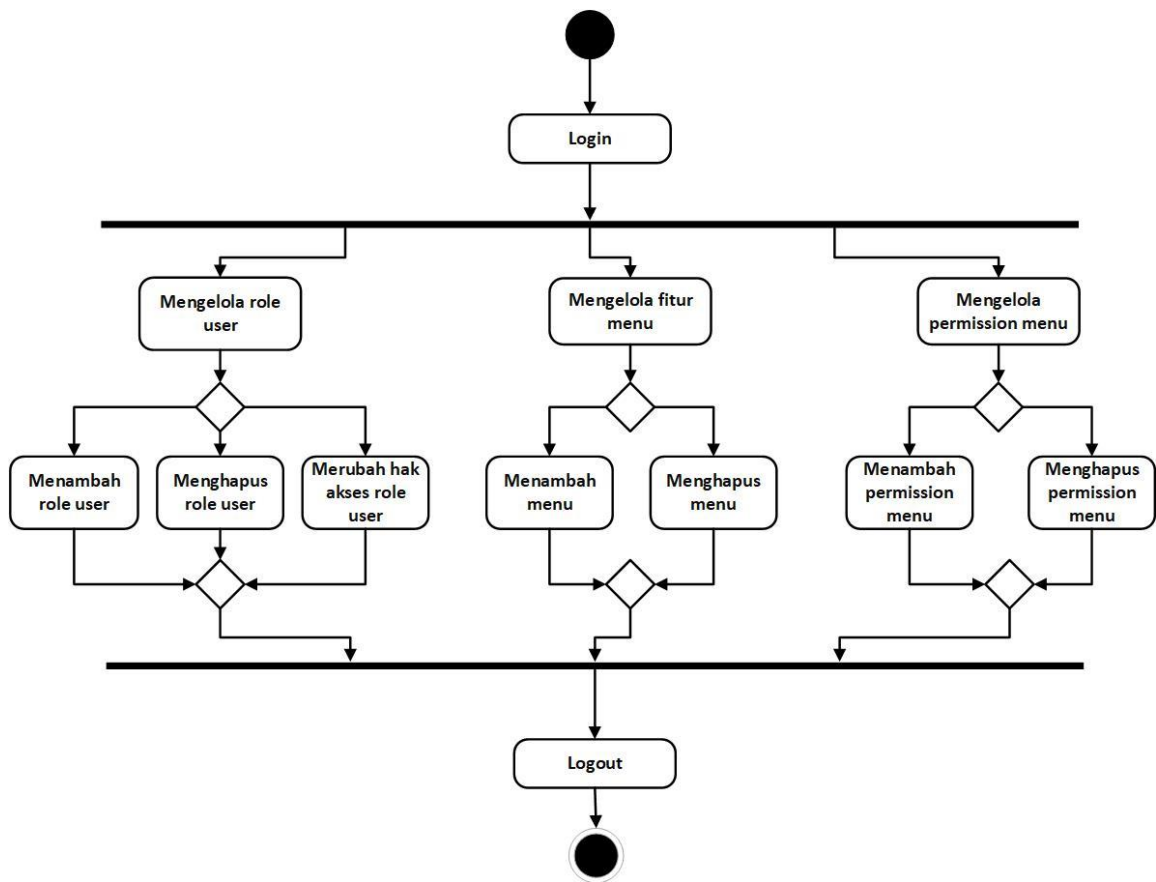
30. Skenario *Use Case* Mencetak Log Transaksional Jasa Kepelabuhanan
Host to Host

**Tabel 3.32 Skenario Use Case Mencetak Log Transaksional
Jasa Kepelabuhanan Host to Host**

Aktor	: agen kapal
Kondisi awal	: <i>log transaksional host to host</i> belum tersedia
Kondisi akhir	: <i>log transaksional host to host</i> tersedia
Skenario	<ol style="list-style-type: none"> 1. Agen kapal memilih menu manajemen jasa kepelabuhanan 2. Agen kapal memilih menu jasa pelabuhan 3. Agen kapal memilih data pkk 4. Sistem melakukan operasi pengambilan data dari <i>database</i> 5. Agen kapal menekan file <i>download cetak log host to host</i> 6. Sistem melakukan operasi pengambilan data dari <i>database</i> 7. Agen kapal menyimpan dokumen <i>log host to host</i> jasa pelabuhan 8. Agen kapal mencetak dokumen <i>log host to host</i> jasa pelabuhan
Skenario alternatif	: -

3.3.3 Activity Diagram

1. Activity Diagram Admin



Gambar 3.4 Activity Diagram Admin

Pada gambar 7.4 menggambarkan aktivitas kerja sebuah sistem yang dilakukan oleh admin. Admin dapat melakukan *login*, ketika berhasil maka akan masuk menampilkan menu. Setelah masuk ke menu dapat melakukan pengelolaan yang tersedia di sistem. Jika telah selesai menggunakan maka dilakukan *logout*.

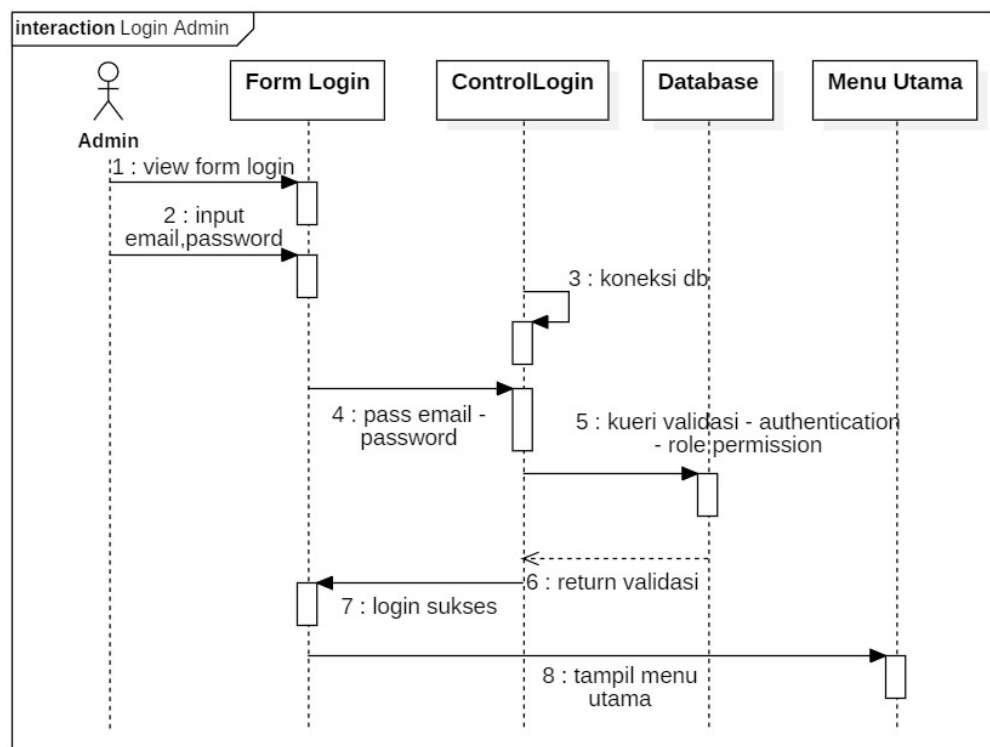
2. Activity Diagram Agen Kapal

Gambar 3.5 Activity Diagram Agen Kapal

Pada gambar 7.5 menggambarkan aktivitas kerja sebuah sistem yang dilakukan oleh agen kapal. Agen kapal dapat melakukan *login*, ketika berhasil maka akan masuk menampilkan menu. Setelah masuk ke menu dapat melakukan pengelolaan yang tersedia di sistem. Jika telah selesai menggunakan maka dilakukan *logout*.

3.3.4 Sequence Diagram

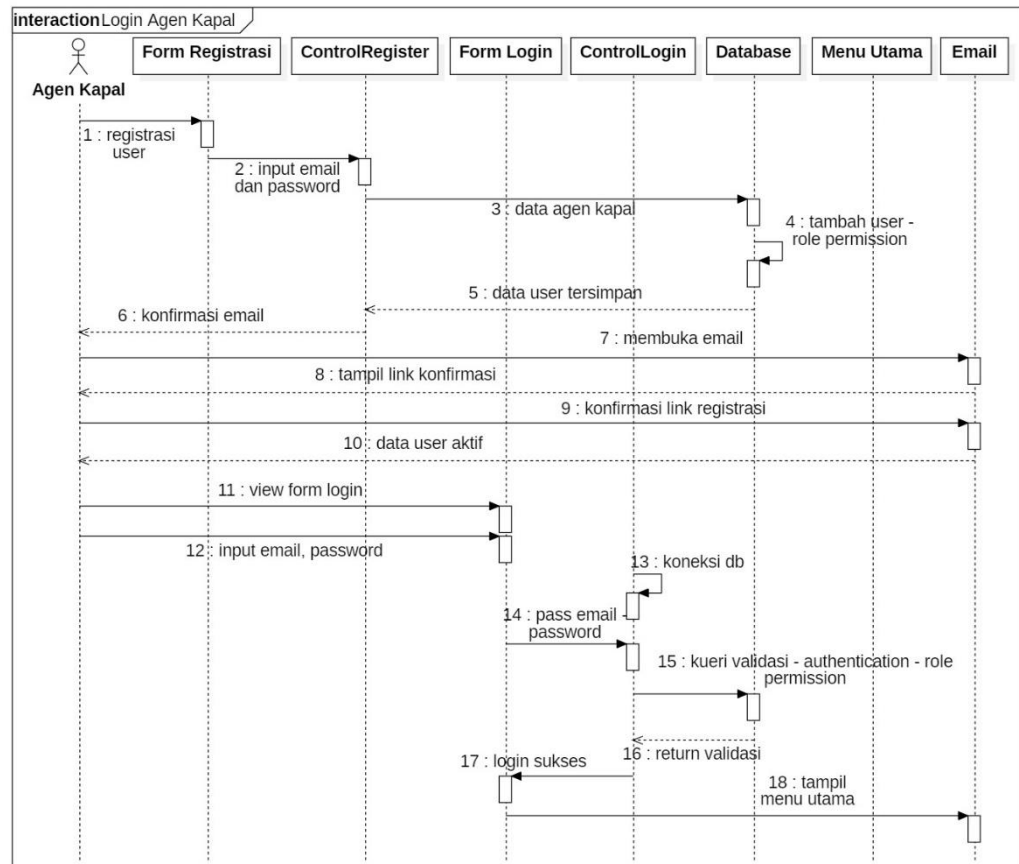
1. Sequence Diagram Login Admin



Gambar 3.6 Sequence Diagram Login Admin

Pada gambar 7.6 menunjukkan proses *login* admin sampai tampil pesan kesalahan jika gagal dan masuk ke aplikasi jika berhasil *login*.

