

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

TUGAS AKHIR

Disusun oleh:

MAILANY

4311931003

Disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Sains
Terapan Teknik Multimedia dan Jaringan



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

HALAMAN PENGESAHAN

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

Disusun oleh:

MAILANY

4311931003

Telah diuji dan dipertahankan di depan Tim Penguji
dalam Sidang Tugas Akhir
pada tanggal
dan dinyatakan **LULUS**.

Batam, 2021

Disetujui dan disahkan oleh:

Pembimbing I,

Supardianto, S.S.T., M. Eng.
NIK. 113105

Tim Penguji,

Penguji I,

Penguji II,

Metta Santiputri, S.T., M.Sc., Ph.D
NIK. 100017

Hamdani Arif, S.Pd., M.Sc.
NIK. 117175

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*.

Pada konteks praktek di lapangan itu sendiri, pembuatan aplikasi B-SIMS tersebut, penulis tidak bekerja sendiri, melainkan bekerja secara berkelompok mengingat aplikasi yang dibuat merupakan lingkup pekerjaan yang besar dan kompleks. Adapun tim dalam pembuatan aplikasi B-SIMS terdapat 7 orang, dimana dalam tim tersebut mencakup tugas dan fungsi sebagai *project manager*, *programmer backend*, *programmer frontend*, *system analyst*, dan *database analyst*. Sedangkan penulis sendiri memiliki tugas dan fungsi sebagai *programmer frontend*.

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 2.1 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 (Yuhfizhar, 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.

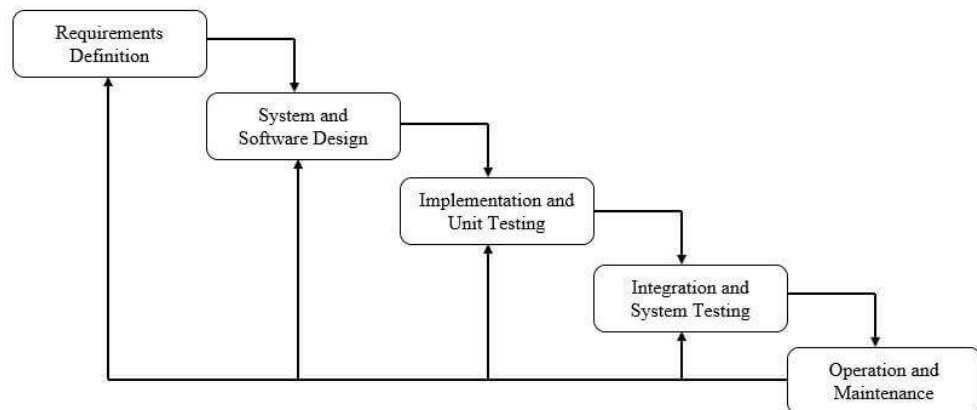
2.8 MySQL (*My Structured Query Language*)

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 2.1 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 kuisioner 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.

Usability merupakan parameter yang cukup berpengaruh pada keberhasilan sebuah aplikasi. *Usability* menurut ISO 9241-11 bila dijelaskan adalah sebagai berikut:

1. Efektif

Efektif adalah ketetapan pengguna dalam lingkungan tertentu untuk mencapai sebuah tujuan tertentu

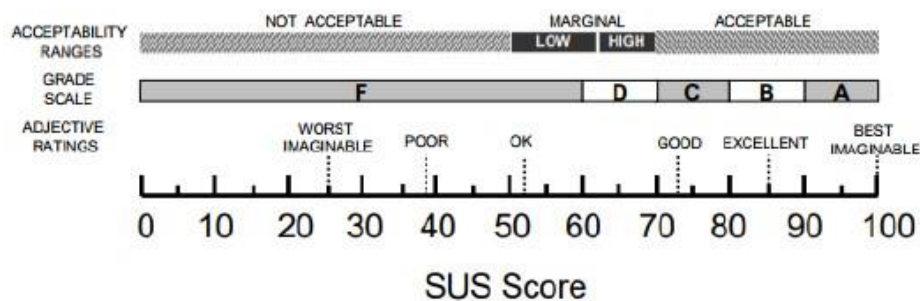
2. Efisien

Efisien adalah kemampuan yang digunakan *user* dalam mencapai tujuan

3. Kepuasan

Kepuasan adalah kebebasan dari ketidaknyaman dan perilaku positif dari sebuah produk.

Berikut adalah gambar kategori pengujian skor SUS.



Gambar 2.2 Kategori 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 2.2 Standar Kuisioner SUS

No	Item Pernyataan	1= Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Ragu-Ragu, 4 = Setuju, 5 = Sangat Setuju				
		1	2	3	4	5
1	Saya berpikir akan menggunakan sistem ini lagi					
2	Saya merasa sistem ini rumit untuk digunakan (memuat banyak hal yang tidak perlu)					
3	Saya merasa sistem ini mudah digunakan					

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. (J, C, A, & P, 2020)

$$((P1-1)+(P3-1)+(P5-1)+(P7-1)+(P9-1)+(5-P2)+(5-P4)+(5-P6)+(5-P8)+(5-P10)) \times 2.5$$

Selanjutnya dicari skor rata-rata dari skor SUS dari setiap responden dengan cara menjumlahkan seluruh skor dan dibagi jumlah responden. Berikut rumus untuk memperoleh rata-rata skor SUS:

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

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

Dimana \bar{x} merupakan skor rata-rata, $\sum x$ adalah jumlah skor, dan n adalah jumlah responden. Dari hasil perhitungan skor rata-rata SUS dapat disimpulkan dalam kategori nilai *Net Promoter Score* (NPS), *acceptable*, *grade*, seperti pada Tabel 2.3 di bawah ini:

Tabel 2.3 Acceptable Range

Acceptability	Range
Acceptable (High)	62-100
Acceptable (Low)	49-61
Not Acceptable	0-50

Pada Tabel 2.4 merupakan cara membaca skor untuk mengetahui kisaran penerimaan terhadap sistem dan peringkat hasil penilaian pada sistem.

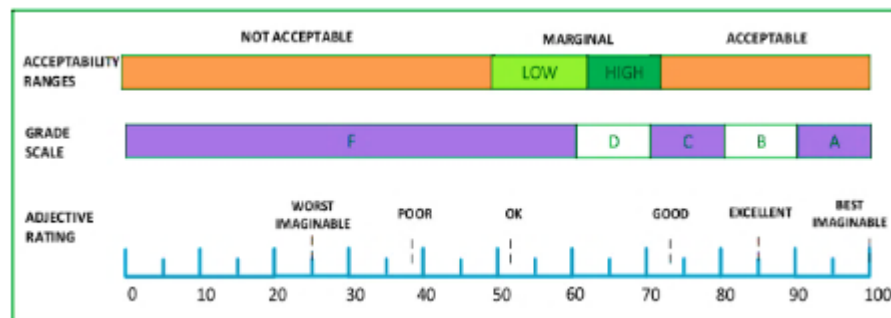
Tabel 2.4 SUS Skor

Skor	Peringkat	Grade
Skor ≥ 86	Best Imaginable	A
Skor ≥ 72 dan < 86	Excellent	B
Skor ≥ 52 dan < 72	Good	C
Skor ≥ 38 dan < 52	Ok/Fair	D
Skor ≥ 25 dan < 38	Poor	F

<25	Worst Imaginable	F
-----	------------------	---

Lalu pada Tabel 2.5 untuk menyatakan peringkat skala *grade* berdasarkan hasil penilaian menggunakan SUS.

Tabel 2.5 Interpretasi Skor SUS



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 2.6 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

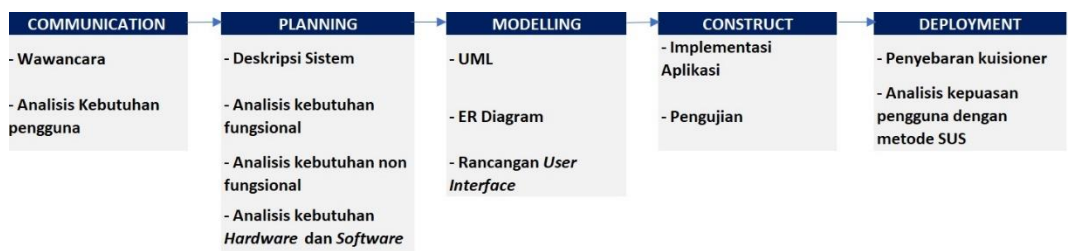
ANALISIS DAN PERANCANGAN

3.1 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*

3.2 Desain Pengembangan Aplikasi

3.2.1 Communication

Pada tahap ini terdapat 2 (dua) tahap yang dilakukan, yaitu sebagai berikut.

3.2.1.1 Wawancara

Metode yang digunakan untuk mengembangkan sistem perangkat lunak ini pada tahap ini dilakukan komunikasi berupa wawancara tidak terstruktur kepada pimpinan departemen Operasional TIK di BP Batam dan juga pegawai admin aplikasi internal BUP. Berdasarkan hasil wawancara, 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.

3.2.1.2 Analisis Kebutuhan Pengguna

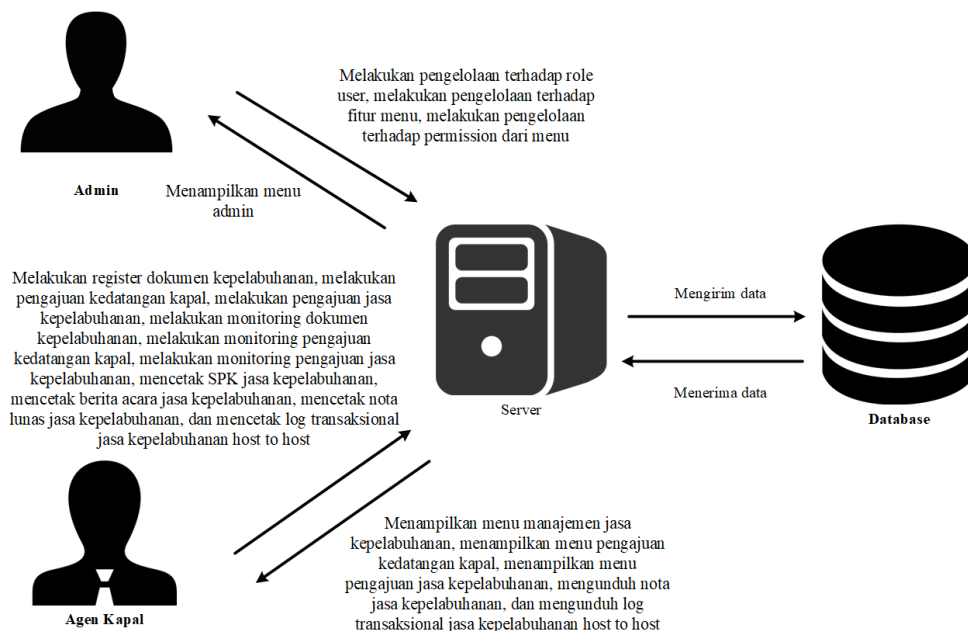
Dari hasil wawancara tersebut, penulis dapat memahami beberapa hal yang perlu dilakukan dalam pengerjaan Tugas Akhir ini. Sehingga, data yang dibutuhkan ialah:

1. BUP membutuhkan sebuah aplikasi yang dapat mengakomodir kegiatan perizinan jasa kepelabuhanan secara online untuk para agen kapal dan admin
2. Media untuk mengakses aplikasi dapat berupa PC
3. Fitur pada aplikasi perizininan kegiatan kepelabuhanan secara *online* yang dibutuhkan, ialah berupa: Fitur berupa kelola hak akses *user* terhadap menu aplikasi, manajemen terkait dokumen pelabuhan, manajemen terkait pengajuan kedatangan kapal, manajemen terkait pengajuan jasa kepelabuhan, dan cetak form yang berkaitan dengan kegiatan kepelabuhanan.

3.2.2 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. 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 Februari 2021 s/d 1 Agustus 2021 dalam waktu kurang lebih 6 bulan.

3.2.2.1 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 Agen Kapal. Adapun penjelasan ruang lingkup pengguna adalah sebagai berikut:

1. *Admin*
2. *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.
3. Agen Kapal
4. Agen 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 nota lunas jasa kepelabuhanan, dan mencetak *log* transaksional jasa kepelabuhanan *host to host*.

3.2.2.2 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 3.1 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 form

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 dapat diintegrasikan dengan sistem perizinan pelabuhan yang telah ada
- c. Sistem dapat berjalan pada berbagai macam jenis *browser*
- d. Sistem memiliki fungsi *login* pengguna untuk dapat masuk ke dalam sistem yang memiliki keamanan *password*
- e. Aplikasi memiliki *registrasi user* yang dilengkapi dengan konfirmasi *link* email untuk menjamin keamanan sistem
- f. Sistem memiliki keamanan *token based authentication* untuk verifikasi identitas user yang sudah ter-otorisasi ke dalam sistem dan memiliki *expired time* untuk menjamin keamanan sistem
- g. Database sistem menggunakan *password* untuk menjamin keamanan data
- h. Sistem mampu melakukan *loading data* besar dengan *response time* dibatasi 2 menit
- i. Sistem tersedia selama 24 jam sehari dan 365 hari dalam setahun
- j. Sistem menyertakan pengamanan dari virus, worm, trojan, dan jenis virus lainnya
- k. Sistem memiliki antarmuka yang dapat dipahami
- l. Memiliki dukungan edukasi oleh para pengembang melalui pelatihan aplikasi maupun tutorial sistem dalam bentuk pdf.