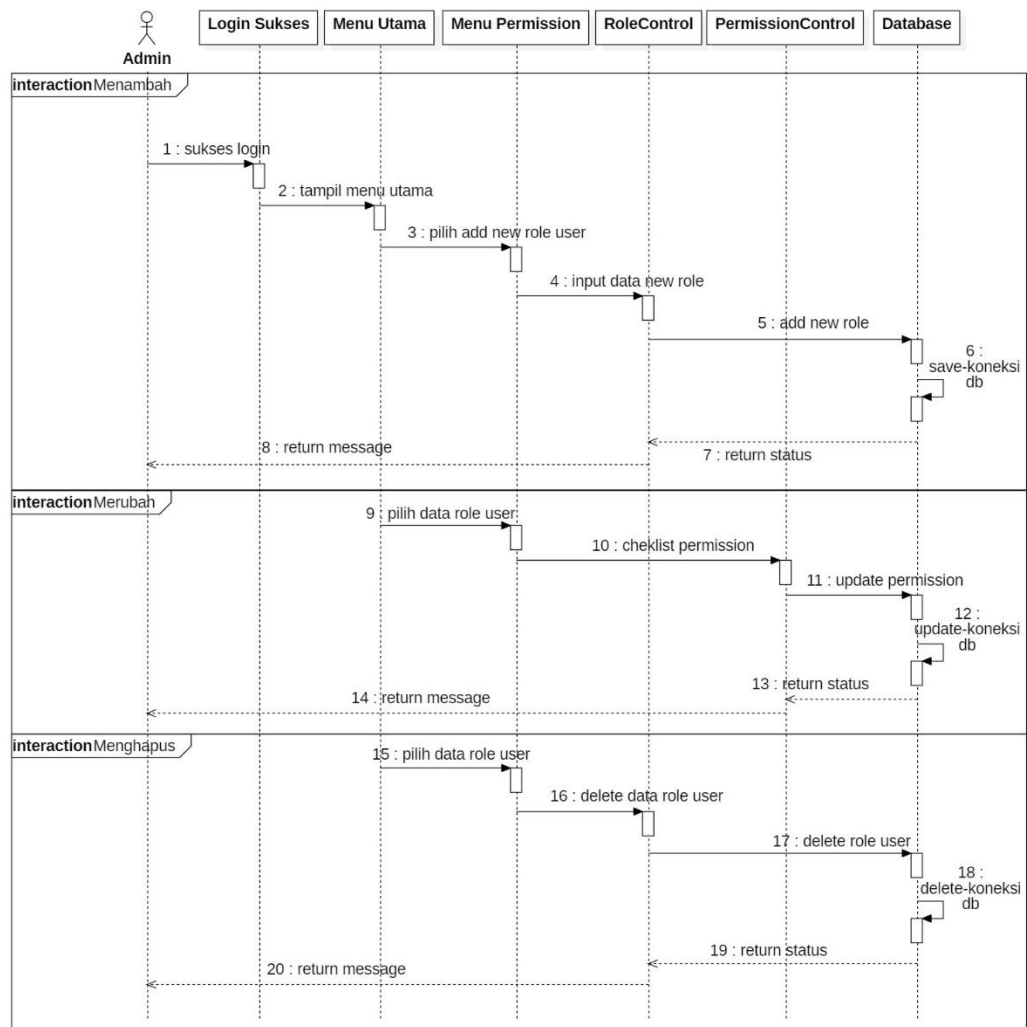
2. Sequence Diagram Login Agen Kapal



Gambar 3.7 Sequence Diagram Login Agen Kapal

Pada gambar 7.7 menunjukkan proses *login* agen kapal dimulai dari *registrasi* dan konfirmasi email terlebih dahulu sampai tampil pesan kesalahan jika gagal dan masuk ke aplikasi jika berhasil *login*.

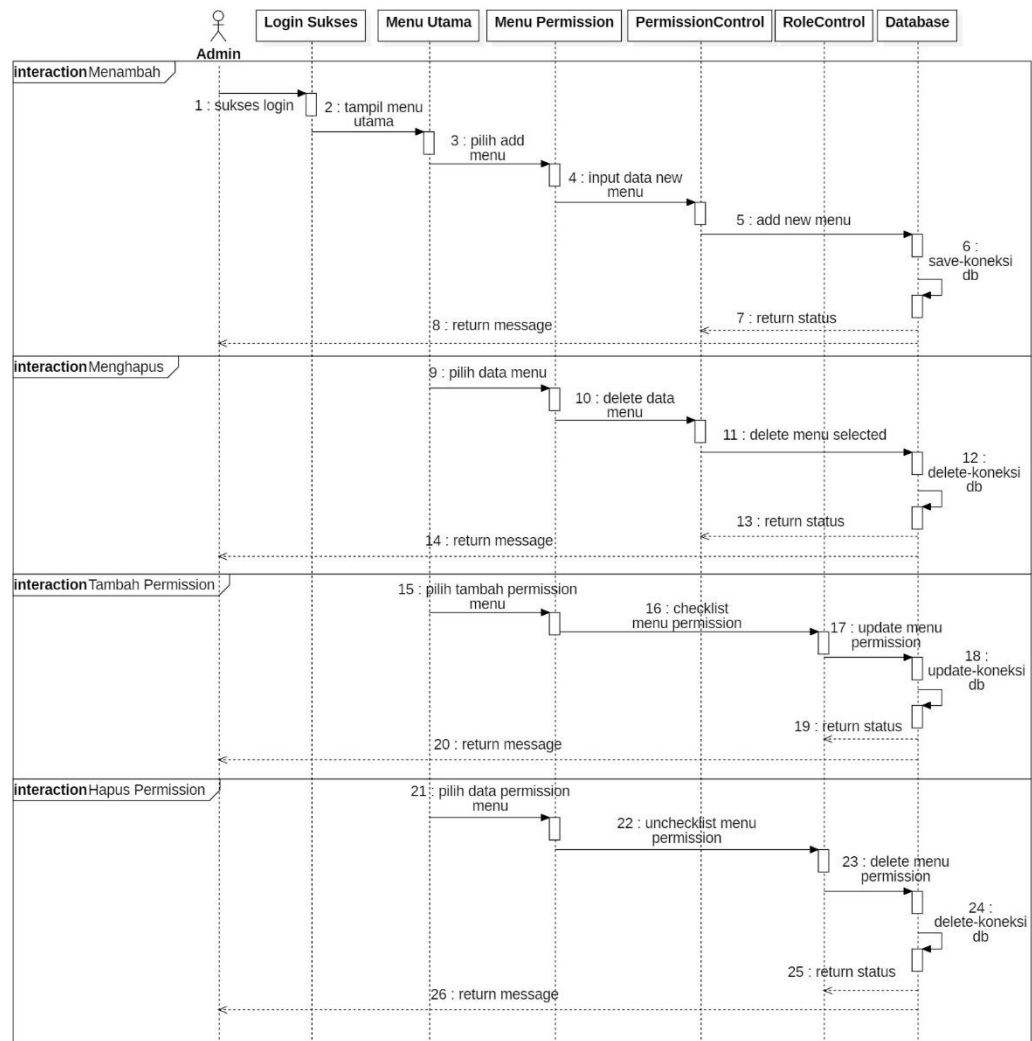
3. Sequence Diagram Manajemen Role User



Gambar 3.8 Sequence Diagram Manajemen Role User

Pada gambar 7.8 menunjukkan proses manajemen *role user* seperti: proses menambah *role user*, merubah hak akses *role user*, dan menghapus data *role user*. Pada proses nya admin harus berhasil masuk ke dalam sistem agar dapat melakukan manajemen *role user*. Aktor memasukkan data yang dikelola dan jika valid maka data dapat disimpan, dirubah bahkan dihapus di *database*.

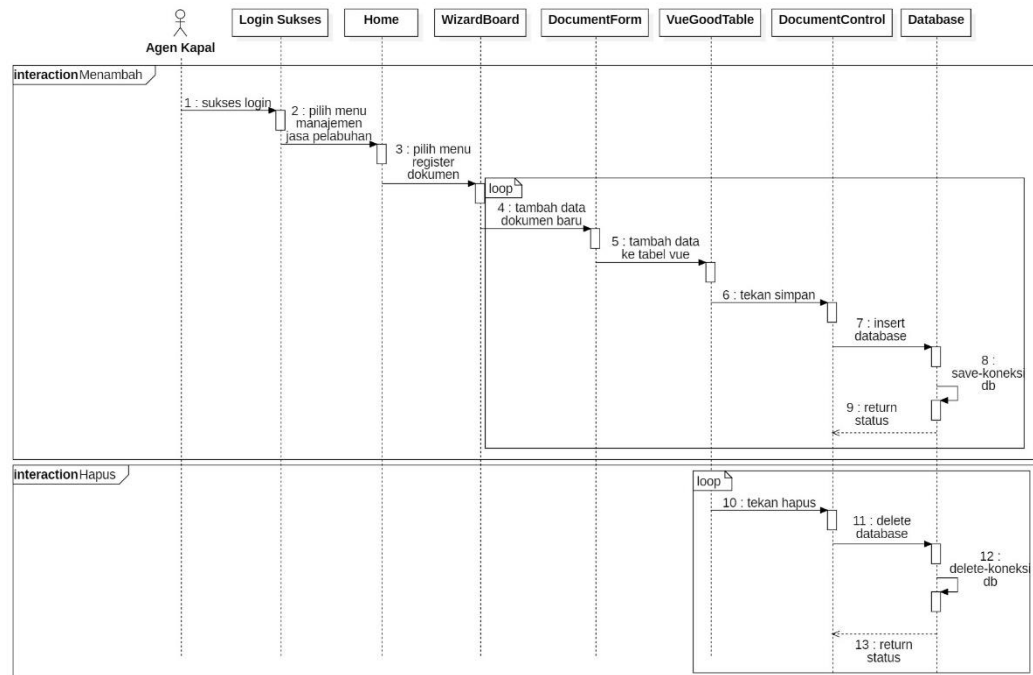
4. Sequence Diagram Manajemen Fitur Menu



Gambar 3.9 Sequence Diagram Manajemen Fitur Menu

Pada gambar 7.9 menunjukkan proses manajemen fitur menu oleh admin seperti: proses menambah fitur menu, menghapus fitur menu, menambah *permission* menu, dan menghapus *permission* menu. Pada proses nya admin harus berhasil masuk ke dalam sistem agar dapat melakukan manajemen fitur menu. Aktor memasukkan data yang dikelola dan jika valid, data dapat ditambah, dirubah, dan dihapus di *database*.

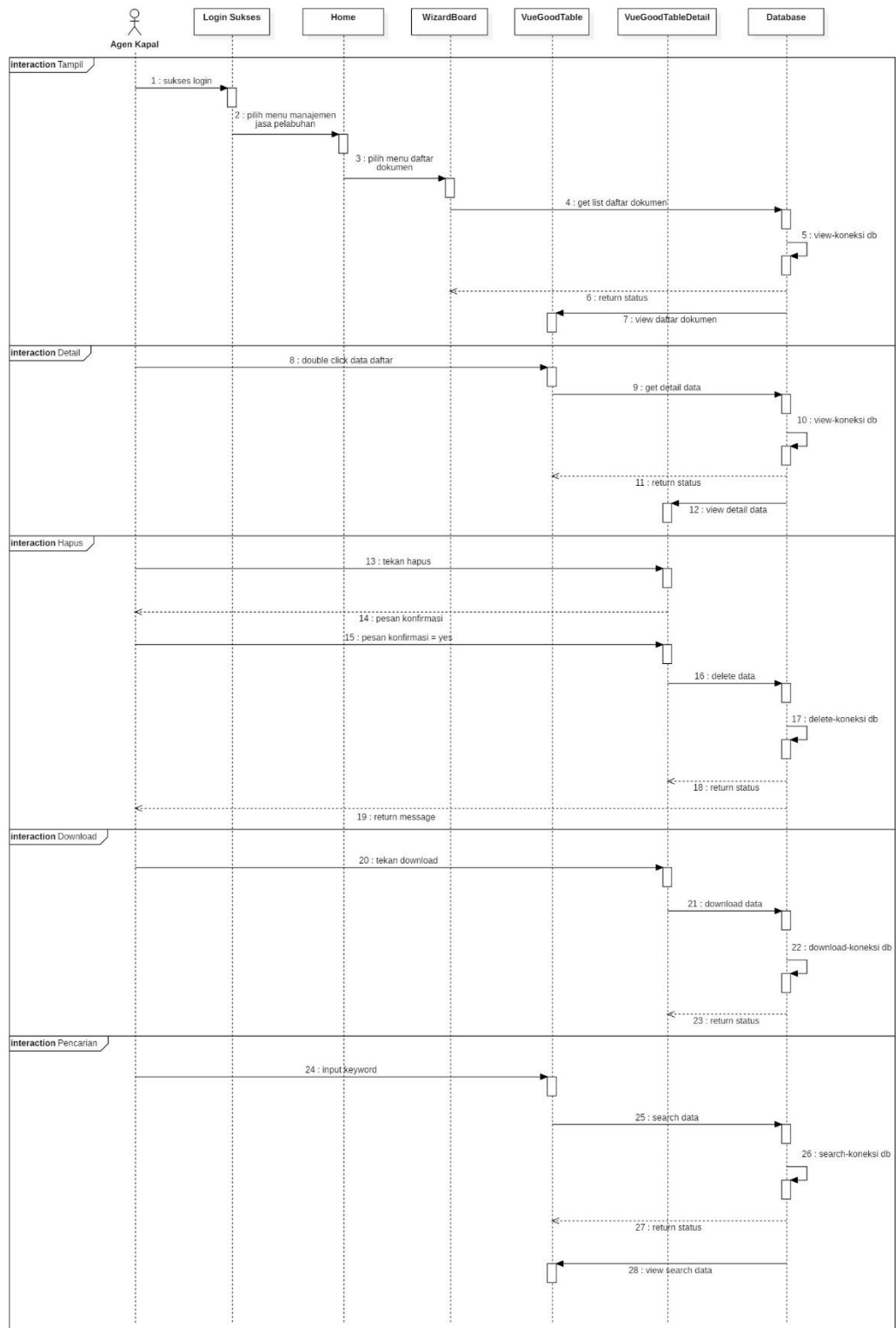
5. Sequence Diagram Manajemen Dokumen Kepelabuhanan (Menambah dan Menghapus)