1. Analisis Kebutuhan Perangkat Keras

Adapun perangkat keras yang akan digunakan dalam pembuatan Tugas Akhir dijabarkan pada Tabel 3.2 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

2. Analisis Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam pembuatan aplikasi diantaranya:

- a. Sistem Operasi pada *server* menggunakan *CentOS 8*
- b. Web Server menggunakan *Apahe 2.4*
- c. *Ms Visio 2019* untuk mendesain *Use Case Diagram*, *Activity Diagram*, *ER Diagram*, dan *Sequence Diagram*
- d. *Microsoft Visual Studio 2017* digunakan untuk mendesain antarmuka
- e. *MySQL 8.0.17* dan *Oracle Database 12c* digunakan untuk *database server*
- f. *Navicat 15* digunakan untuk menjalankan perintah-perintah *SQL*.
- g. *GitLab* sebagai layanan *remote access* ke *git repositories* dalam mengelola siklus pengembangan aplikasi.

3.2.3 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*.

3.2.3.1 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:

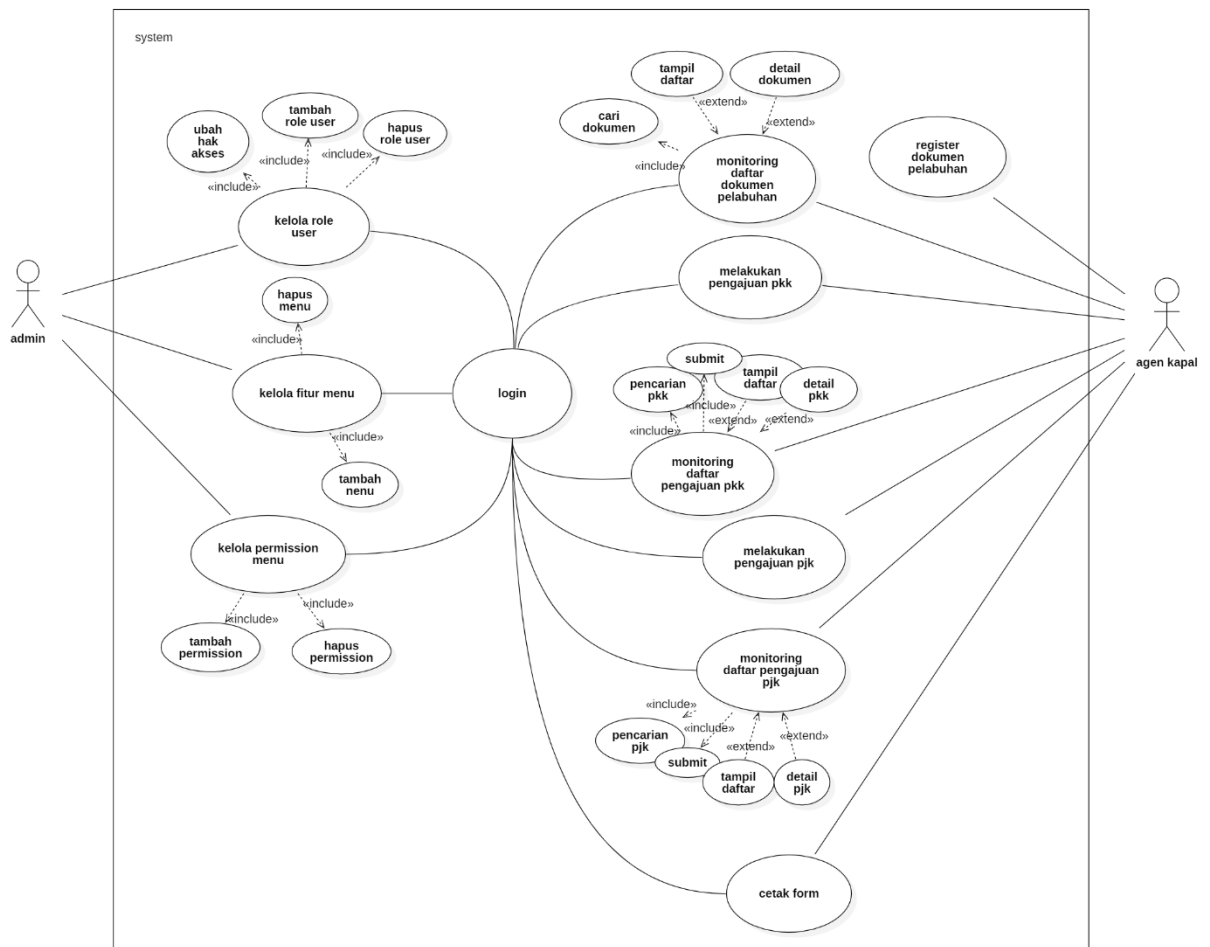
1. Admin dan Agen Kapal dapat menggunakan fitur *Login* (masuk kedalam sistem dengan memasukkan email dan password)
2. Admin dapat menggunakan fitur Mengelola Role User (melakukan penambahan user berdasarkan *role user* yang sudah ada dan melakukan penghapusan *user*)
3. Admin dapat menggunakan fitur 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 Form seperti: form SPK Jasa Kepelabuhanan (seperti melakukan unduh dan cetak Surat Perintah kerja (SPK) untuk jasa pandu), form Berita Acara Jasa Kepelabuhanan (seperti melakukan unduh dan cetak berita acara untuk jasa pandu), form Nota Lunas Jasa Kepelabuhanan (seperti melakukan unduh dan cetak nota lunas untuk jasa pandu), dan 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

3.2.3.2 Skenario Use Case

1. Skenario *Use Case* Login Admin

Tabel 3.3 Skenario Use Case Login Admin

Identifikasi			
Nomor use case:		3.3	
Nama use case:		Login Admin	
Deskripsi:		Proses login admin ke dalam aplikasi	
Aktor:		Admin	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem

1.	Klik menu <i>login</i>		
		2.	Tampil <i>form login</i>
3.	Masukkan <i>email</i> dan <i>password</i> pada <i>form login</i>		
		4.	Sistem melakukan validasi akun berdasarkan <i>role user</i> yang ada
5.	Tampil halaman utama admin		

2. Skenario *Use Case* Login Agen Kapal

Tabel 3.4 Skenario *Use Case Login* Agen Kapal

Identifikasi			
Nomor use case:		3.4	
Nama use case:		Login agen kapal	
Deskripsi:		Proses login agen kapal ke dalam aplikasi	
Aktor:		Agen kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Klik menu registrasi		
		2.	Tampil form registrasi
3.	Masukkan email dan password pada form registrasi		
		4.	Sistem mengirim link konfirmasi ke email yang di register
5.	Melakukan konfirmasi email pada link yang dikirim		
		6.	Sistem melakukan validasi

			akun berdasarkan <i>role user</i> baru di <i>database</i>
7.	Klik menu <i>login</i>		
		8.	Tampil form <i>login</i>
9.	Masukkan <i>email</i> dan <i>password</i> pada form <i>login</i>		
		10.	Sistem melakukan validasi akun berdasarkan <i>role user</i> yang ada
11.	Tampil halaman utama agen kapal		

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

Tabel 3.5 Skenario *Use Case* Mengelola *Role User*

Identifikasi			
Nomor use case:		3.5	
Nama use case:		Kelola <i>role user</i>	
Deskripsi:		Proses untuk melihat halaman kelola <i>role user</i> pada menu admin di aplikasi	
Aktor:		Admin	
No	Aksi Aktor	No	Respon Sistem
1.	Klik menu <i>permission</i>		
		2.	Tampil menu utama admin
3.	Klik tambah <i>role user</i>		
		4.	Tampilan menu tambah
5.	Klik hapus <i>role user</i>		
		6.	Data <i>role user</i> dihapus di <i>database</i>
7.	<i>Cheklis</i> hak akses <i>role user</i>		
		8.	Data hak akses <i>role user</i>

			dirubah di <i>database</i>
--	--	--	----------------------------

4. Skenario *Use Case* Mengelola Fitur Menu

Tabel 3.6 Skenario *Use Case* Mengelola Fitur Menu

Identifikasi			
Nomor use case:		3.6	
Nama use case		Kelola Fitur Menu	
Deskripsi:		Proses untuk melihat halaman kelola fitur menu pada menu admin di aplikasi	
Aktor:		Admin	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Klik menu <i>permission</i>		
		2.	Tampil menu utama admin
3.	Klik tambah menu		
		4.	Tampil form tambah menu
5.	Klik hapus menu		
		6.	Data dihapus dari <i>database</i>

5. Skenario *Use Case* Mengelola *Permission* Menu

Tabel 3.7 Skenario *Use Case* Mengelola *Permission* Menu

Identifikasi			
Nomor use case:		3.7	
Nama use case:		Kelola <i>Permission</i> Menu	
Deskripsi:		Proses untuk melihat halaman kelola fitur <i>permission</i> pada menu admin di aplikasi	
Aktor:		Admin	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Klik menu <i>permission</i>		

		2.	Tampil menu utama admin
3.	Klik tambah menu		
		4.	Tampil form tambah menu
5.	Klik hapus menu		
		6.	Data dihapus dari <i>database</i>

6. Skenario *Use Case Register* Dokumen Kepelabuhanan

Tabel 3.8 Skenario Use Case Menambah Dokumen Kepelabuhanan

Identifikasi			
Nomor use case:		3.8	
Nama use case:		Menambah Dokumen Pelabuhan	
Deskripsi:		Proses untuk menambah dokumen kepelabuhanan di aplikasi	
Aktor:		Agen Kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu manajemen jasa kepelabuhanan		
		2.	Tampil menu utama
3.	Pilih menu <i>register</i> dokumen pelabuhan		
		4.	Tampil form <i>register</i>
5.	Mengisi data form dan klik tambah		
		6.	Data tampil di <i>grid table</i>
7.	Klik simpan		
		8.	Data disimpan ke <i>database</i>

7. Skenario *Use Case Monitoring* Daftar Dokumen Kepelabuhanan

**Tabel 3.9 Skenario Use Case *Monitoring* Daftar Dokumen
Kepelabuhanan**

Identifikasi			
Nomor use case:		3.9	
Nama use case:		Monitoring Daftar Dokumen Pelabuhan	
Deskripsi:		Proses untuk monitoring daftar dokumen pelabuhan di aplikasi	
Aktor:		Agen Kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu manajemen jasa kepelabuhanan		
		2.	Tampil menu utama
3.	Pilih menu daftar dokumen		
		4.	Tampil tabel daftar dokumen yang di <i>register</i>
5.	Pilih fitur pencarian dan masukkan kata kunci		
		6.	Tampil data pencarian dari <i>database</i>
7.	Klik 2x data pada <i>grid table</i> daftar dokumen		
		8.	Tampil detail data

8. Skenario *Use Case* Menambah Permohonan Pengajuan Kedatangan Kapal (PKK)

Tabel 3.10 Skenario Use Case Menambah Permohonan PKK

Identifikasi	
Nomor use case:	3.10
Nama use case:	Tambah Permohonan PKK

Deskripsi:	Proses untuk menambah permohonan pkk di aplikasi		
Aktor:	Agen Kapal		
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu pengajuan kedatangan kapal		
		2.	Tampil menu utama
3.	Pilih menu <i>register</i> permohonan		
		4.	Tampil form <i>register</i>
5.	Mengisi data form dan klik simpan		
		6.	Data disimpan ke <i>database</i>

9. Skenario *Use Case Monitoring* Daftar Permohonan Pengajuan Kedatangan Kapal (PKK)

Tabel 3.11 Skenario Use Case *Monitoring* Daftar Permohonan Pengajuan Kedatangan Kapal (PKK)

Identifikasi			
Nomor use case:		3.11	
Nama use case:		Monitoring Daftar Permohonan PKK	
Deskripsi:		Proses untuk monitoring daftar permohonan PKK di aplikasi	
Aktor:		Agen Kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu pengajuan kedatangan kapal		
		2.	Tampil menu utama

3.	Pilih menu daftar permohonan		
		4.	Tampil tabel daftar permohonan pkk yang di <i>register</i>
5.	Pilih fitur pencarian dan masukkan kata kunci		
		6.	Tampil data pencarian dari <i>database</i>
7.	Klik 2x data pada <i>grid table</i> daftar dokumen		
		8.	Tampil detail data
8.	Tekan <i>submit</i> pada data daftar		
		9.	Tampil data daftar dengan status <i>submitted</i>

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

Tabel 3.12 Skenario Use Case Menambah Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Identifikasi			
Nomor use case:		3.12	
Nama use case:		Tambah Permohonan PJK	
Deskripsi:		Proses untuk menambah permohonan PJK	
Aktor:		Agen Kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu pengajuan jasa kepelabuhanan		
		2.	Tampil menu utama

3.	Pilih menu <i>register</i> permohonan		
		4.	Tampil form <i>register</i>
5.	Mengisi data form dan klik simpan		
		6.	Data disimpan ke <i>database</i>

11. Skenario *Use Case Monitoring* Daftar Pengajuan Jasa Kepelabuhanan

Tabel 3.13 Skenario Use Case *Monitoring* Daftar Pengajuan Jasa Kepelabuhanan

Identifikasi			
Nomor use case:		3.13	
Nama use case:		Monitoring Daftar Permohonan PJK	
Deskripsi:		Proses untuk monitoring daftar permohonan PJK di aplikasi	
Aktor:		Agen Kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu pengajuan jasa kepelabuhanan		
		2.	Tampil menu utama
3.	Pilih menu daftar permohonan		
		4.	Tampil tabel daftar permohonan pjk yang di <i>register</i>
5.	Pilih fitur pencarian dan masukkan kata kunci		
		6.	Tampil data pencarian dari <i>database</i>
7.	Klik 2x data pada <i>grid</i>		

	<i>table</i> daftar dokumen		
		8.	Tampil detail data
8.	Tekan <i>submit</i> pada data daftar		
		9.	Tampil data daftar dengan status <i>submitted</i>

12. Skenario *Use Case* Mencetak Form Kegiatan Jasa Kepelabuhanan

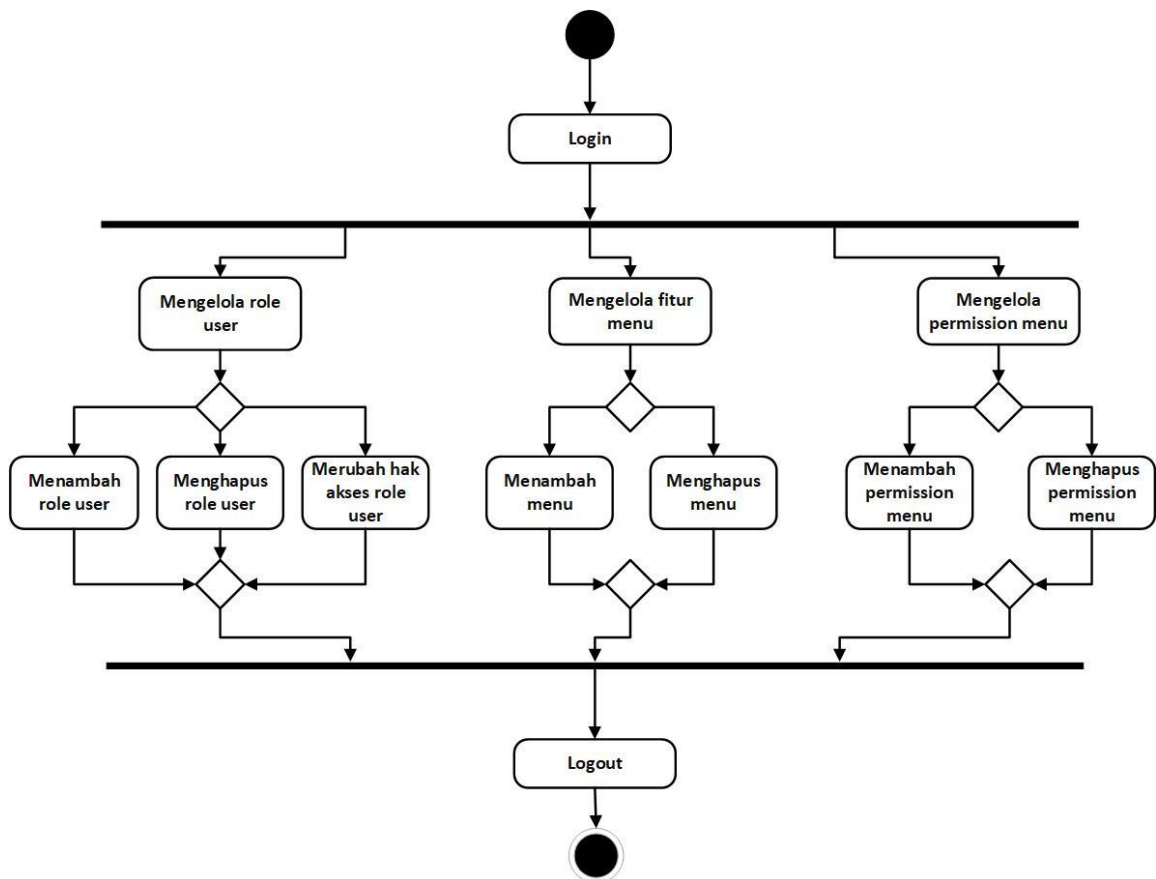
Tabel 3.14 Skenario Use Case Mencetak Form Kegiatan Jasa Kepelabuhanan

Identifikasi			
Nomor use case:		3.14	
Nama use case:		Cetak Form	
Deskripsi:		Proses untuk mencetak form kegiatan jasa kepelabuhanan	
Aktor:		Agen Kapal	
Skenario Utama			
No	Aksi Aktor	No	Respon Sistem
1.	Pilih menu manajemen jasa kepelabuhanan		
		2.	Tampil menu utama
3.	Pilih menu jasa pelabuhan		
		4.	Tampil menu pencarian
5.	Pilih data PKK		
		6.	Tampil data penggunaan jasa pelabuhan pada <i>grid table</i>
7.	Pilih cetak SPK		
		8.	Tampil menu cetak SPK
9.	Pilih cetak BA		
		10.	Tampil menu cetak BA

11.	Pilih cetak Nota Lunas		
		12.	Tampil menu cetak Nota Lunas
13.	Pilih cetak <i>Log transaksional</i>		
		14.	Tampil menu cetak <i>Log Transaksional</i>

3.2.3.3 Activity Diagram

1. Activity Diagram Admin

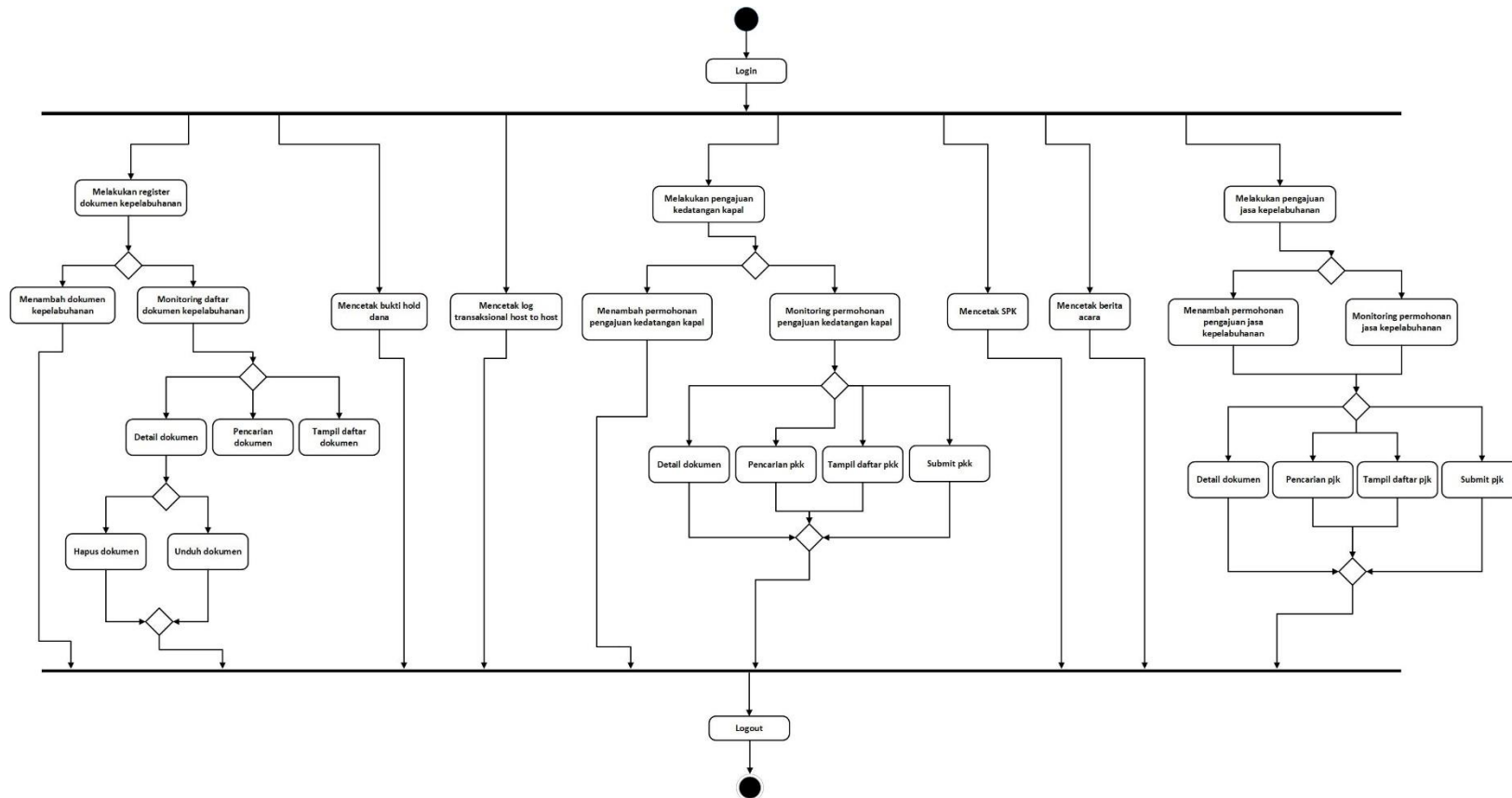


Gambar 3.4 Activity Diagram Admin

Pada gambar 3.4 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

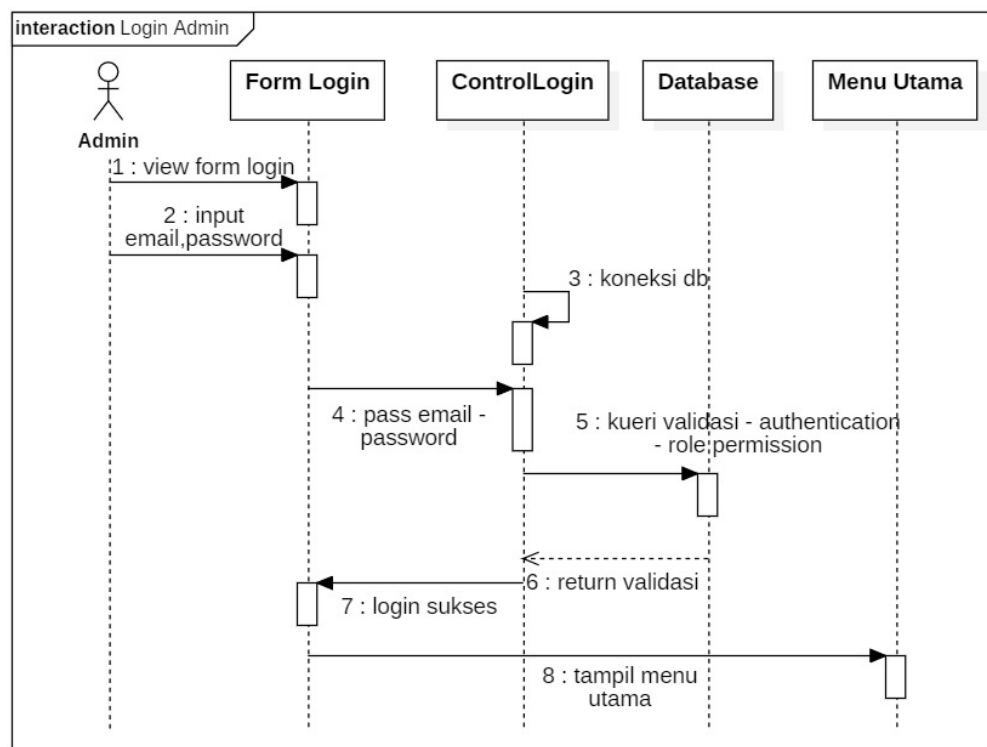


Gambar 3.5 Activity Diagram Agen Kapal

Pada Gambar 3.5 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*.

3.2.3.4 Sequence Diagram

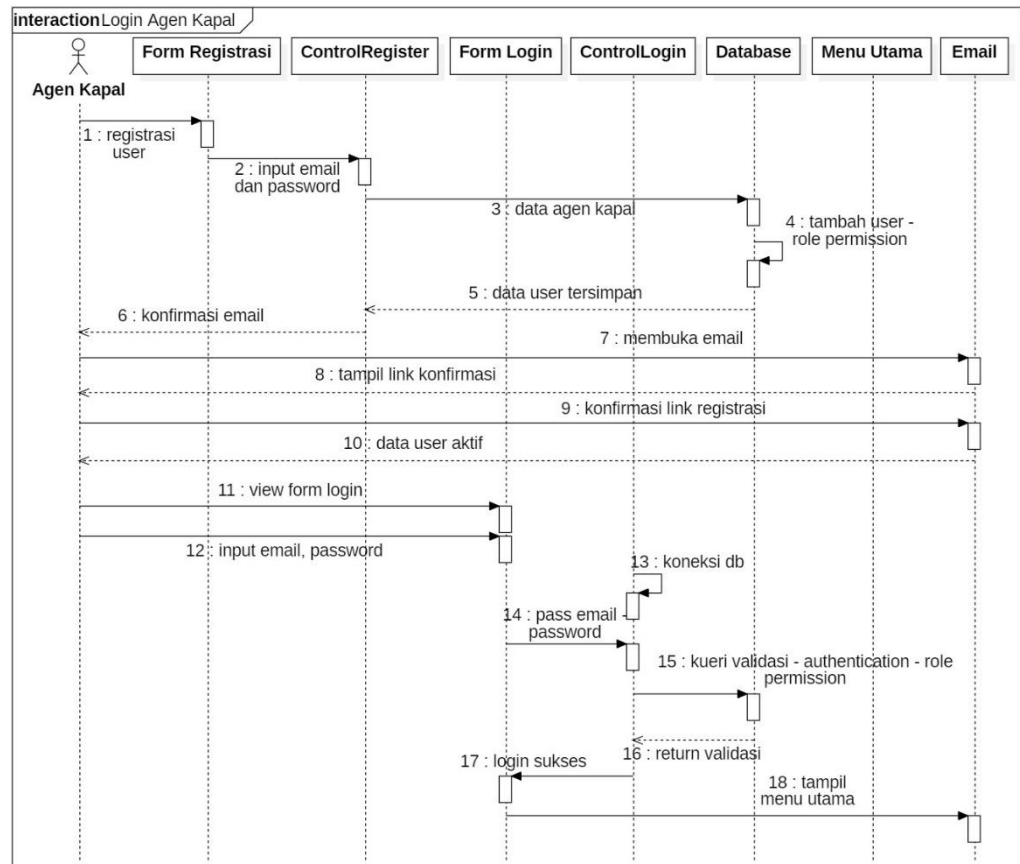
1. Sequence Diagram Login Admin



Gambar 3.6 Sequence Diagram Login Admin

Pada gambar 3.6 menunjukkan proses *login* admin sampai tampil pesan kesalahan jika gagal dan masuk ke aplikasi jika berhasil *login*.