Gambar 3.10 Sequence Diagram Manajemen Dokumen Kepelabuhanan (Menambah dan Menghapus)

Pada gambar 7.10 menunjukkan proses manajemen dokumen kepelabuhanan seperti menambah dan menghapus dokumen yang di-*register*. Pada proses nya agen kapal harus berhasil masuk ke dalam sistem agar dapat melakukan manajemen dokumen kepelabuhanan. Aktor memasukkan data yang dikelola dan jika valid, data dapat ditambah dan dihapus di *database*.

6. *Sequence Diagram Monitoring* Dokumen Kepelabuhanan (Tampil daftar dokumen, detail daftar dokumen, hapus daftar dokumen, *download* dokumen, dan pencarian Dokumen)

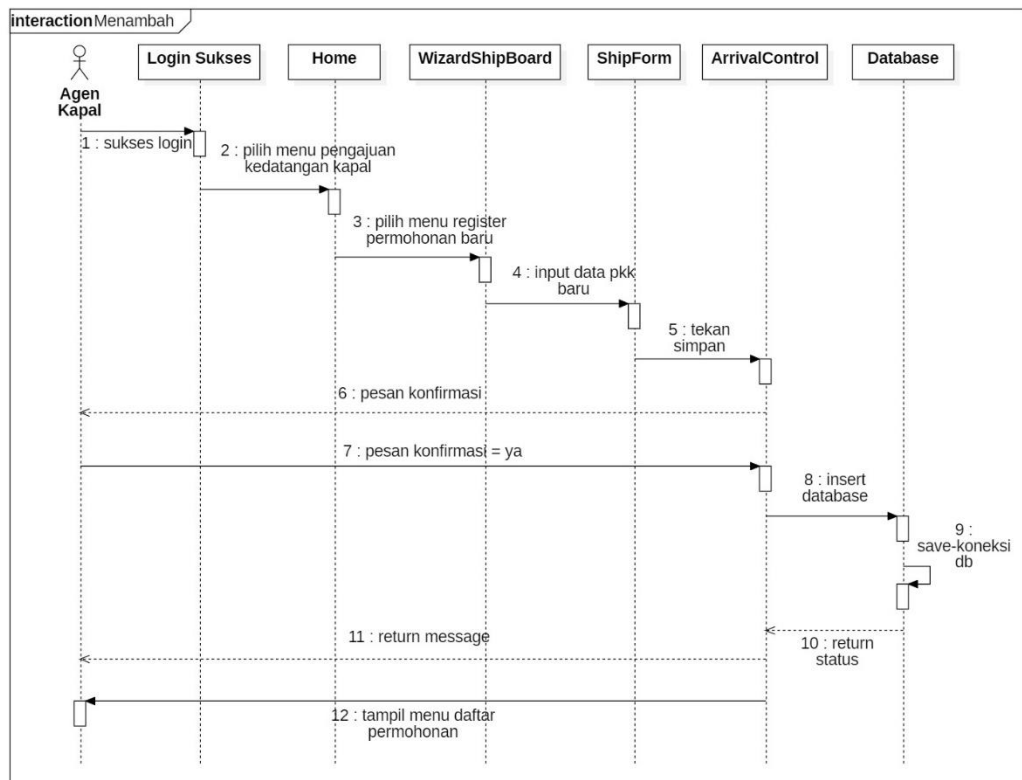


Gambar 3.11 Sequence Diagram *Monitoring* Dokumen Kepelabuhanan (Tampil daftar dokumen, detail daftar

dokumen, hapus daftar dokumen, *download* dokumen, dan pencarian dokumen)

Pada gambar 7.11 menunjukkan proses *monitoring* dokumen kepelabuhanan seperti: tampil daftar dokumen yang sudah di-*register*, melihat detail dari data dokumen yang sudah di-*register*, menghapus daftar dokumen, mengunduh daftar dokumen, dan melakukan pencarian data dari daftar dokumen yang sudah di-*register*. Pada proses nya agen kapal harus berhasil masuk ke dalam sistem agar dapat melakukan manajemen dokumen kepelabuhanan. Aktor memasukkan data yang dikelola dan jika valid data dapat ditambah, diunduh, dicari, dan dihapus di *database*.

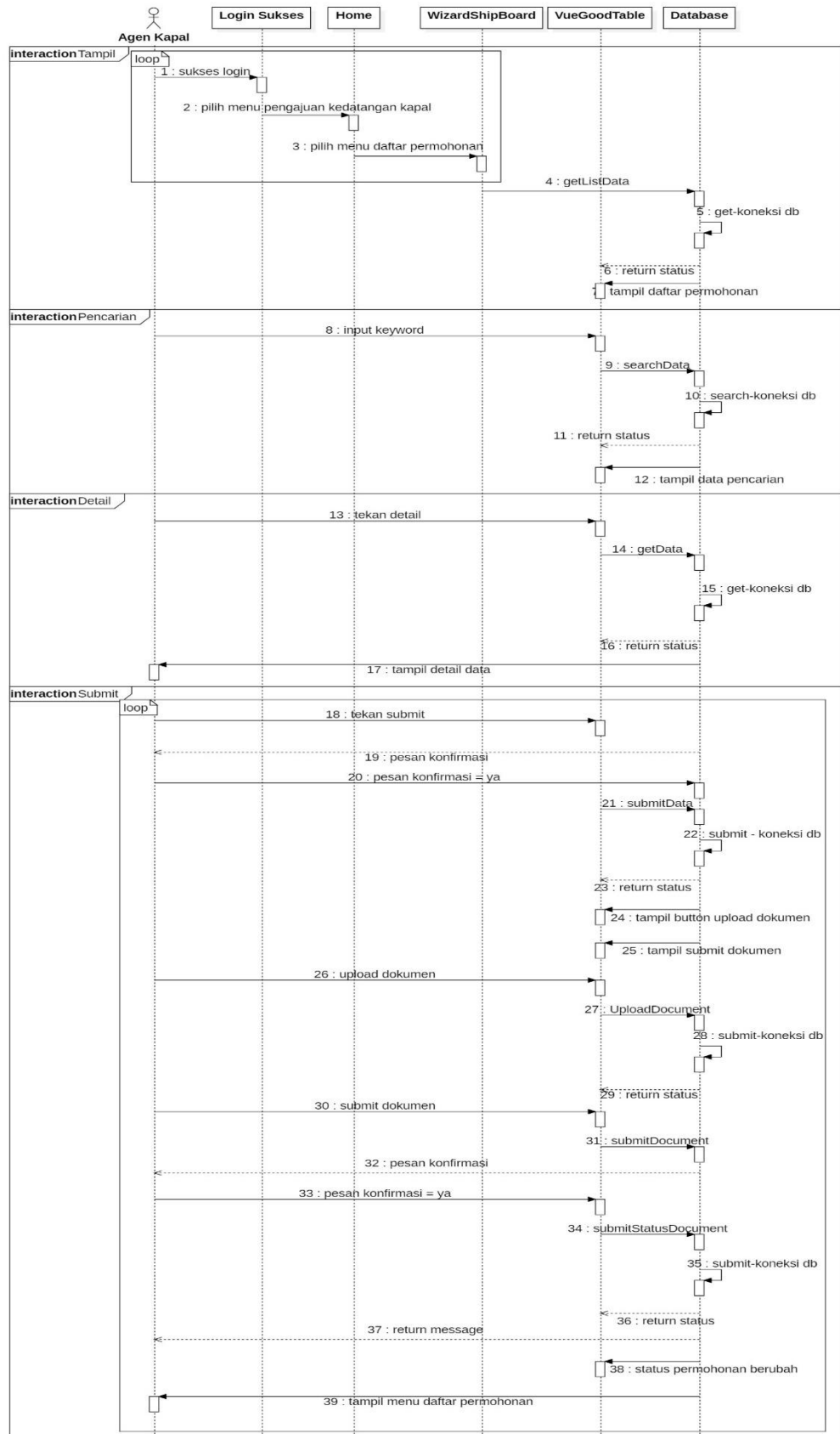
7. *Sequence Diagram* Menambah Permohonan Pengajuan Kedatangan Kapal



3.12 Sequence Diagram Menambah Permohonan Pengajuan Kedatangan Kapal

Pada gambar 7.12 menunjukkan proses menambah permohonan pengajuan kedatangan kapal. Pada proses nya agen kapal harus berhasil masuk ke dalam sistem agar dapat melakukan penambahan pengajuan kedatangan kapal. Aktor memasukkan data yang dikelola dan jika valid data dapat ditambah ke *database*.

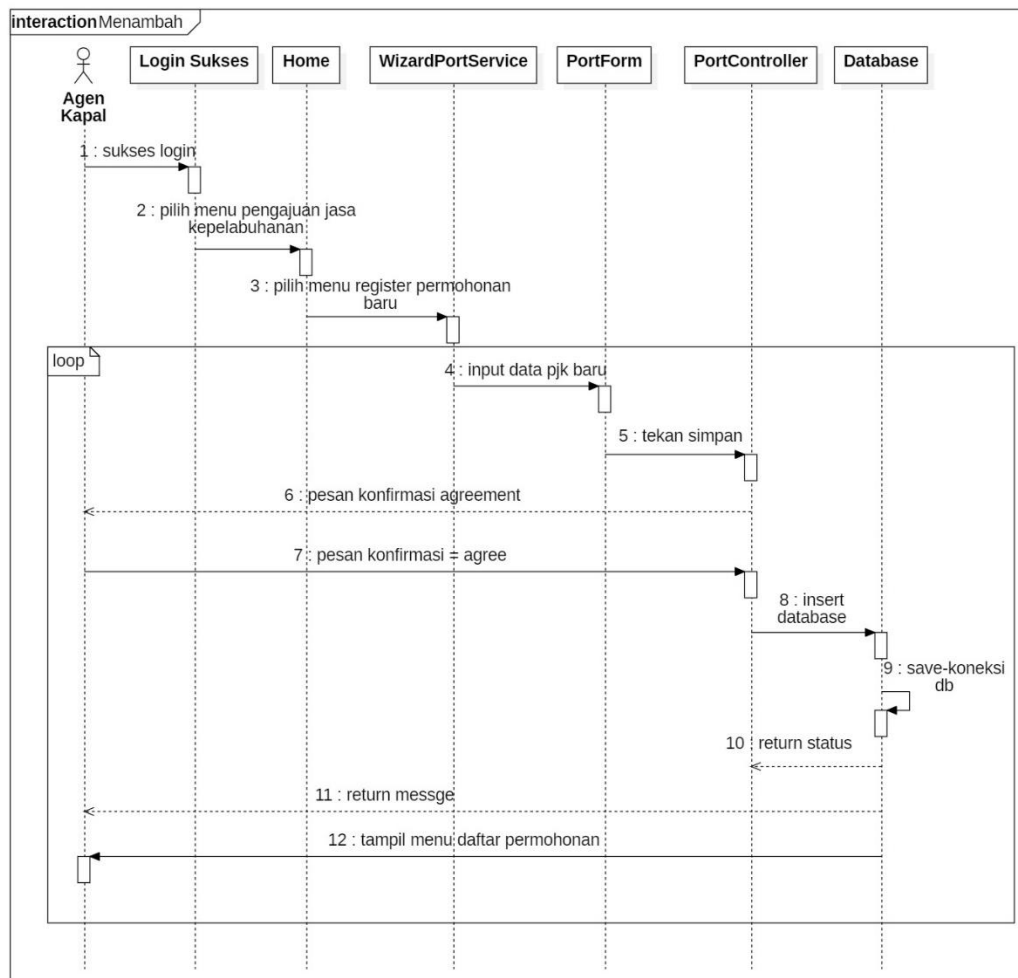
8. *Sequence Diagram Monitoring* Pengajuan Kedatangan Kapal/PKK
(Tampil daftar permohonan, pencarian permohonan, detail permohonan, dan *submit* dokumen permohonan)