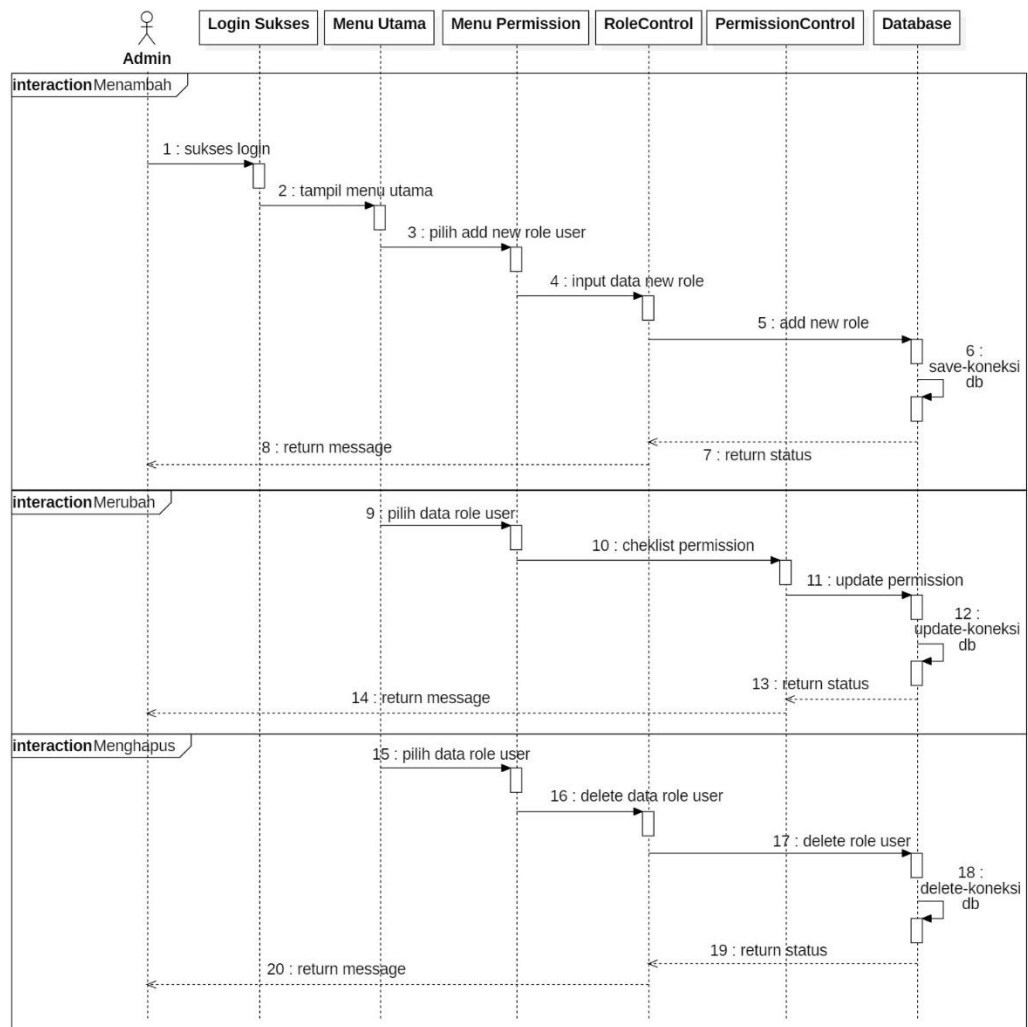
2. Sequence Diagram Login Agen Kapal



Gambar 3.7 Sequence Diagram Login Agen Kapal

Pada gambar 3.7 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*.

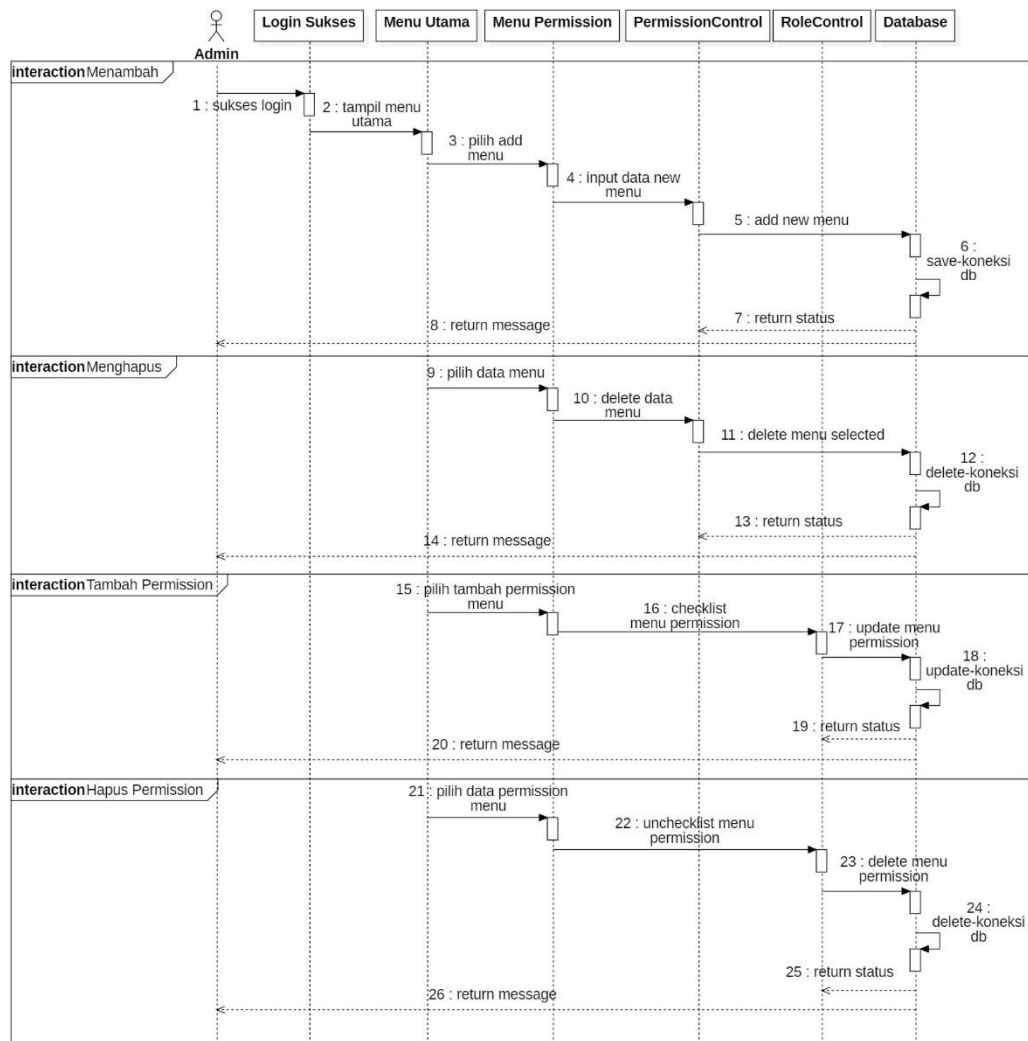
3. Sequence Diagram Manajemen Role User



Gambar 3.8 Sequence Diagram Manajemen Role User

Pada gambar 3.8 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*.

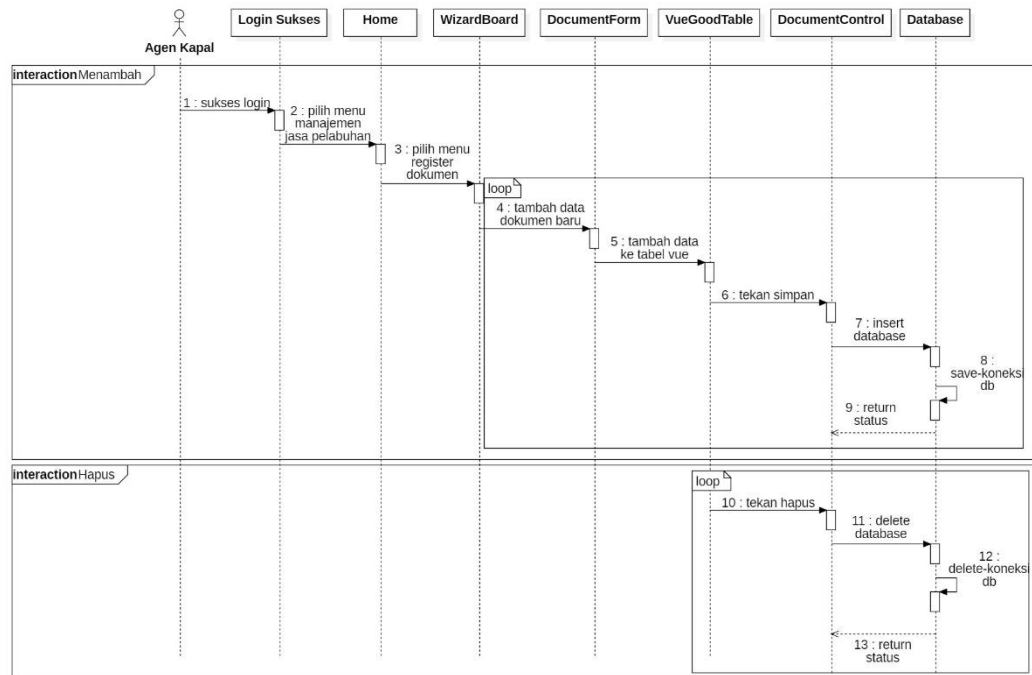
4. Sequence Diagram Manajemen Fitur Menu



Gambar 3.9 Sequence Diagram Manajemen Fitur Menu

Pada gambar 3.9 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*.

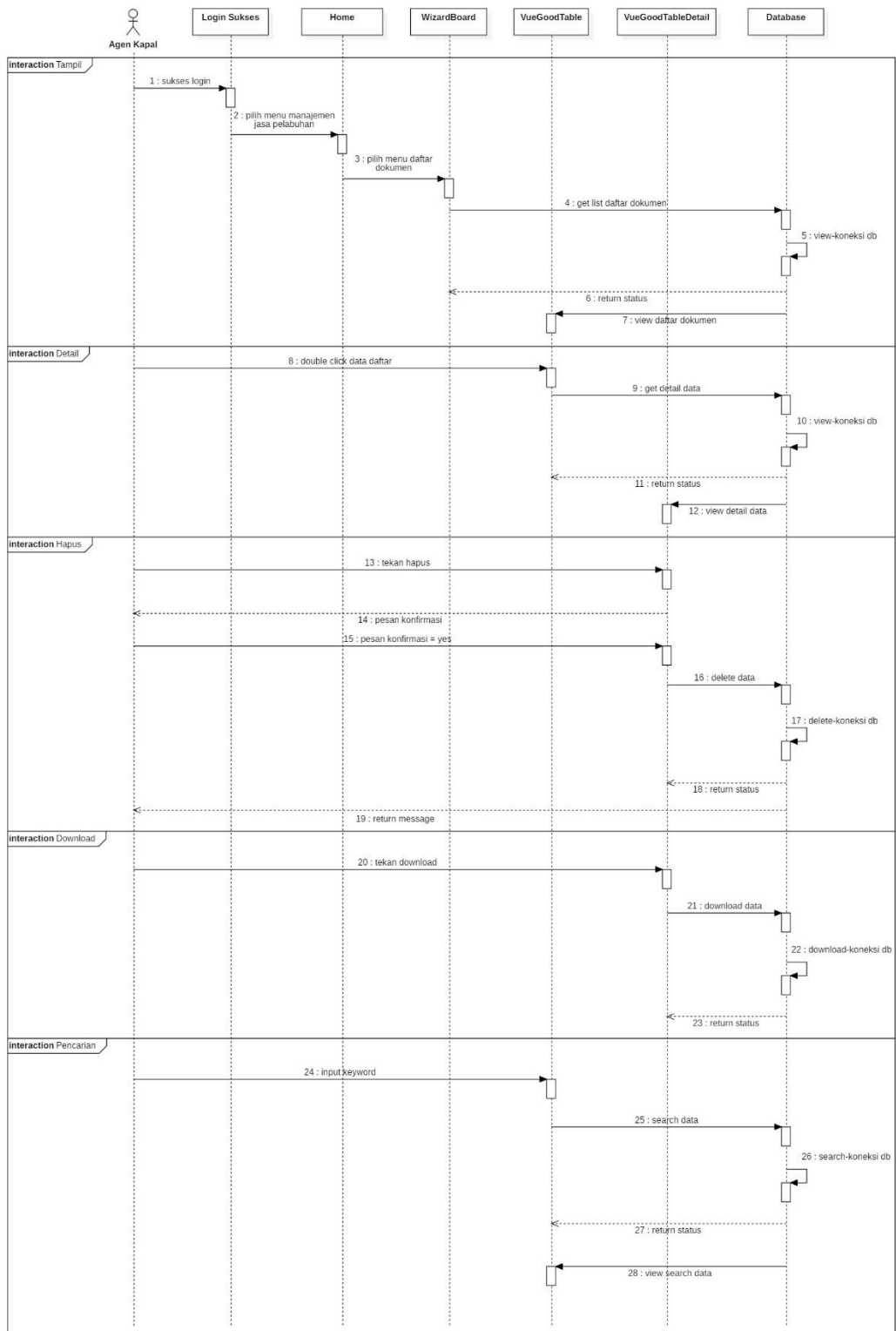
5. Sequence Diagram Manajemen Dokumen Kepelabuhanan (Menambah dan Menghapus)



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

Pada gambar 3.10 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)

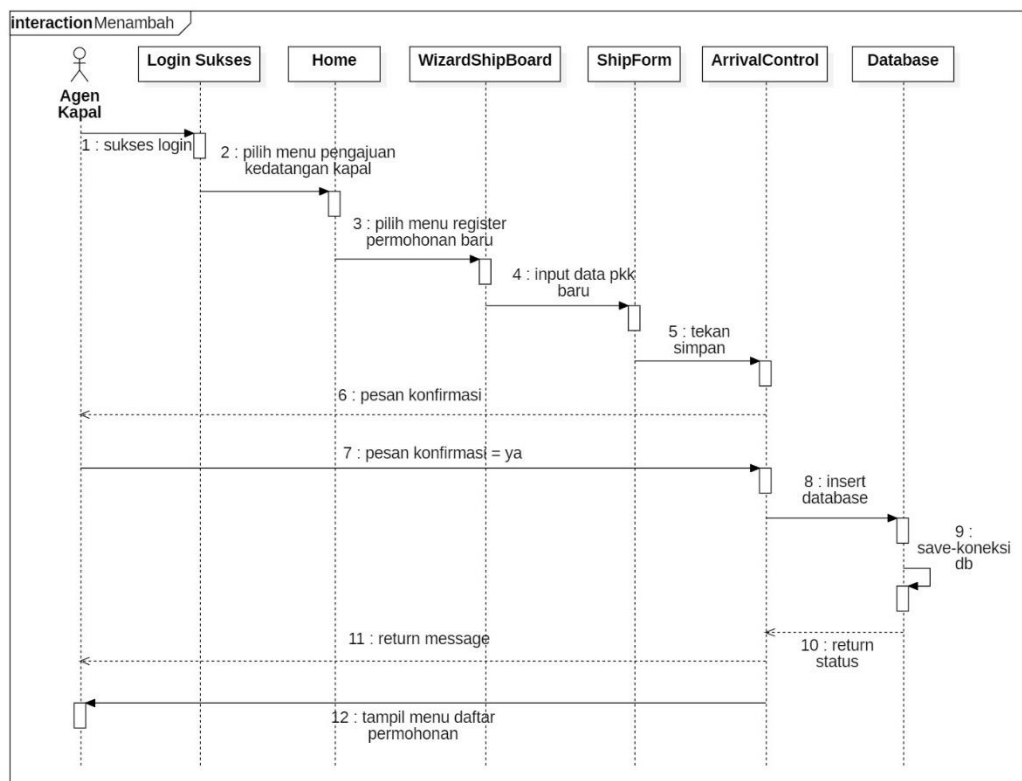


Gambar 3.11 Sequence Diagram *Monitoring* Dokumen Kepelabuhananan (Tampil daftar dokumen, detail daftar

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

Pada gambar 3.11 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*.

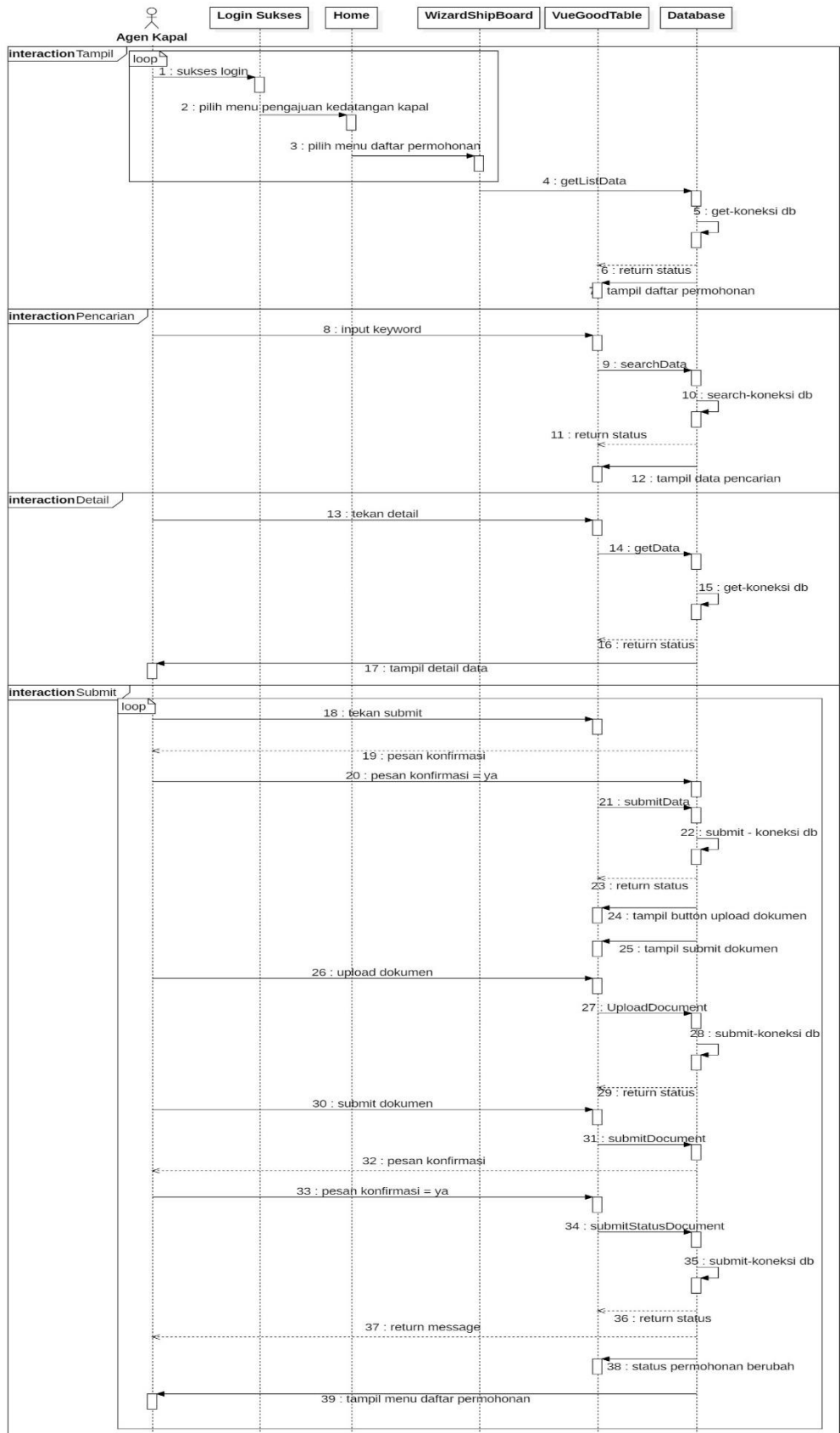
7. Sequence Diagram Menambah Permohonan Pengajuan Kedatangan Kapal



Gambar 3.12 Sequence Diagram Menambah Permohonan Pengajuan Kedatangan Kapal

Pada gambar 3.12 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*.

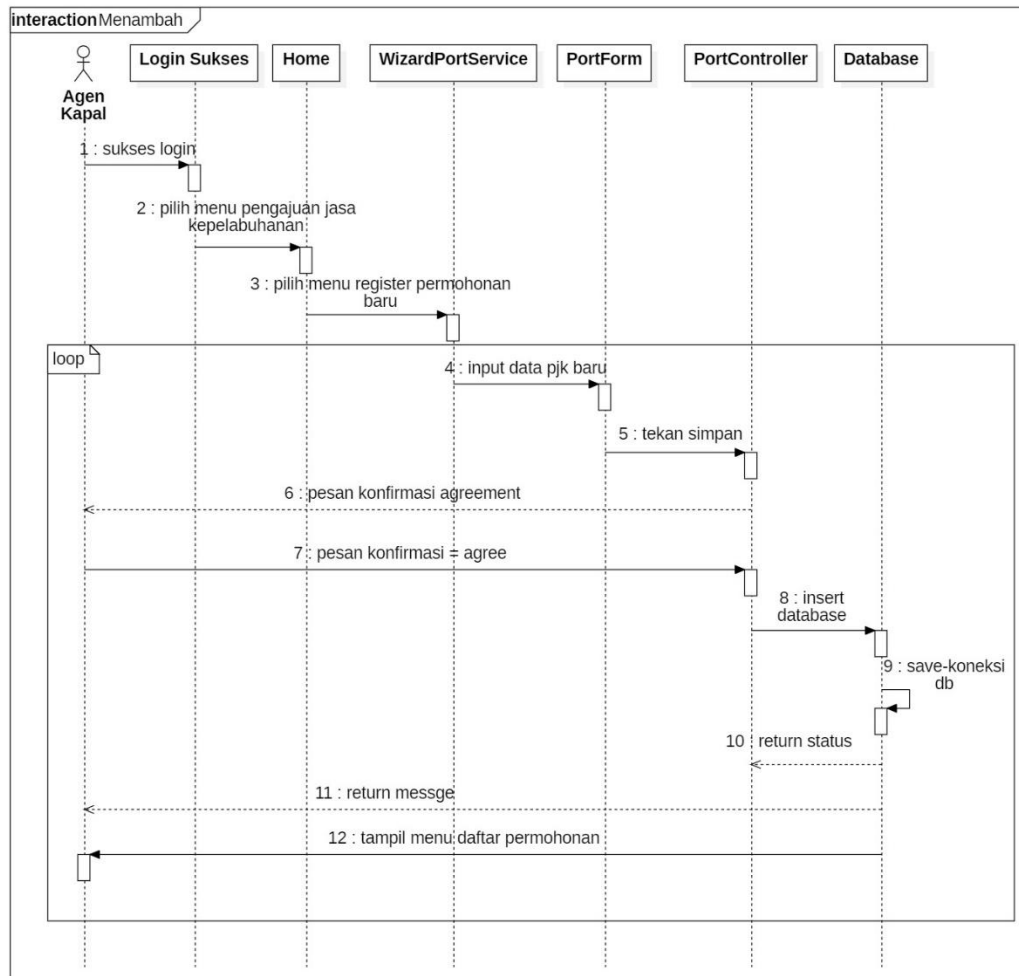
8. *Sequence Diagram Monitoring* Pengajuan Kedatangan Kapal/PKK
(Tampil daftar permohonan, pencarian permohonan, detail permohonan, dan *submit* dokumen permohonan)



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

Pada gambar 3.13 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)