3.13 Sequence Diagram Monitoring Pengajuan Kedatangan Kapal/PKK (Tampil daftar permohonan, pencarian permohonan, detail permohonan, dan *submit* dokumen permohonan)

Pada gambar 7.13 menunjukkan proses *monitoring* pengajuan kedatangan kapal seperti: tampil daftar permohonan, melihat detail dari permohonan yang sudah di-*register*, melakukan *submit* dari dokumen pkk, dan melakukan pencarian data dari daftar permohonan pkk yang sudah di-*register*. Pada proses nya agen kapal harus berhasil masuk ke dalam sistem agar dapat melakukan monitoring pengajuan kedatangan kapal. Aktor memasukkan data yang dikelola dan jika valid data dapat ditambah dan dicari di *database*.

9. *Sequence Diagram* Menambah Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

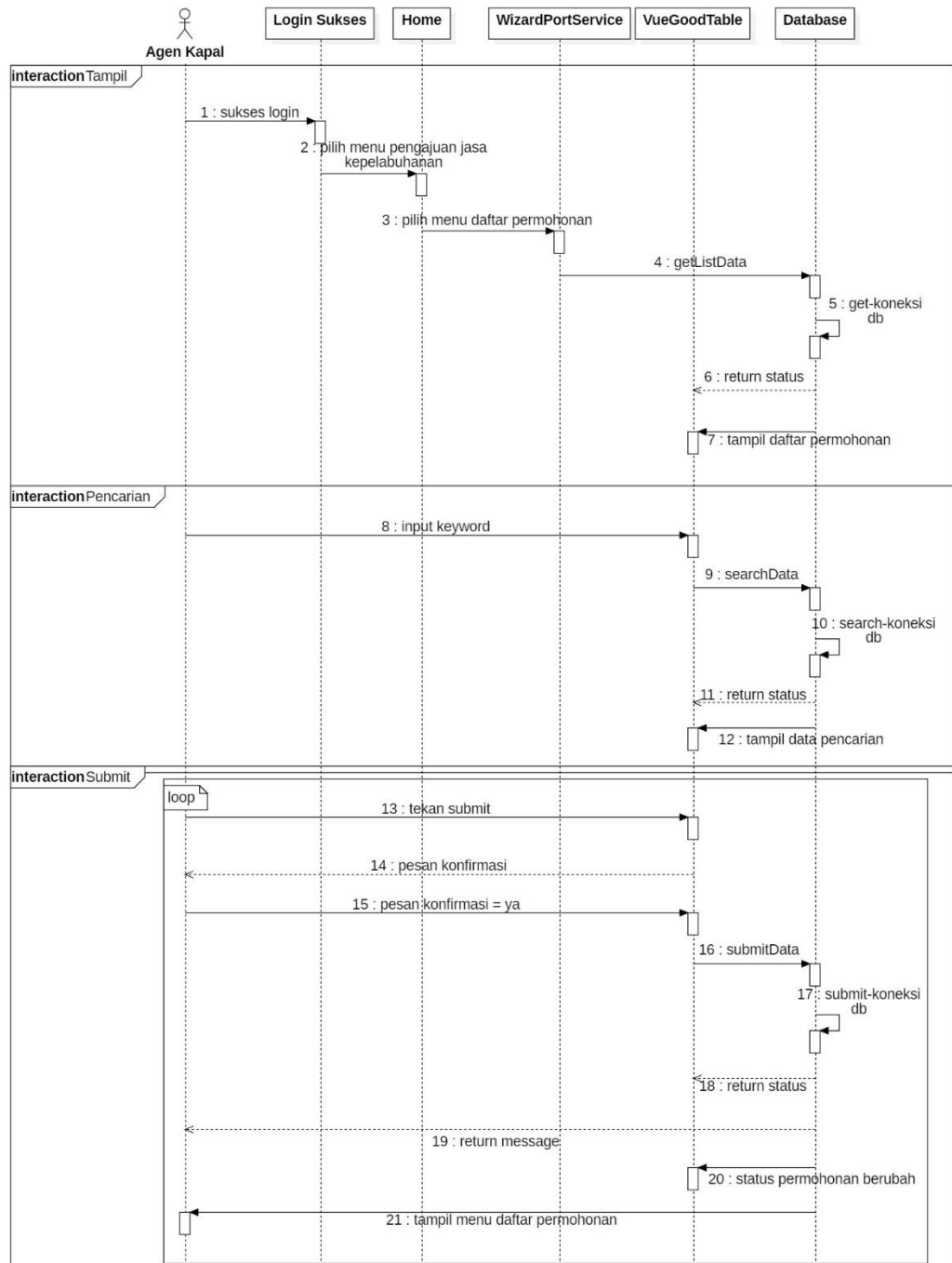


Gambar 3.14 Sequence Diagram Menambah Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Pada gambar 7.14 menunjukkan proses menambah permohonan pengajuan jasa pelabuhan. Pada proses nya agen kapal harus berhasil masuk ke dalam sistem agar dapat melakukan penambahan pengajuan jasa kepelabuhanan. Aktor memasukkan data yang dikelola dan jika valid data dapat ditambah ke *database*.

10. Sequence Diagram Monitoring Pengajuan Jasa

Kepelabuhanan/PJK (Tampil daftar permohonan, pencarian permohonan, dan *submit* dokumen permohonan)

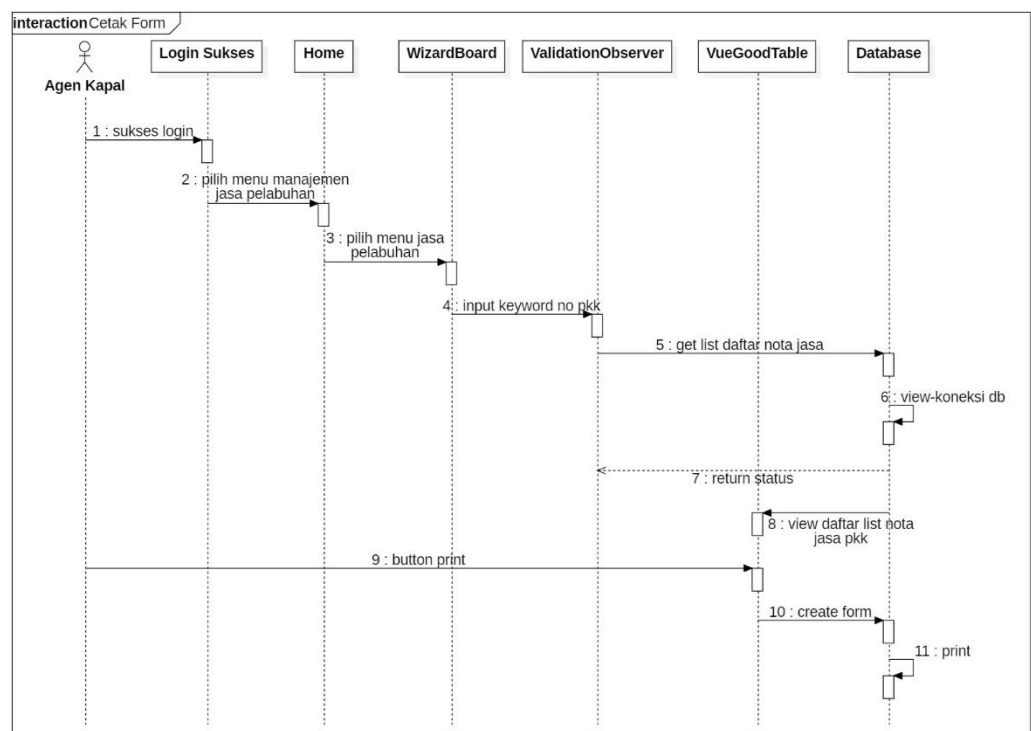


Gambar 3.15 Sequence Diagram *Monitoring* Pengajuan Jasa Kepelabuhanan/PJK (Tampil daftar permohonan, pencarian permohonan, dan submit dokumen permohonan)

Pada gambar 7.15 menunjukkan proses *monitoring* pengajuan jasa kepelabuhanan seperti: tampil daftar permohonan, melakukan *submit* dari permohonan pjk, dan melakukan pencarian data dari daftar

permohonan pjk yang sudah di-*register*. Pada proses nya agen kapal harus berhasil masuk ke dalam sistem agar dapat melakukan monitoring pengajuan jasa kepelabuhanan. Aktor memasukkan data yang dikelola dan jika valid data dapat ditambah dan dicari di *database*.

11. *Sequence Diagram* Mencetak Form (SPK/Surat Perintah Kerja, Berita Acara, Bukti *Hold* Dana Jasa, Nota Lunas, dan Log Transaksional *Host to Host*) Jasa Kepelabuhanan



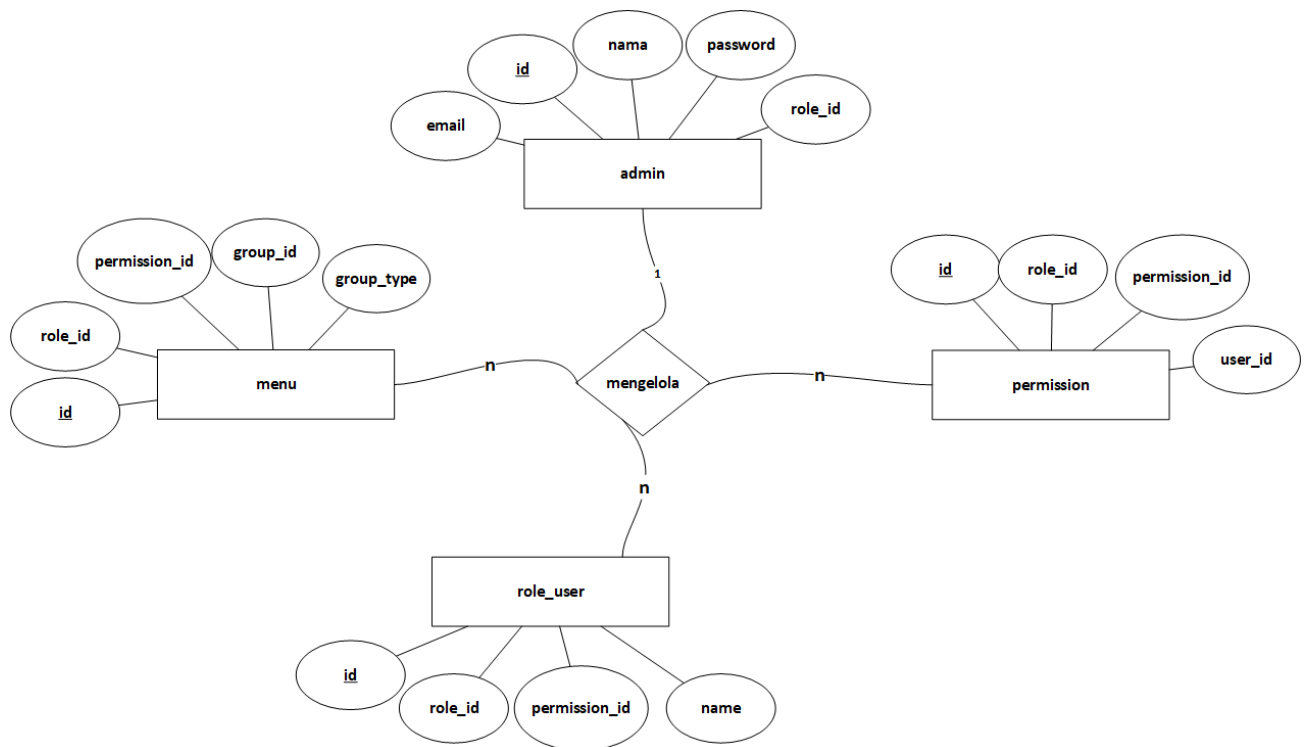
Gambar 3.16 Sequence Diagram Mencetak Form (SPK/Surat Perintah Kerja, Berita Acara, Bukti *Hold* Dana Jasa, Nota Lunas, dan Log Transaksional *Host to Host*) Jasa Kepelabuhanan