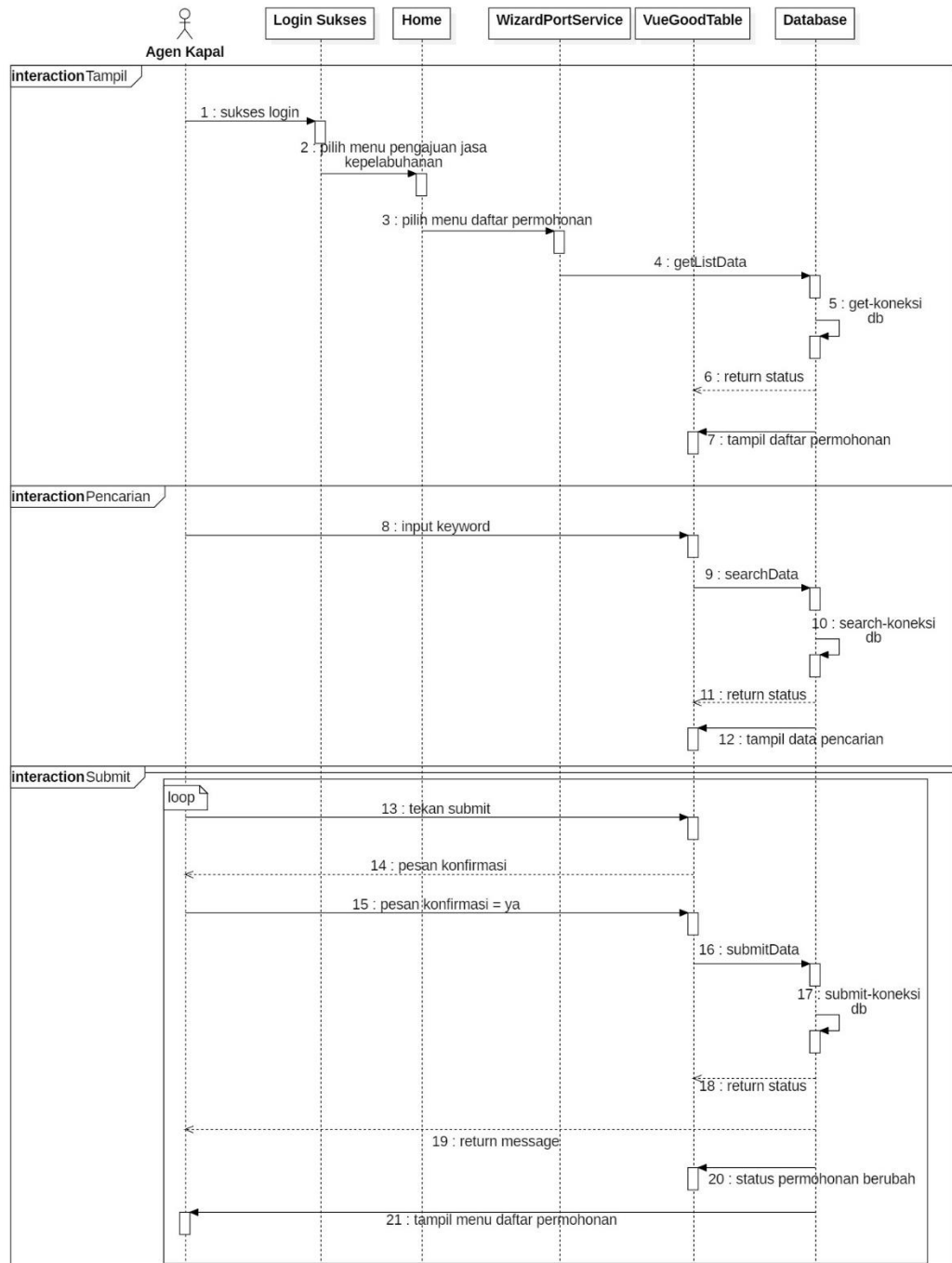


Gambar 3.14 Sequence Diagram Menambah Permohonan Pengajuan Jasa Kepelabuhanan (PJK)

Pada gambar 3.14 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)

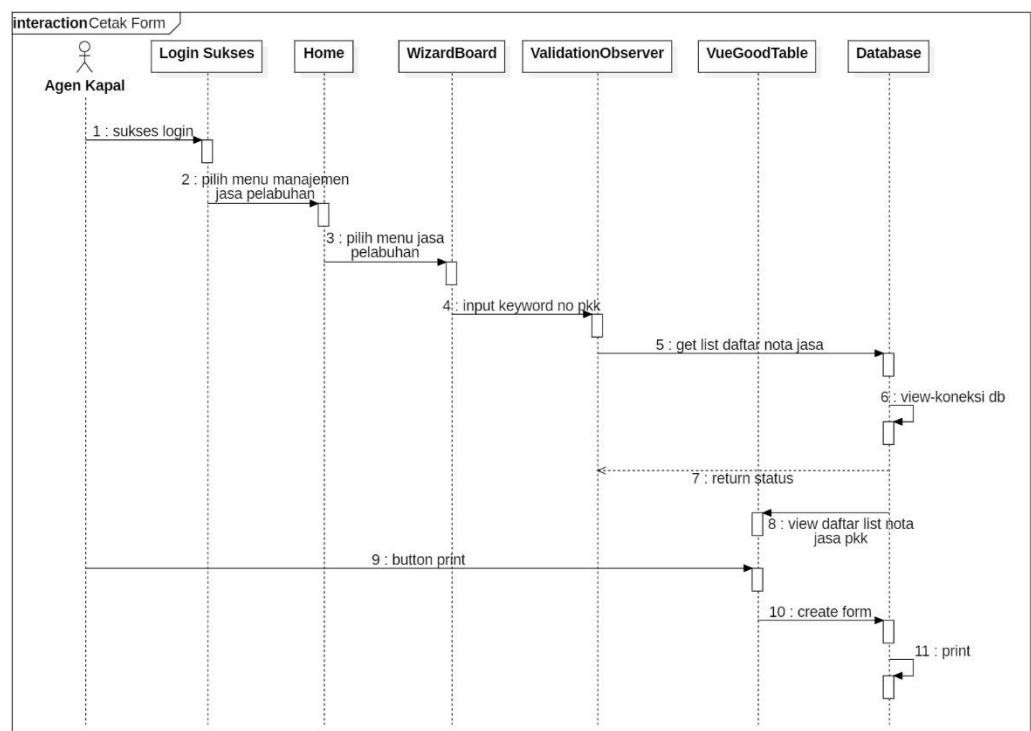


Gambar 3.15 Sequence Diagram *Monitoring* Pengajuan Jasa Kepelabuhanan/PJK (Tampil daftar permohonan, pencarian permohonan, dan submit dokumen permohonan)

Pada gambar 3.15 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, 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

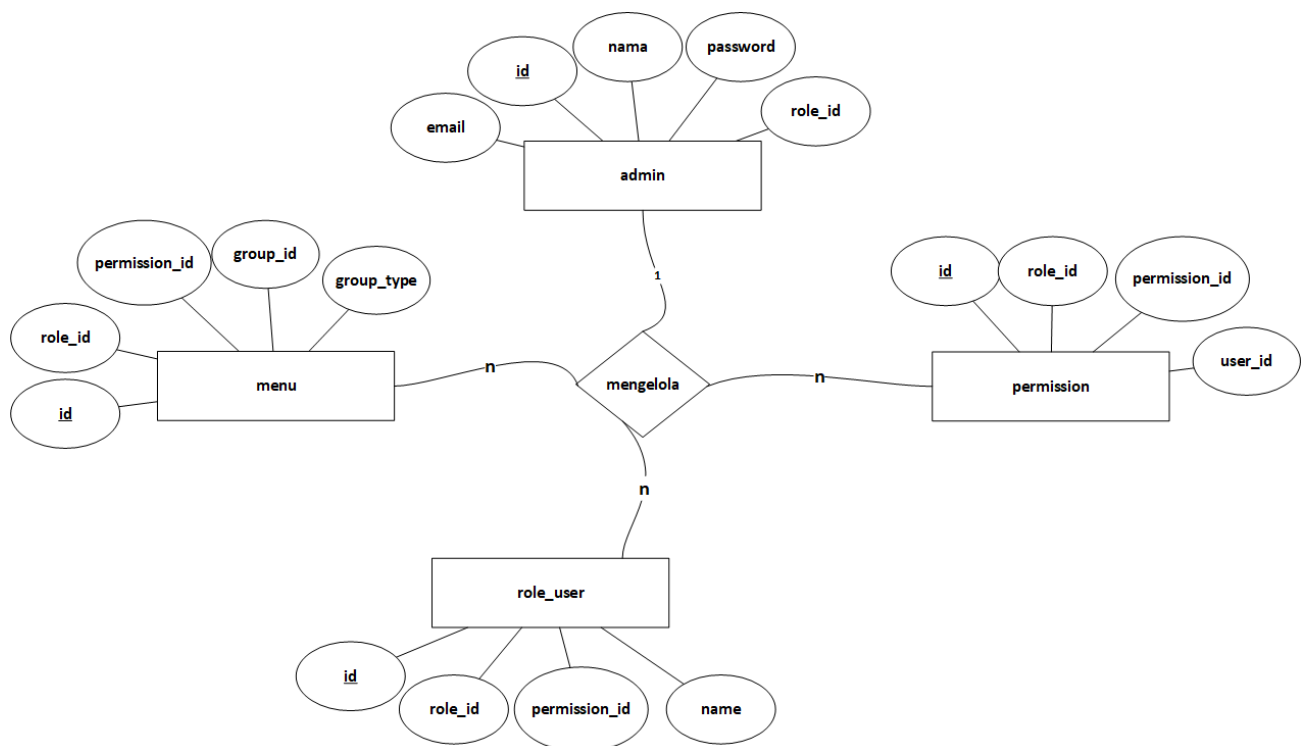
Pada gambar 3.16 menunjukkan proses untuk membuat form seperti: form SPK (Surat Perintah Kerja), form berita acara jasa kepelabuhanan, form nota lunas jasa kepelabuhanan, dan log transaksional jasa kepelabuhanan *host to host*. Data yang sudah tersimpan di *database* akan ditampilkan sebagai form yang dapat dicetak.

3.2.3.5 Class Diagram

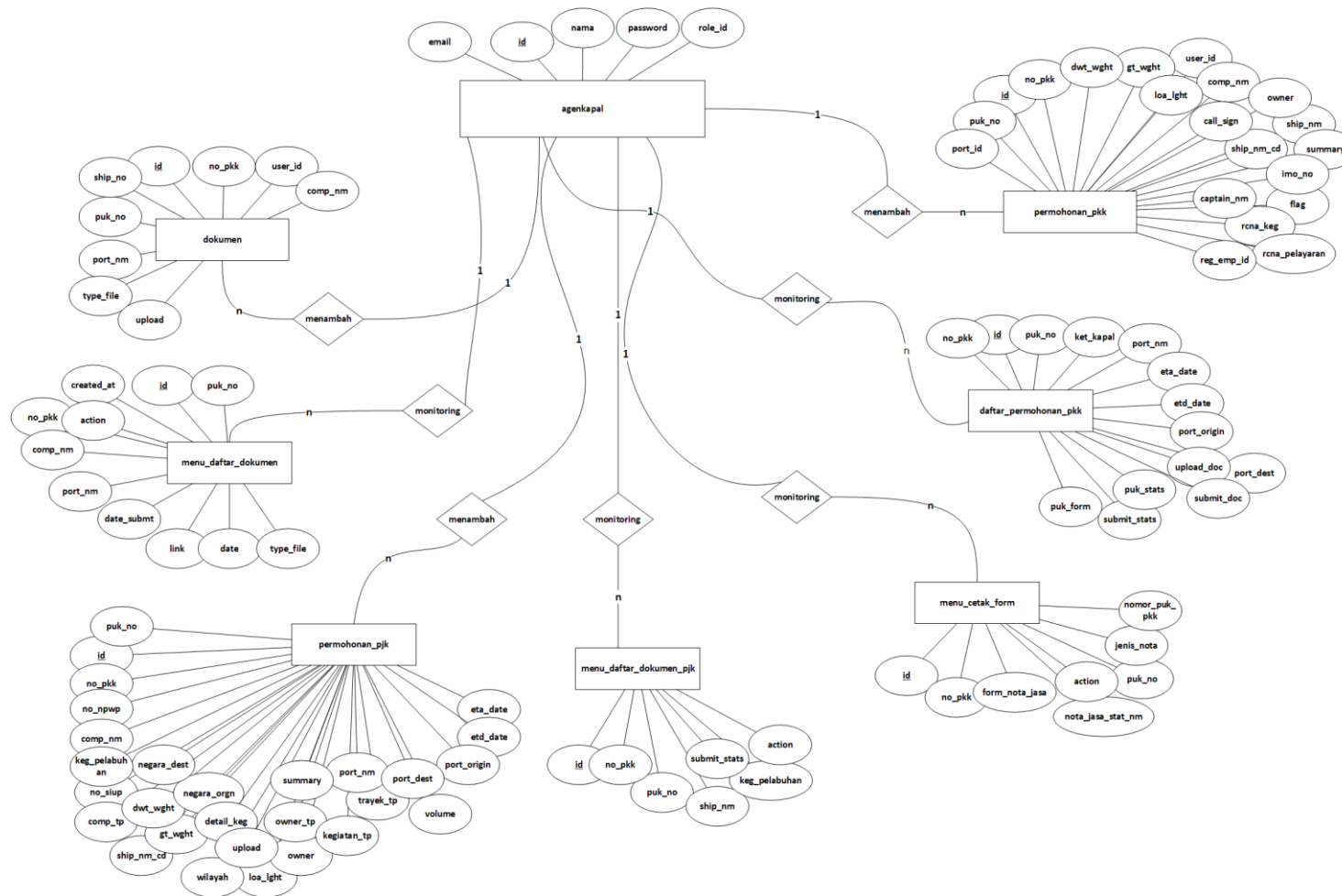
Pada Gambar 3.17 diatas merupakan rancangan form yang dibuat dalam bentuk *class diagram*. Adapun interaksi antar kelas atau objek telah dibahas pada *sequence diagram*.

3.2.3.6 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 3.18 merupakan desain *ERD* untuk *user* Admin, sedangkan Gambar 3.19 merupakan desain *ERD* untuk *user* Agen Kapal. Desain *ERD* menjelaskan bahwa *ERD* memiliki berbagai macam entitas dalam membangun aplikasi B-SIMS.

3.2.3.7 Rancangan Antarmuka

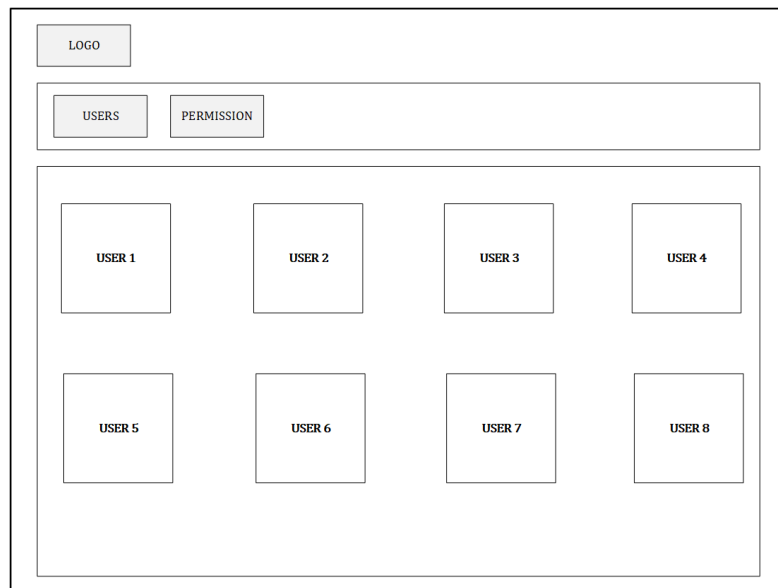
1. Form Login *Admin* dan Agen Kapal

The wireframe shows a login interface divided into two main sections. The left section contains a box for 'LOGO/TUDUL' at the top and a larger box for 'DESCRIPTION APP' below it. The right section contains input fields for 'EMAIL' and 'PASSWORD', a link for 'LUPA KATA SANDI' (Forgot Password), a 'SIGN IN' button, and a link for 'BUAT AKUN BARU' (Create New Account).

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.

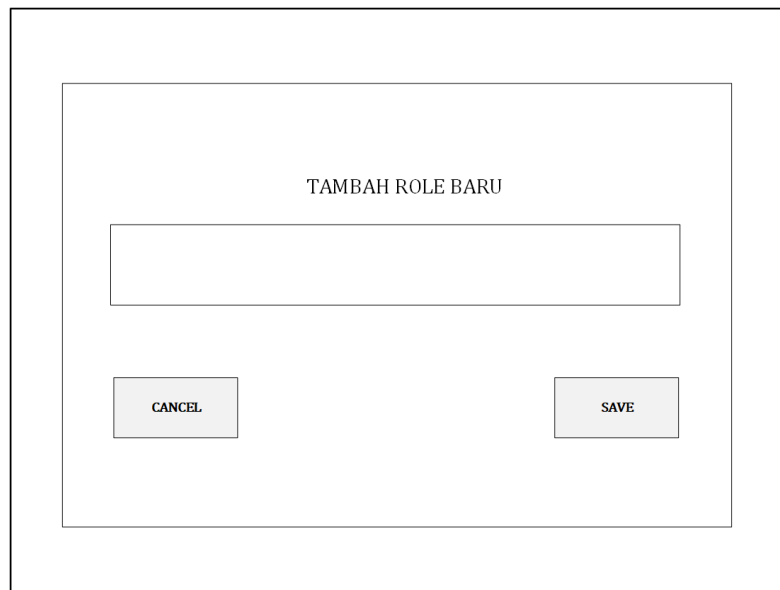
2. Menu Utama *Admin*



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*



TAMBAH ROLE BARU

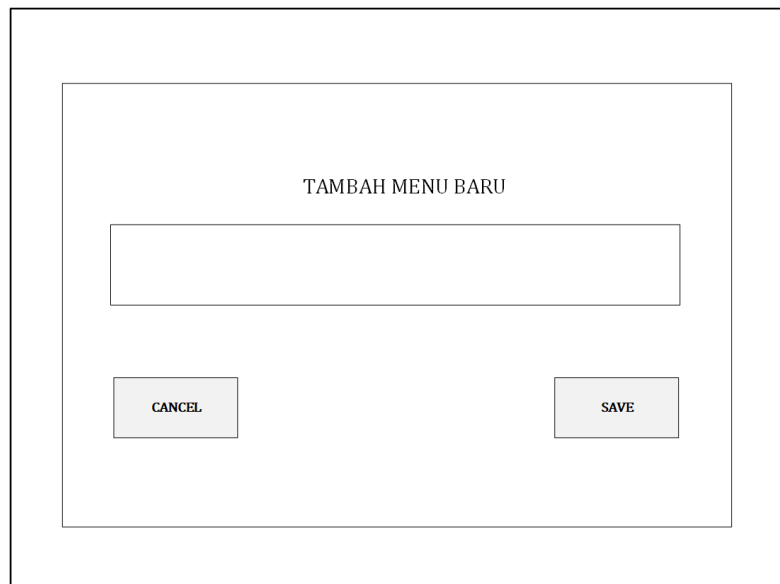
CANCEL

SAVE

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

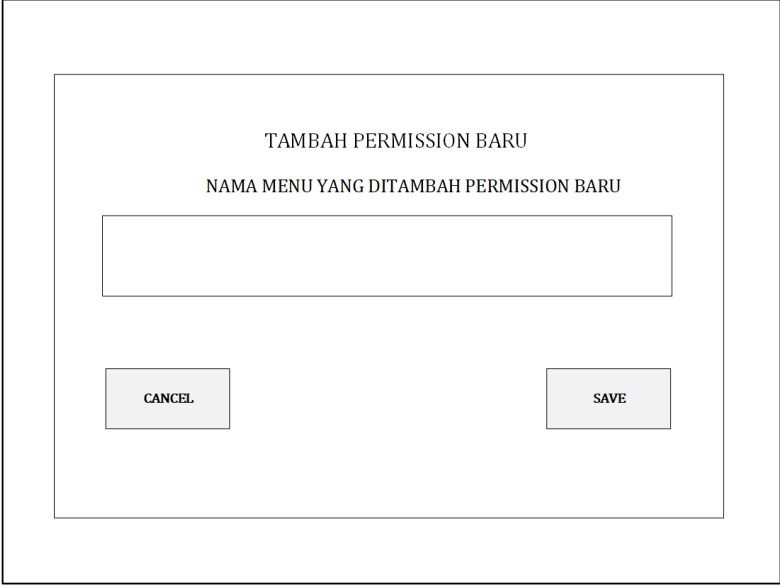
SAVE

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*



The image shows a web form titled "TAMBAH PERMISSION BARU" (Add New Permission). Below the title is a label "NAMA MENU YANG DITAMBAH PERMISSION BARU" (Name of the menu to be added permission). Underneath this label is a large, empty rectangular text input field. At the bottom of the form, there are two buttons: "CANCEL" on the left and "SAVE" on the right. The entire form is enclosed in a light gray border.

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.

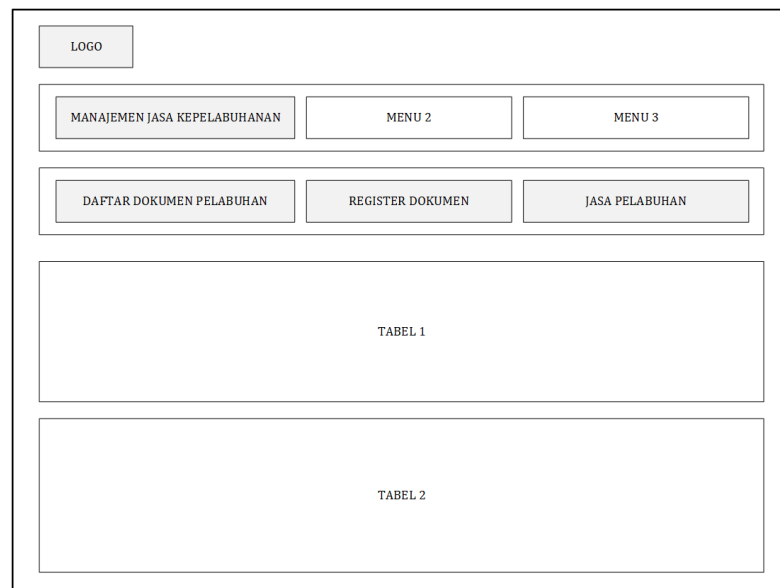
7. Menu Utama Agen Kapal

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.

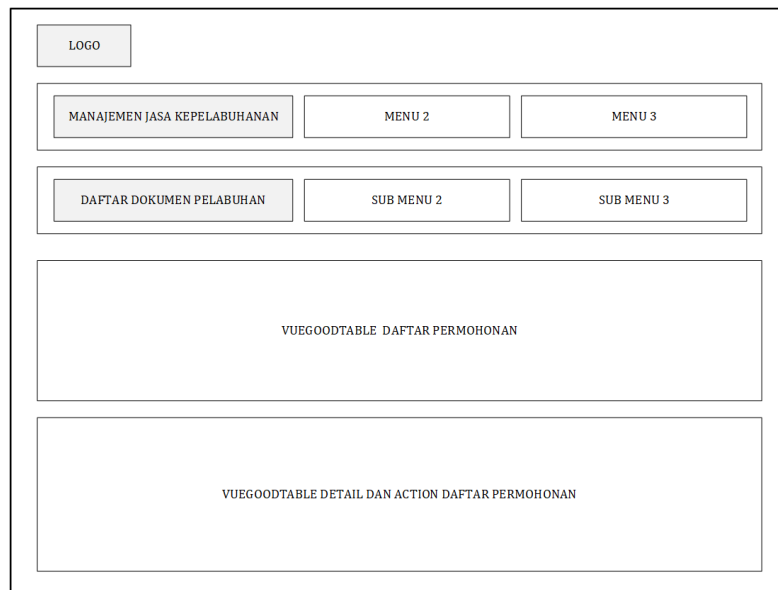
8. Menu Utama Manajemen Jasa Kepelabuhanan



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.

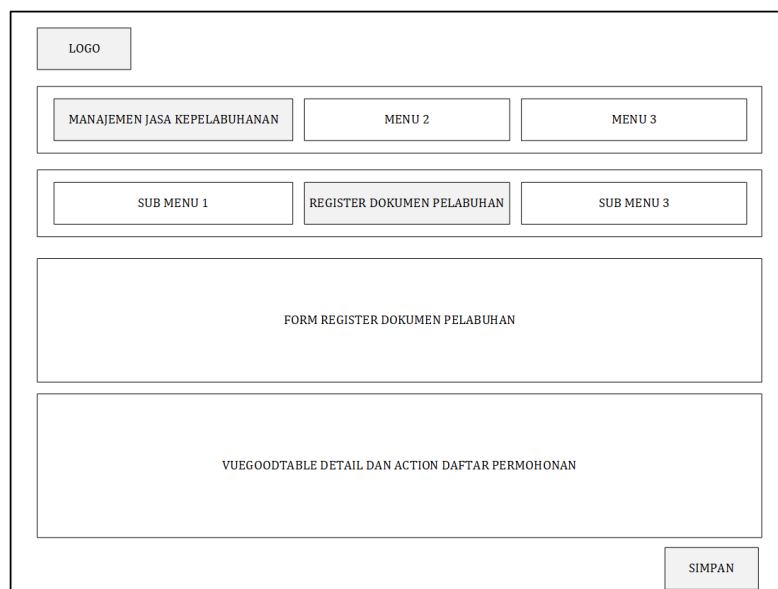
9. Menu Utama Daftar Dokumen Pelabuhan



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.

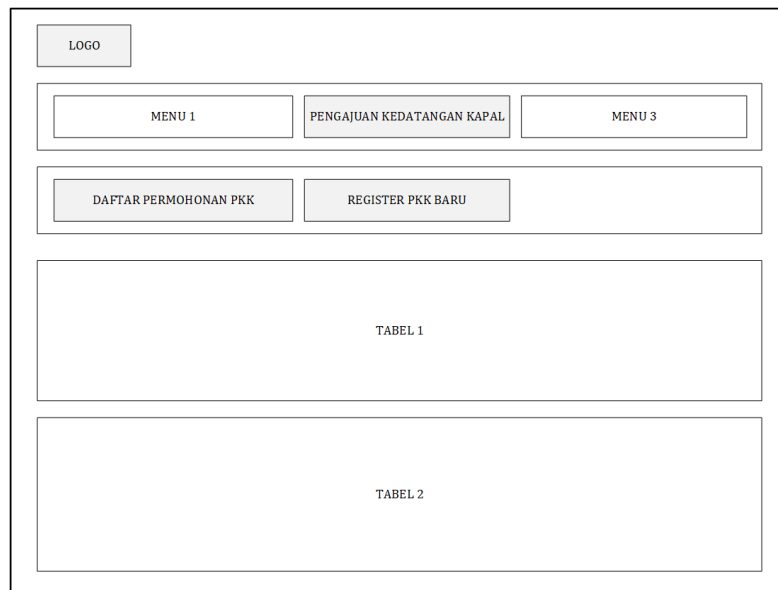
11. Menu Utama Cetak Form

The diagram illustrates the layout of the 'Menu Utama Cetak Form' (Main Menu Print Form). It features a top navigation bar with a 'LOGO' button. Below this, a horizontal menu contains three items: 'MANAJEMEN JASA KEPELABUHAN', 'MENU 2', and 'MENU 3'. A secondary horizontal menu below it includes 'SUB MENU 1', 'SUB MENU 2', and 'JASA PELABUHAN'. The main content area is divided into several sections: a large search box labeled 'FORM Pencarian NO PKK/PUK', a 'TAMBAH JASA BARU' (Add New Service) button, and a large table area labeled 'VUEGOODTABLE ACTION DAN DETAIL UNTUK CETAK FORM' (VueGoodTable Action and Detail for Print 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.