Pada gambar 7.16 menunjukkan proses untuk membuat form seperti: form SPK (Surat Perintah Kerja), form berita acara jasa kepelabuhanan, form bukti *hold* dana jasa kepelabuhanan, form nota lunas jasa kepelabuhanan, dan log transaksional jasa kepelabuhanan *host to host*. Data yang sudah tersimpan di *database* akan ditampilkan sebagai form yang dapat dicetak.

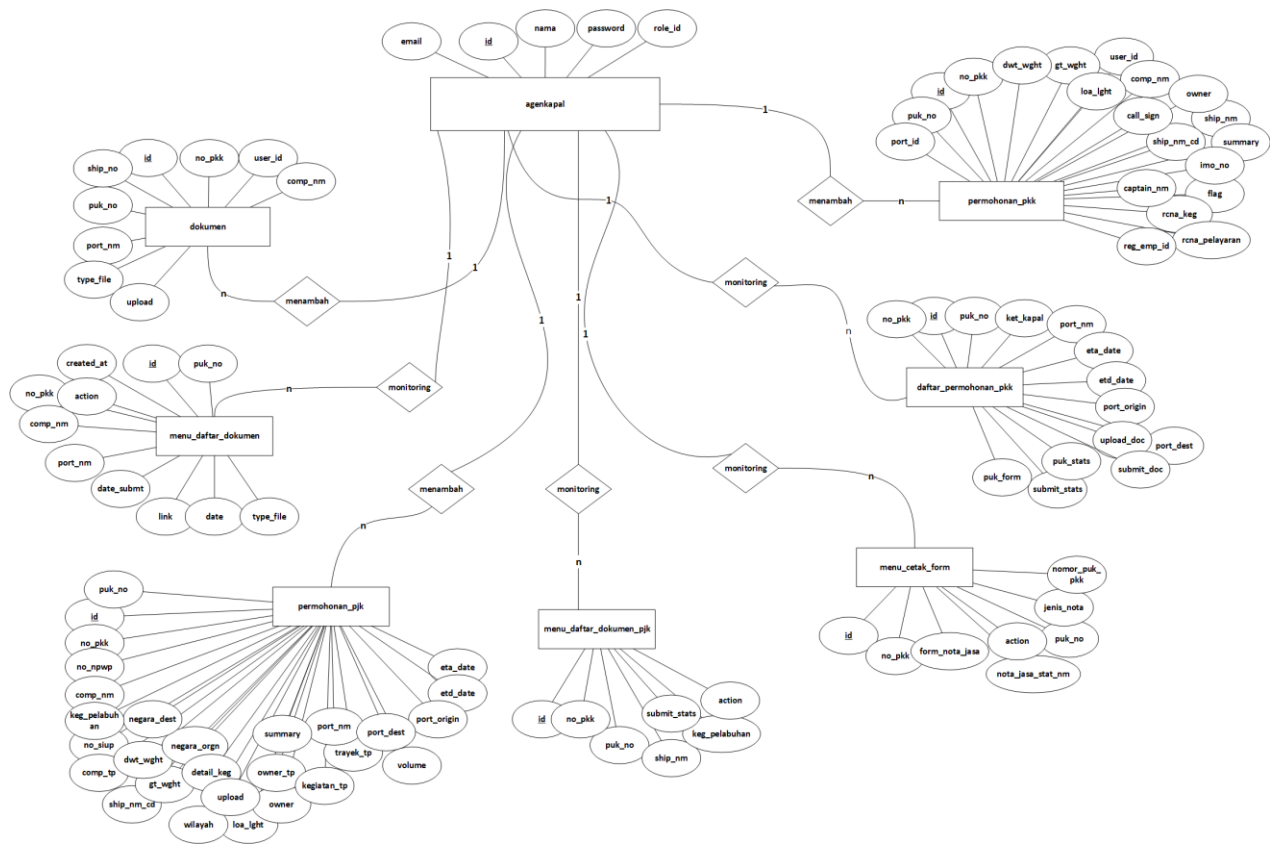
Pada Gambar 7.6 diatas merupakan rancangan form yang dibuat dalam bentuk *class diagram*. Adapun interaksi antar kelas atau objek telah dibahas pada *sequence diagram*.

3.3.6 EERD (*Enhanced Entity Relational Diagram*)

EERD merupakan model perancangan hubungan antar entitas (tabel) dari sebuah basis data.



Gambar 3.18 Entity Relationship Diagram Admin



Gambar 3.19 Entity Relationship Diagram Agen Kapal

Pada Gambar 7.18 merupakan desain *ERD* untuk *user* Admin, sedangkan Gambar 7.19 merupakan desain *ERD* untuk *user* Agen Kapal. Desain *ERD* menjelaskan bahwa *ERD* memiliki berbagai macam entitas dalam membangun aplikasi B-SIMS.

3.3.7 Rancangan Antarmuka

1. Form Login *Admin* dan Agen Kapal

The form is divided into two main sections. The left section contains a box for 'LOGO/JUDUL' at the top and a larger box for 'DESCRIPTION APP' below it. The right section contains input fields for 'EMAIL' and 'PASSWORD', followed by a link for 'LUPA KATA SANDI'. Below these is a 'SIGN IN' button and a link for 'BUAT AKUN BARU'.

Gambar 3.20 Form Login Admin dan Agen Kapal

Pada Gambar 3.20 merupakan antarmuka *login*. Untuk dapat *login* harus memasukkan *email* dan *password* yang sudah terdaftar.

2. Menu Utama Admin

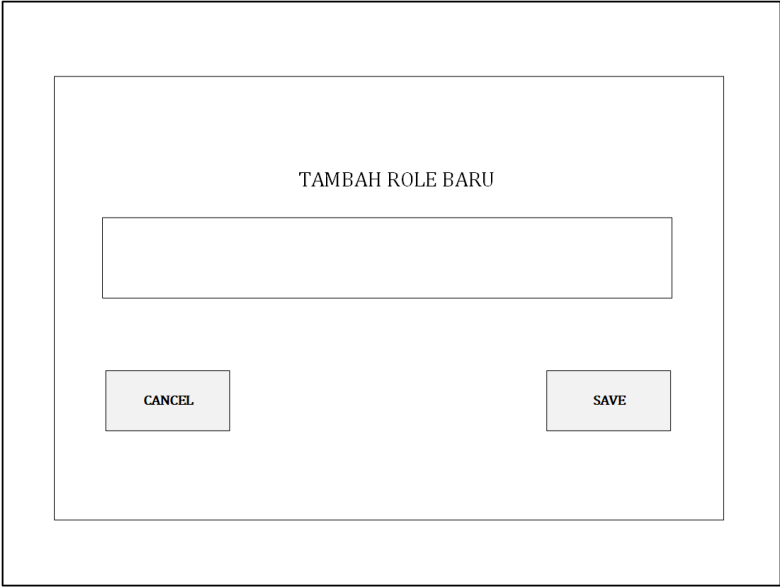
The main admin menu layout features a 'LOGO' box at the top left. Below it, there are two tabs: 'USERS' and 'PERMISSION'. The main content area displays a grid of eight user cards, labeled 'USER 1' through 'USER 8', arranged in two rows of four.

Gambar 3.21 Menu Utama Admin

Pada Gambar 3.21 merupakan antarmuka menu utama admin. Pada menu ini menampilkan menu *users* dan menu *permission*. Untuk menu *users* berisi daftar dari pengguna aplikasi yang sudah terdaftar, dan untuk menu

permission berisi fitur menu seperti: tambah *role user* baru, tambah menu baru, tambah *permission role user*, dan ubah hak akses *permission* menu *role user*.

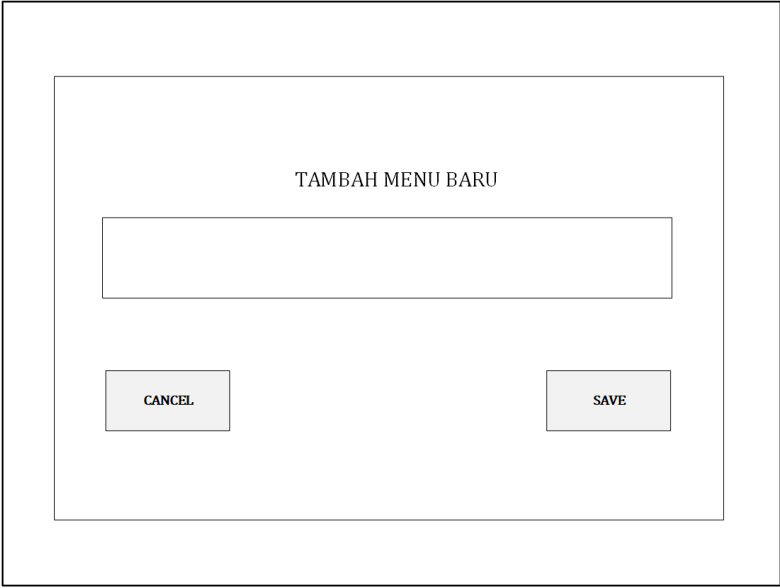
3. Form Tambah Role User Baru

The image shows a web form titled "TAMBAH ROLE BARU" (Add New Role). It features a single text input field for entering the role name. Below the input field, there are two buttons: "CANCEL" on the left and "SAVE" on the right. The entire form is enclosed in a rectangular border.

Gambar 3.22 Form Tambah Role User Baru

Pada Gambar 3.22 merupakan antarmuka form tambah *role user* baru yang hanya dapat diakses oleh admin. *Role user* ini adalah para pengguna yang dapat masuk ke aplikasi B-SIMS.

4. Form Tambah Menu



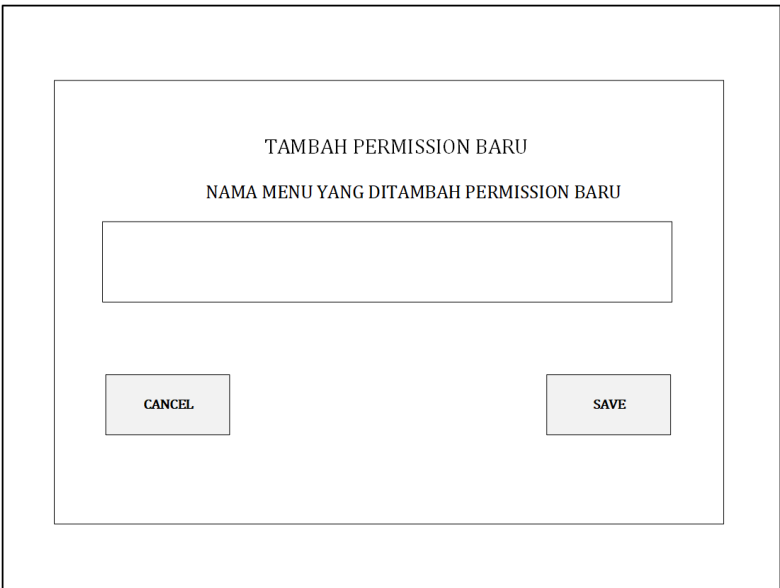
TAMBAH MENU BARU

CANCEL SAVE

Gambar 3.23 Form Tambah Menu

Pada Gambar 3.23 merupakan antarmuka form tambah menu baru yang hanya dapat diakses oleh admin. Menu ini berisi menu yang terdaftar di aplikasi B-SIMS yang dapat diakses oleh pengguna berdasarkan *permission* yang ada.

5. Form Tambah *Permission* Menu Role User



TAMBAH PERMISSION BARU

NAMA MENU YANG DITAMBAH PERMISSION BARU

CANCEL SAVE

Gambar 3.24 Form Tambah *Permission* Menu Role User

Pada Gambar 3.24 merupakan antarmuka form tambah *permission* menu

role *user* yang hanya dapat diakses oleh admin. Pada form ini admin menambahkan jenis *permission* baru untuk daftar menu yang sudah terdaftar di aplikasi B-SIMS.

6. Menu Ubah Hak Akses Menu *Role User*

Gambar 3.25 Menu Ubah Hak Akses Menu *Role User*

Pada Gambar 3.25 merupakan antarmuka menu ubah hak akses *role user* yang hanya diakses oleh admin. Pada menu ini admin melakukan *checklist* pada pilihan *permission* yang ada di setiap menu yang sudah terdaftar di aplikasi B-SIMS.

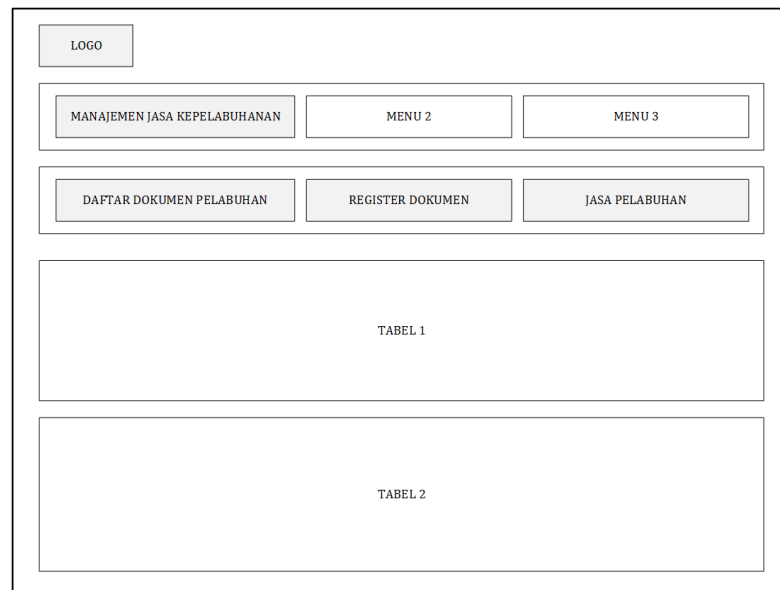
7. Menu Utama Agen Kapal



Gambar 3.26 Menu Utama Agen Kapal

Pada Gambar 3.26 merupakan antarmuka menu utama agen kapal Ketika berhasil login ke dalam aplikasi. Pada menu ini terdapat tiga menu utama yaitu: menu Manajemen Jasa Kepelabuhanan, menu Pengajuan Kedatangan Kapal (PKK), dan menu Pengajuan Jasa Kepelabuhanan (PJK). Masing-masing dari menu utama ini memiliki sub menu lainnya.

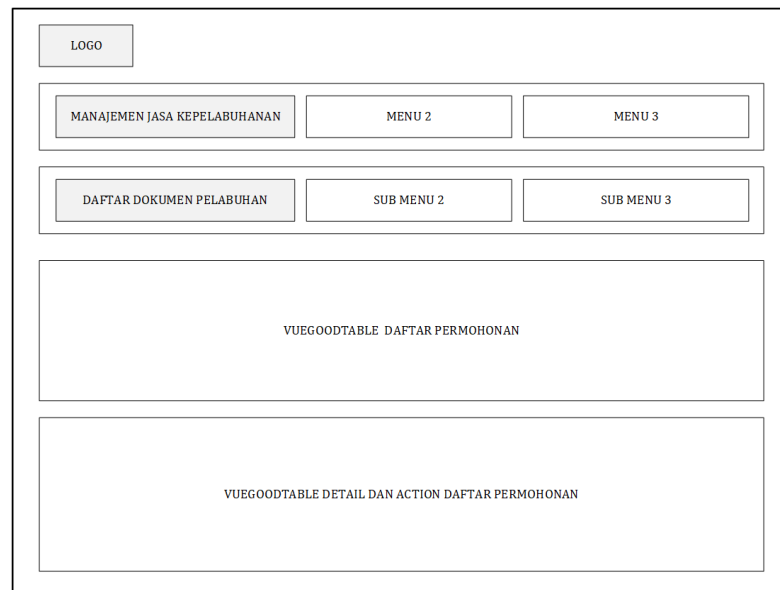
8. Menu Utama Manajemen Jasa Kepelabuhanan



**Gambar 3.27 Menu Utama Manajemen Jasa
Kepelabuhanan**

Pada Gambar 3.27 merupakan antarmuka menu utama manajemen jasa kepelabuhanan yang diakses oleh agen kapal. Pada menu ini memiliki tiga sub menu yaitu: sub menu Daftar Dokumen Pelabuhan, sub menu Register Dokumen, dan sub menu Jasa Pelabuhan. Untuk penjelasan dari sub menu akan dijabarkan pada gambar selanjutnya.

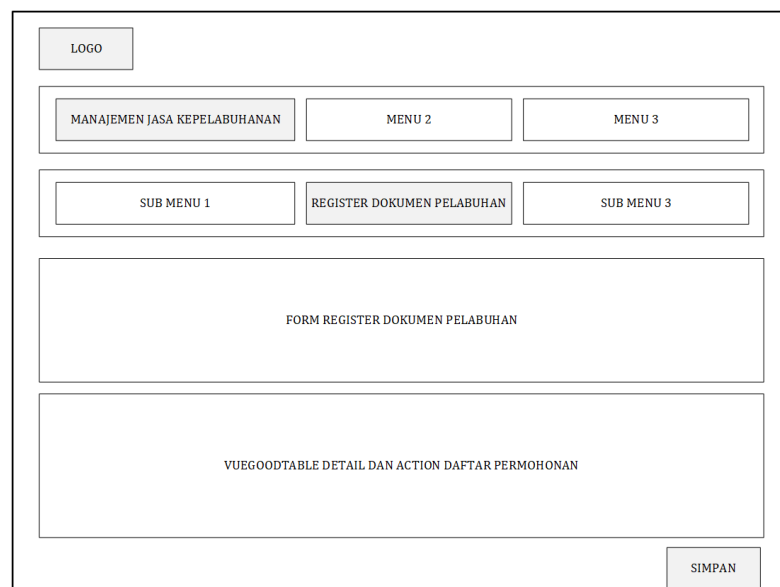
9. Menu Utama Daftar Dokumen Pelabuhan



Gambar 3.28 Menu Utama Daftar Dokumen Pelabuhan

Pada Gambar 3.28 merupakan antarmuka sub menu utama daftar dokumen pelabuhan yang diakses oleh agen kapal. Pada menu ini berisi tabel dari daftar dokumen pelabuhan yang sudah di-*register* oleh agen kapal ke dalam sistem.

10. Menu Utama *Register* Dokumen Pelabuhan



Gambar 3.29 Menu Utama *Register* Dokumen Pelabuhan

Pada Gambar 3.29 merupakan antarmuka sub menu utama *register*

dokumen pelabuhan yang diakses oleh agen kapal. Pada menu ini agen kapal melakukan *register* dengan melakukan *upload* pada form dokumen yang berhubungan dengan kegiatan kapal yang dimiliki oleh perusahaan kapal.

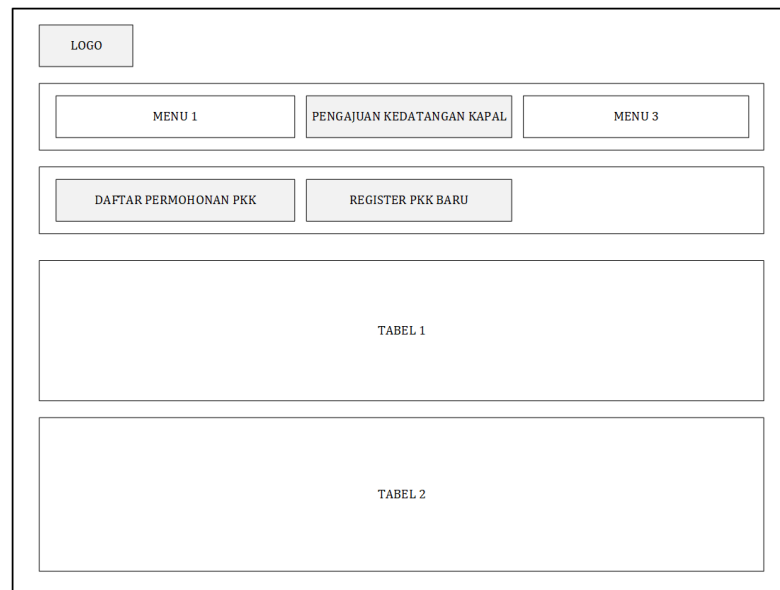
11. Menu Utama Cetak Form

The diagram illustrates the layout of the 'Menu Utama Cetak Form' (Main Menu Print Form). It features a top section with a 'LOGO' button. Below this is a row of three buttons: 'MANAJEMEN JASA KEPELABUHAN', 'MENU 2', and 'MENU 3'. The next row contains three buttons: 'SUB MENU 1', 'SUB MENU 2', and 'JASA PELABUHAN'. A large rectangular area is designated for 'FORM Pencarian NO PKK/PUK'. Below the search form is a 'TAMBAH JASA BARU' button. At the bottom, a large box is labeled 'VUEGOODTABLE ACTION DAN DETAIL UNTUK CETAK FORM'.

Gambar 3.30 Menu Utama Cetak Form

Pada Gambar 3.30 merupakan antarmuka sub menu utama jasa pelabuhan yang diakses oleh agen kapal. Pada menu ini agen kapal dapat melakukan cetak form kegiatan berdasarkan pencarian no pkk/puk yang ada sesuai dengan status nota jasa dalam sistem. Pada menu ini agen kapal juga dapat menambahkan kegiatan jasa pelabuhan baru.

12. Menu Utama Pengajuan Kedatangan Kapal/PKK



Gambar 3.31 Menu Utama Pengajuan Kedatangan Kapal/PKK

Pada Gambar 3.31 merupakan antarmuka menu utama pengajuan kedatangan kapal yang diakses oleh agen kapal. Pada menu ini memiliki dua sub menu yaitu: sub menu Daftar Permohonan PKK dan sub menu Register PKK Baru. Untuk penjelasan dari sub menu akan dijabarkan pada gambar selanjutnya.

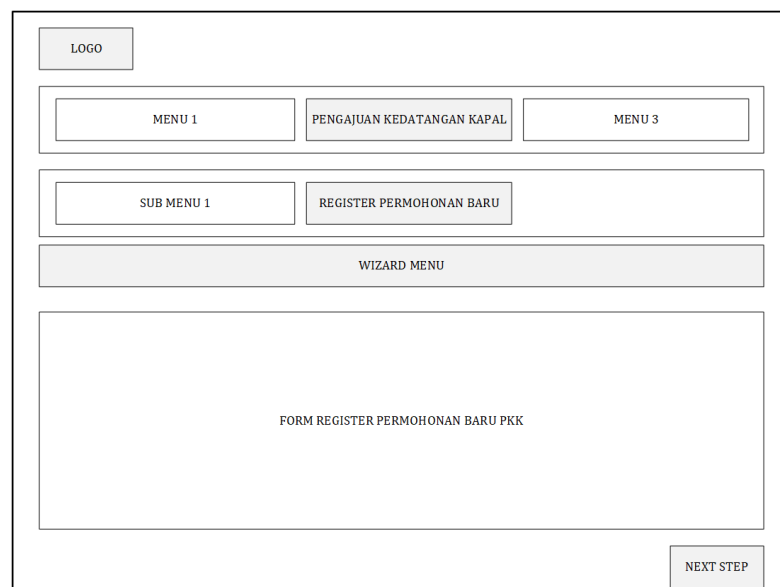
13. Menu Utama Daftar Permohonan PKK



Gambar 3.32 Menu Utama Daftar Permohonan PKK

Pada Gambar 3.32 merupakan antarmuka sub menu utama daftar permohonan PKK yang diakses oleh agen kapal. Pada menu ini berisi tabel dari daftar permohonan pkk yang sudah di-*register* oleh agen kapal ke dalam sistem.

14. Menu Utama *Register* Permohonan PKK Baru



Gambar 3.33 Menu Utama Register Permohonan PKK Baru

Pada Gambar 3.33 merupakan antarmuka sub menu utama *register*

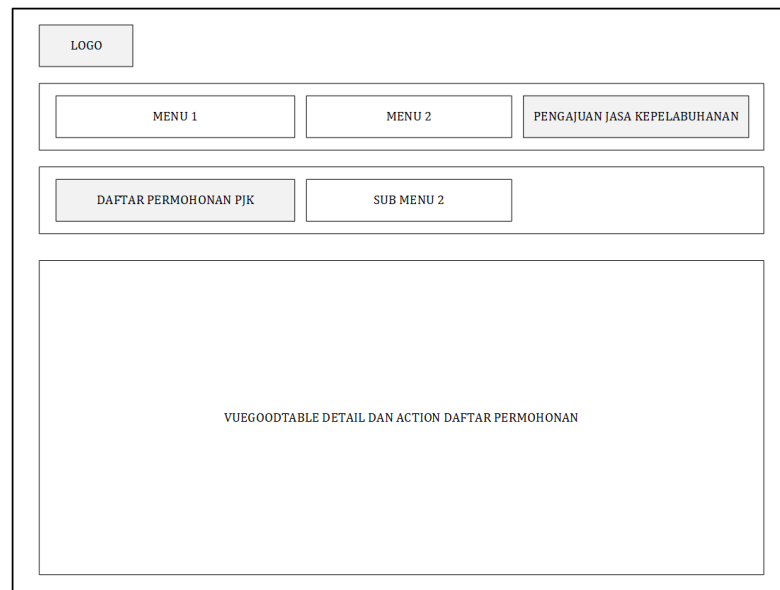
permohonan PKK yang diakses oleh agen kapal. Pada menu ini agen kapal melakukan *register* dengan mengisi *wizard* form yang berhubungan dengan kegiatan pkk yang dimiliki oleh perusahaan kapal.

15. Menu Utama Pengajuan Jasa Kepelabuhanan/PJK

Gambar 3.34 Menu Utama Pengajuan Jasa Kepelabuhanan/PJK

Pada Gambar 3.34 merupakan antarmuka menu utama pengajuan jasa kepelabuhanan yang diakses oleh agen kapal. Pada menu ini memiliki dua sub menu yaitu: sub menu Daftar Permohonan PJK dan sub menu Register PJK Baru. Untuk penjelasan dari sub menu akan dijabarkan pada gambar selanjutnya.

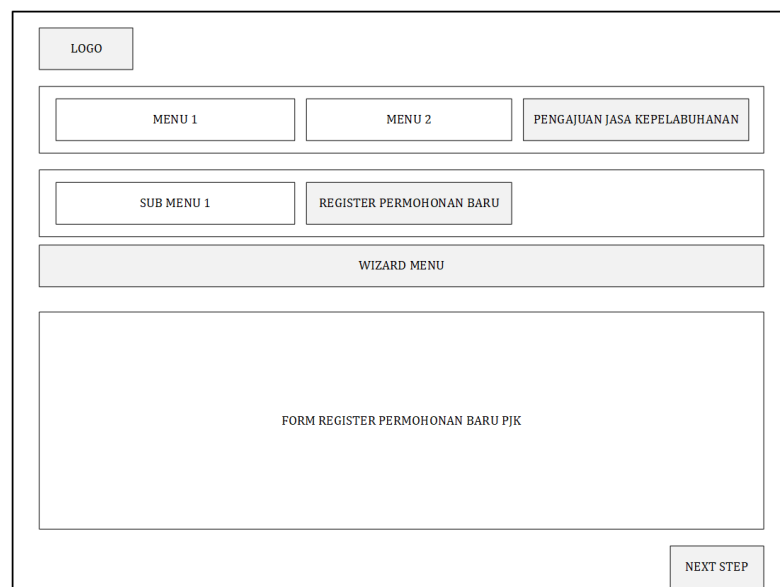
16. Menu Utama Daftar Permohonan PJK



Gambar 3.35 Menu Utama Daftar Permohonan PJK

Pada Gambar 3.35 merupakan antarmuka sub menu utama daftar permohonan PJK yang diakses oleh agen kapal. Pada menu ini berisi tabel dari daftar permohonan pjk yang sudah di-*register* oleh agen kapal ke dalam sistem.

17. Menu Utama *Register* Permohonan PJK Baru



Gambar 3.36 Menu Utama Register Permohonan PJK Baru

Pada Gambar 3.36 merupakan antarmuka sub menu utama *register*

permohonan PJK yang diakses oleh agen kapal. Pada menu ini agen kapal melakukan *register* dengan mengisi *wizard* form yang berhubungan dengan kegiatan pjk yang dimiliki oleh perusahaan kapal.

3.4 Desain Penelitian

3.4.1 Pengujian *Black-Box*

Pengujian *black-box* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak apakah berhasil atau tidak. Pengujian *black-box* cukup meninjau *input* dan *output* perangkat lunak tanpa pengetahuan tentang internal programnya. Pengujian ini menggunakan skenario pengujian dengan membuat daftar bagian-bagian yang akan diuji dan hasil yang diharapkan. Adapun skenario pengujian dijabarkan pada Tabel 3.33 di bawah ini :

Tabel 3.33 Skenario Pengujian *Black Box*

No	Test Case	Skenario	Hasil yang diharapkan
1	<i>Login</i>	Mengisi <i>email</i> dan <i>password</i> dengan agent1@example.com “secret”, klik <i>sign in</i>	Sistem berhasil <i>login</i> dan masuk membuka akses menu utama
		Mengosongkan <i>email</i> dan <i>password</i> , mengisi <i>email</i> dan mengosongkan <i>password</i> , mengosongkan <i>email</i> dan mengisi <i>password</i> , mengisi <i>email</i> dan <i>password</i> sembarangan, lalu klik <i>login</i>	Sistem akan menolak dan menampilkan pesan “ <i>Invalid email and password</i> ”
2	Menambah <i>Role User</i>	Membuka form tambah <i>role user</i> , mengisikan <i>role user</i> baru, lalu klik <i>save</i>	Sistem menyimpan data <i>user</i> baru dan menampilkan data pada grid <i>user</i>
		Membuka form tambah <i>role user</i> , mengosongkan data <i>role user</i> ,	Sistem akan menolak dan menampilkan pesan “ <i>the given</i>

		lalu klik <i>save</i>	<i>data was invalid</i>
3	Menambah Menu <i>User</i>	Membuka form tambah menu <i>user</i> , mengisikan menu baru, lalu klik <i>save</i>	Sistem menyimpan data menu baru dan menampilkan data pada grid menu
		Membuka form tambah menu <i>user</i> , mengosongkan data menu, lalu klik <i>save</i>	Sistem akan menolak dan menampilkan pesan “ <i>the given data was invalid</i> ”
4	Menambah <i>Permission</i>	Membuka form tambah <i>permission</i> menu <i>role user</i> , mengisikan <i>permission</i> baru, lalu klik <i>save</i>	Sistem menyimpan data <i>permission</i> baru dan menampilkan data pada grid <i>permission</i>
		Membuka form tambah <i>permission</i> menu <i>role user</i> , mengosongkan data <i>permission</i> , lalu klik <i>save</i>	Sistem akan menolak dan menampilkan pesan “ <i>the given data was invalid</i> ”
5	Mengubah Hak Akses <i>Permission</i>	Membuka Menu Ubah Hak Akses Menu Role User dan memilih dengan <i>checkboxlist</i> , lalu klik <i>save</i>	Sistem menyimpan data hak akses <i>permission</i> baru dan menampilkan data pada grid <i>permission</i>
6	Menghapus <i>Permission</i> Menu	Membuka menu utama admin, memilih menu <i>permission</i> , klik <i>icon delete</i> , klik <i>yes</i>	Sistem meminta konfirmasi hapus, ya berarti dihapus, sistem menghapus data dari <i>database</i> dan menu <i>permission</i>
7	Menambah Dokumen Pelabuhan	Membuka sub menu <i>register dokumen</i> , mengisi seluruh <i>field form</i> yang ada, klik simpan	Sistem menyimpan data dokumen pelabuhan baru dan menampilkan data pada grid sub menu daftar dokumen
		Membuka sub menu <i>register dokumen</i> , mengosongkan salah satu <i>field form</i> yang ada, klik	Sistem akan menolak dan menampilkan pesan “mohon untuk mengisi detail file

		simpan	dokumen kepelabuhanan”
8	Tampil Daftar Dokumen pelabuhan	Membuka sub menu daftar dokumen pelabuhan	Sistem akan menampilkan tabel daftar dokumen pelabuhan dari <i>database</i>
9	Pencarian Daftar Dokumen Pelabuhan	Membuka sub menu daftar dokumen pelabuhan, mengisi <i>keyword</i> pencarian	Sistem akan menampilkan data pencarian daftar dokumen pelabuhan pada tabel
			Jika tidak ada, akan tampil pesan “ <i>no data for table</i> ”
10	Melihat Detail Daftar Dokumen Pelabuhan	Membuka sub menu daftar dokumen pelabuhan, klik 2x pada salah satu data daftar dokumen pelabuhan pada tabel	Sistem akan menampilkan data detail daftar dokumen pelabuhan
11	Mengunduh Dokumen Pelabuhan	Membuka sub menu daftar dokumen pelabuhan, klik 2x pada salah satu data daftar dokumen pelabuhan pada tabel, klik <i>download</i>	Sistem akan melakukan <i>download</i> pada data daftar dokumen pelabuhan
12	Menghapus Dokumen Pelabuhan	Membuka sub menu daftar dokumen pelabuhan, klik 2x pada salah satu data daftar dokumen pelabuhan pada tabel, klik hapus	Sistem akan melakukan hapus pada data pada tabel dan <i>database</i>
13	Menambah Permohonan PKK	Membuka sub menu daftar permohonan pkk, mengisi seluruh <i>field form</i> yang ada pada masing-masing <i>wizard form</i> , klik simpan, klik <i>agree</i>	Sistem menyimpan data permohonan pkk baru dan menampilkan data pada grid sub menu daftar permohonan pkk
		Membuka sub menu daftar permohonan pkk, mengosongkan salah satu <i>field form</i> yang ada	Sistem akan menolak dan menampilkan pesan “ <i>the field is required</i> ”

		pada <i>wizard form</i> pertama, klik simpan	
		Membuka sub menu daftar permohonan pkk, mengosongkan salah satu <i>field form</i> yang ada pada <i>wizard form</i> kedua, klik simpan	Sistem akan menolak dan menampilkan pesan “ <i>the field is required</i> ”
		Membuka sub menu daftar permohonan pkk, mengosongkan salah satu <i>field form</i> yang ada pada <i>wizard form</i> ketiga, klik simpan	Sistem akan menolak dan menampilkan pesan “ <i>the field is required</i> ”
14	Tampil Daftar Permohonan PKK	Membuka sub menu daftar permohonan pkk	Sistem akan menampilkan tabel daftar permohonan pkk dari <i>database</i>
15	Pencarian Daftar Permohonan PKK	Membuka sub menu daftar permohonan pkk, mengisi <i>keyword</i> pencarian	Sistem akan menampilkan data pencarian daftar permohonan pkk pada tabel
			Jika tidak ada, akan tampil pesan “ <i>no data for table</i> ”
16	Melihat Detail Daftar Permohonan PKK	Membuka sub menu daftar permohonan pkk, klik <i>button detail</i> pada salah satu data daftar permohonan pkk pada tabel	Sistem akan menampilkan data detail daftar permohonan pkk
17	Melakukan <i>Submit</i> Persetujuan Permohonan PKK	Membuka sub menu daftar permohonan pkk, klik <i>button submit</i> pada salah satu data daftar permohonan pkk pada tabel, klik <i>yes</i>	Sistem akan menampilkan data daftar permohonan pkk dengan status yang sudah berubah menjadi “ <i>submitted</i> ”
18	Menambah	Membuka sub menu daftar	Sistem menyimpan data

	Permohonan PJK	permohonan pjk, mengisi seluruh <i>field form</i> yang ada pada masing-masing <i>wizard form</i> , klik simpan, klik <i>agree</i>	permohonan pjk baru dan menampilkan data pada grid sub menu daftar permohonan pjk
		Membuka sub menu daftar permohonan pjk, mengosongkan salah satu <i>field form</i> yang ada pada <i>wizard form</i> pertama, klik simpan	Sistem akan menolak dan menampilkan pesan “ <i>the field is required</i> ”
		Membuka sub menu daftar permohonan pjk, mengosongkan salah satu <i>field form</i> yang ada pada <i>wizard form</i> kedua, klik simpan	Sistem akan menolak dan menampilkan pesan “mohon memilih rencana kegiatan kepelabuhanan anda”
		Membuka sub menu daftar permohonan pjk, mengosongkan salah satu <i>field form</i> yang ada pada <i>wizard form</i> ketiga, klik simpan	Sistem akan menolak dan menampilkan pesan “mohon untuk melengkapi form yang tersedia”
		Membuka sub menu daftar permohonan pjk, mengosongkan <i>field form</i> yang ada pada <i>wizard form</i> keempat, klik simpan	Sistem akan menolak dan menampilkan pesan “pilih file”
19	Tampil Daftar Permohonan PJK	Membuka sub menu daftar permohonan pjk	Sistem akan menampilkan tabel daftar permohonan pjk dari <i>database</i>
20	Pencarian Daftar Permohonan PJK	Membuka sub menu daftar permohonan pjk, mengisi <i>keyword</i> pencarian	Sistem akan menampilkan data pencarian daftar permohonan pjk pada tabel
			Jika tidak ada, akan tampil

			pesan “no data for table”
21	Melihat Detail Daftar Permohonan PJK	Membuka sub menu daftar permohonan pjk, klik <i>button detail</i> pada salah satu data daftar permohonan pjk pada tabel	Sistem akan menampilkan data detail daftar permohonan pjk
22	Melakukan <i>Submit</i> Persetujuan Permohonan PJK	Membuka sub menu daftar permohonan pjk, klik <i>button submit</i> pada salah satu data daftar permohonan pjk pada tabel, klik <i>yes</i>	Sistem akan menampilkan data daftar permohonan pjk dengan status yang sudah berubah menjadi “submitted”
23	Mencetak Form	Membuka sub menu jasa pelabuhan, melakukan pencarian nomor puk/pkk pada grid tabel yang tampil, klik salah satu data nomor puk/pkk, lalu klik <i>button cetak SPK</i> untuk mencetak SPK, klik <i>button Berita Acara</i> itu mencetak BA Jasa Pelabuhan, klik <i>button Bukti Hold Dana</i> untuk mencetak Bukti Hold Dana, klik <i>button Nota Lunas</i> untuk mencetak Nota Lunas Jasa Kepelabuhanan, klik <i>button Log Transaksional Host to Host</i> untuk mencetak <i>log transaksional</i>	Sistem akan mencetak form berdasarkan jenis kegiatan jasa pelabuhan yang tersedia

3.4.2 Pengujian *System Usability Scale* (SUS)

Pengujian SUS digunakan untuk menganalisis sistem informasi berbasis komputer dengan aspek berupa: *Learnability* (Mudah dipelajari), *Efficiency*

(Efisien), *Memorability* (Kemudahan dalam mengingat), *Errors* (Pencegahan kesalahan), dan *Satisfaction* (Kepuasan pengguna) (Nielsen, 1994).

Berikut merupakan pengujian usability testing yang dilakukan dengan menggunakan metode perhitungan Skala Likert:

$$P = \frac{f}{N} \times 100\%$$

Dimana:

P = Persentase

f = Frekuensi data

N = Jumlah sampel yang diolah

Persentase Interval	Kriteria
0% – 19,99%	Sangat Tidak Setuju
20% – 39,99%	Tidak Setuju
40% – 59,99%	Netral
60% – 79,99%	Setuju
80% – 100%	Sangat Setuju

Gambar 3.37 Tabel Persentasi Interval

Untuk melakukan analisis penilaian pengguna, maka akan dilakukan menggunakan kuesioner. Setiap pernyataan dalam kuesioner diklasifikasikan berdasarkan aspek SUS yang telah disesuaikan dengan kebutuhan penulis dan spesifikasi aplikasi yang dibuat.

Pernyataan-pernyataan yang sudah diklasifikasikan ke dalam SUS dapat dilihat pada Tabel 3.34 berikut:

Tabel 3.34 Rancangan Kuesioner

No	Item Pernyataan	1= Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Ragu-Ragu, 4 = Setuju, 5 = Sangat Setuju
----	-----------------	--

1	Saya berpikir akan menggunakan sistem ini lagi	1	2	3	4	5
2	Saya merasa sistem ini rumit untuk digunakan (memuat banyak hal yang tidak perlu)	1	2	3	4	5
3	Saya merasa sistem ini mudah digunakan	1	2	3	4	5
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini	1	2	3	4	5
5	Saya merasa fitur-fitur yang disediakan pada situs ini dirancang dan disiapkan dengan baik	1	2	3	4	5
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)	1	2	3	4	5
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat	1	2	3	4	5
8	Saya merasa sistem ini membingungkan	1	2	3	4	5
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini	1	2	3	4	5
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini	1	2	3	4	5

Kemudian dalam melakukan pengujian aplikasi, akan diberikan kepada 30 responden (Roscoe, 1975), ukuran minimal responden untuk mendapatkan hasil yang baik. Adapun teknik pengambilan sampel yang digunakan yaitu *purposive sampling*, dimana merupakan teknik pengambilan sumber data dengan pertimbangan tertentu (Sugiyono, 2012). Pertimbangan yang dimaksud dalam penelitian ini dengan menentukan kriteria yaitu orang yang mengetahui secara umum dengan bidang *IT* serta telah menggunakan minimal satu aplikasi sistem informasi.

3.5 Lokasi dan Waktu Penelitian

Lokasi penelitian dilakukan di IT Centre BP Batam yang beralamat di Jl. Sudirman No.1, Batam Centre, Kota Batam, Indonesia. Adapun waktu yang digunakan selama penelitian yaitu sejak 1 Maret 2021 s/d 1 November 2021 dalam waktu kurang lebih 8 bulan.

3.6 Alat dan Bahan Pengumpulan Data

Selama pengumpulan data dibutuhkan beberapa alat dan bahan yang telah dijabarkan pada Tabel 3.35 sebagai berikut:

Tabel 3.35 Alat dan Bahan

Nama Perangkat	Kebutuhan
Laptop HP Pavilion Power Laptop 15-cb0xx	Alat untuk pengolahan data dan pengujian aplikasi
Aplikasi B-SIMS	Sebagai bahan pengujian
Lembar Kuesioner	Bahan untuk mengumpulkan data pengujian aplikasi dari responden
Ms Excel	Aplikasi untuk menghitung dan mengolah data dari responden
Smartphone	Alat untuk mendokumentasi pengujian bersama responden dan kebutuhan lainnya

3.7 Teknik Pengumpulan Data

Teknik pengumpulan data pada penelitian ini dilakukan dalam dua tahapan. Adapun tahapan pertama yaitu pada tahap *communication* yang dalam hal ini dilakukan wawancara. Untuk tahapan kedua yaitu pada tahap *deployment* dilakukan dengan menyebar kuesioner. Adapun langkah-langkah pengumpulan data pada tahap *communication* adalah sebagai berikut:

1. Menghubungi pimpinan departemen untuk mengadakan wawancara
2. Mencatat hasil wawancara

Adapun langkah-langkah pengumpulan data pada tahap *deployment* adalah sebagai berikut:

1. Menghubungi responden untuk melakukan pengujian aplikasi
2. Responden melakukan pengujian seluruh fitur aplikasi
3. Setelah selesai, responden mengisi lembar kuesioner penilaian pengguna untuk dilakukan evaluasi aplikasi.

Daftar Pustaka

- Aminudin. (2015). *Berkenalan dengan Laravel. In Cara Efektif Belajar Framework Laravel (p.2)*. Yogyakarta: Lokomedia.
- Darmadi, H. (2011). *Metode Penelitian Pendidikan*. Bandung: Alfabeta.
- Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis. *Jurnal Pengembangan IT (JPIT)*.
- Jogiyanto. (2009). *Sistem Informasi Manajemen*. Yogyakarta: Penerbit Andi.
- Kristanto, A. (2010). *Kupas Tuntas PHP & MySQL, Cable Book*. Klaten.
- Kristanto, A. (n.d.). *Kupas Tuntas PHP & MySQL, Cable Book*. Klaten: s.n.
- Munawar. (2005). *Pemodelan Visual Dengan UML*. Yogyakarta: Graha Ilmu.
- Nielsen, J. (1994). *Usability Engineering – Interactive Technologies*. Morgan Kauffman.
- Nusantara Halim, B. F., & Amirul Nashrullah, M. F. (n.d.). Analisis dan Implementasi Fitur Penjadwalan Berbasis Kalender Pada Website Novocall. *Journal of Applied Multimedia and Networking (JAMN)*.
- Roscoe, J. (1975). *Fundamental Research Statistic for the Behavioural Sciences, 2nd Edition*. New York: Holt Rinehart & Winston.
- Simarmata, J. (2010). *Rekayasa Web*. Yogyakarta: Penerbit Andi.
- Situmorang, S. (2010). *Analisis Data Untuk Riset Manajemen dan Bisnis*. Medan: USU Press.
- Sommerville, & Ian. (2011). *Software Engineering (9th Edition)*. USA: Pearson Education.
- Sugiyono. (2012). *Metode Penelitian Kuantitatif, Kualitatif dan R & D*. Bandung: CV. Alfabeta.
- Sutedi, A. (2010). *Hukum Perizinan Dalam Sektor Pelayanan Publik*. Jakarta: Sinar Grafika.
- Tullah, R., & Hanafri, M. I. (2014). Evaluasi Penerapan Sistem Informasi Pada Politeknik LP3I Jakarta Dengan Metode Pieces. *Jurnal Sisfotek Global*.
- Yuhefizar. (2012). *Cara Mudah Membangun Website Interaktif Menggunakan CMS Joomla Edisi Revisi*. Jakarta: PT. Elex Media Komputindo.

Z, S., & H, B. S. (2016,2017). *An Indonesian adaption of the System Usability Scale (SUS), "in International Conference on Advanced Computer Science and Information Systems, ICACISIS*. pp. 145-148.