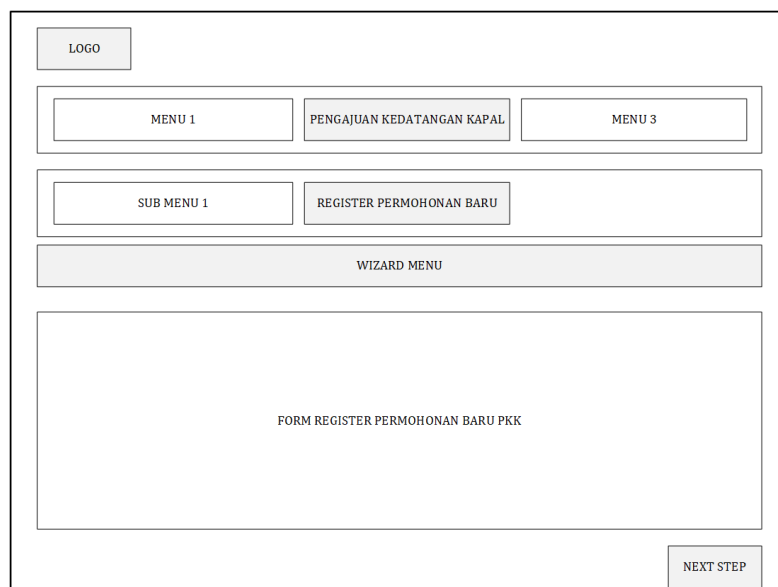
13. Menu Utama Daftar Permohonan PKK



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.

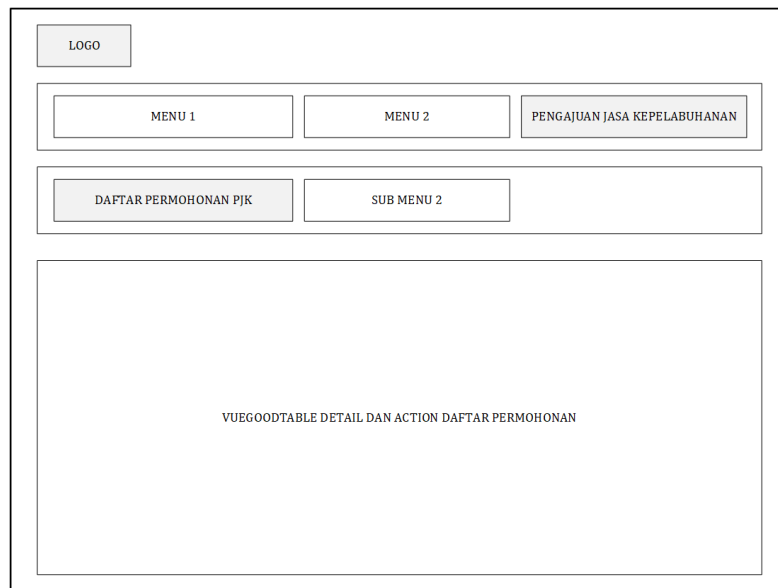
15. Menu Utama Pengajuan Jasa Kepelabuhanan/PJK

The diagram illustrates the layout of the 'Menu Utama Pengajuan Jasa Kepelabuhanan/PJK'. It features a header section with a 'LOGO' box on the left and a navigation bar containing three buttons: 'MENU 1', 'MENU 2', and 'PENGAJUAN JASA KEPELABUHANAN'. Below the navigation bar, there are two main action buttons: 'DAFTAR PERMOHONAN PJK' and 'REGISTER PJK BARU'. The bottom portion of the menu is occupied by a large rectangular area labeled 'TABEL 1', which likely displays a list of submitted requests or services.

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.

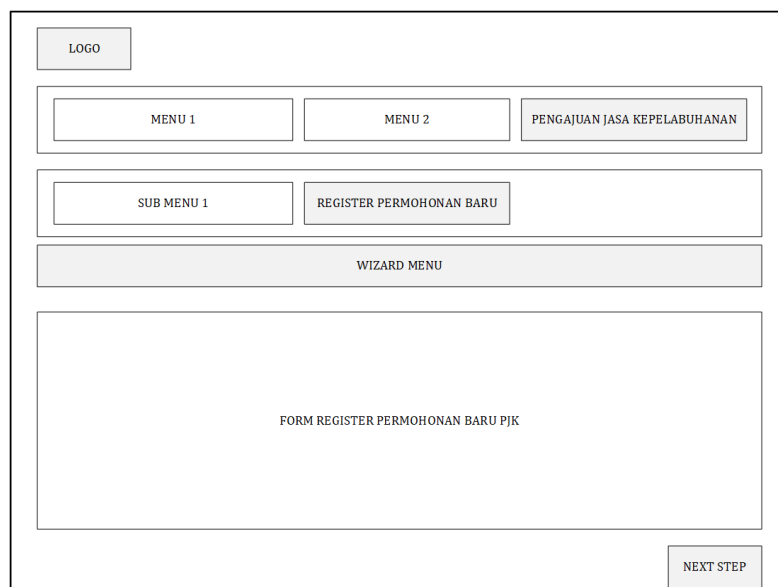
16. Menu Utama Daftar Permohonan PJK



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.

3.2.4 Implementasi Aplikasi B-SIMS

Tahapan implementasi yang dilakukan yaitu: meliputi pembuatan *database*, pembuatan aplikasi berdasarkan desain sistem yang telah disusun sebelumnya. Berdasarkan hasil analisis data diketahui bahwa untuk pembuatan aplikasi B-SIMS berbasis web aplikasi membutuhkan basis data sebagai penyimpanan data pengguna.

Pembuatan kode program aplikasi ini dilakukan dengan menggunakan perangkat lunak yaitu: Visual Studio Code dengan bahasa pemrograman PHP dan Javascript, dan menggunakan perangkat lunak git yang digunakan untuk meletakkan aplikasi di server.

3.2.5 Pengujian Aplikasi B-SIMS

Tahapan pengujian sistem aplikasi dilakukan dengan melakukan pengujian fungsional dari setiap bagian modul program apakah telah berjalan sesuai dengan fungsi sistem yang dirancang. Pengujian yang menggunakan teknik *blackbox testing* yang berfokus pada logika, fungsionalitas, sehingga dapat meminimalisir *error* dan memastikan keluaran aplikasi yang dihasilkan sesuai dengan yang diharapkan.

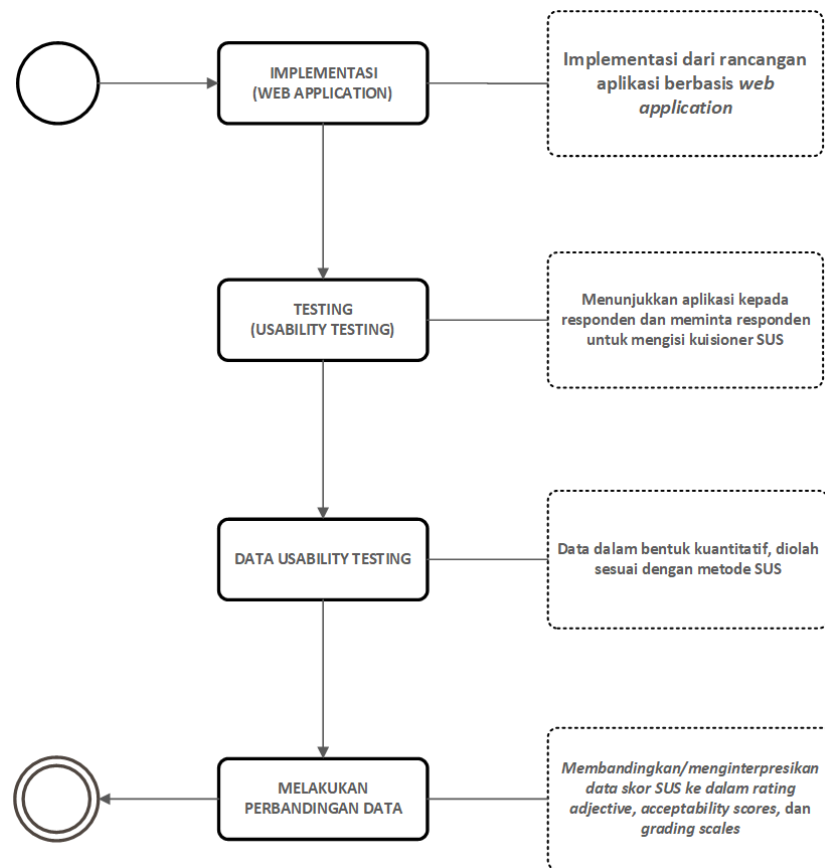
3.2.6 Pemeliharaan Aplikasi B-SIMS

Tahapan pemeliharaan berfungsi selama sistem beroperasi diperlukan karena perubahan bisnis atau lingkungan, atau adanya permintaan kebutuhan baru. Pemeliharaan sistem tetap diperlukan jika mungkin sistem menyisakan masalah-masalah yang tidak terdeteksi selama masa pengujian sistem.

3.3 Evaluasi Sistem Aplikasi

3.3.1 Evaluasi Dengan Metode *System Usability Scale* (SUS)

Evaluasi sistem aplikasi dengan menggunakan metode SUS di terapkan pada aplikasi B-SIMS yang dipergunakan oleh pengguna untuk membantu dalam kegiatan kepelabuhanan secara *online*. Proses evaluasi SUS dijelaskan pada Gambar 3.37.



Gambar 3.37 Diagram Proses Evaluasi dengan Metode SUS

Evaluasi 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). *Usability* juga dapat berarti sesuatu yang mudah digunakan dengan baik, penggunaannya dapat digunakan untuk mengukur kualitas dari suatu produk yang berkaitan

dengan mudah atau tidaknya dalam mempelajari maupun menggunakan suatu produk. Pengujian yang dilakukan pada aplikasi B-SIMS untuk menganalisis tingkat efektivitas dan efisiensi suatu aplikasi dengan menunjukkan bahwa pada kelompok responden tidak mengalami kegagalan dalam menyelesaikan tugas dengan kategori waktu sangat cepat. Sedangkan hasil analisis tingkat kepuasan pengguna menunjukkan bahwa aplikasi dapat diterima, serta menghasilkan rekomendasi perbaikan pada tampilan *interface* dan konten agar lebih memudahkan *user* dalam menggunakan aplikasi (V & H, 2020). Pengujian dengan menggunakan metode *System Usability Scale* (SUS) merupakan pengujian dengan cara melibatkan pengguna akhir (*end user*) dalam proses pengerjaannya dan proses pengujian serta perhitungan lebih rumit namun dapat dilakukan dengan jumlah sampel yang sedikit (I, P, & R, 2015).

Adapun pengujian usability testing yang dilakukan dengan menggunakan metode perhitungan Skala Likert telah dijelaskan pada bagian 2.10 dalam laporan ini.

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 2.2.

Kemudian dalam melakukan pengujian aplikasi, akan diberikan kepada 30 responden (Roscoe, 1975), ukuran minimal responden untuk mendapatkan hasil yang baik. Menurut (Sugiyono, 2010) “Populasi adalah wilayah generalisasi yang terdiri atas obyek/subyek yang mempunyai kualitas dan karakteristik tertentu yang diterapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya”. Populasi penelitian ini adalah admin agen kapal yang terdaftar di *database* BP Batam dan pegawai tetap di lingkungan BP Batam pada satuan kerja Bidang TIK IT Centre BP Batam dan Badan Usaha Pelabuhan BP Batam pada Tabel 3.16.

Tabel 3.15 Populasi Penelitian

(Sumber: SDM BP Batam dan Db Agen Kapal BUP)

No	Jenis	Jumlah
1	Admin Agen Kapal	30 Orang
2	Pegawai Tetap	6 Orang
Jumlah Populasi		36 Orang

Andi Supangat (Supangat, 2007) mengemukakan, Sampel adalah “bagian dari populasi (contoh) untuk dijadikan sebagai bahan penelaahan dengan harapan contoh yang diambil dari populasi tersebut dapat mewakili (*representative*) terhadap populasinya”, oleh karena itu yang digunakan dalam penelitian ini adalah sampel total atau *sampling* jenuh yang diungkapkan oleh (Sugiyono, 2010) bahwa “sampling jenuh adalah teknik penentuan sampel bisa semua anggota populasi digunakan sebagai sampel”. Sampel dalam penelitian ini adalah admin agen kapal yang terdaftar di *database* BP Batam dan pegawai tetap di lingkungan BP Batam pada satuan kerja Bidang TIK IT Centre BP Batam yang berjumlah 36 orang.

Selama pengumpulan data dibutuhkan beberapa alat dan bahan yang telah dijabarkan pada Tabel 3.17 sebagai berikut:

Tabel 3.16 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

Jenis data yang akan digunakan dalam penelitian ini adalah data primer. Data primer adalah data yang di usahakan sendiri proses pengumpulannya oleh peneliti dimana sumber data yang langsung memberikan data kepada pengumpul data (Sugiyono, 2015). Data primer diperoleh dari menyebar kuesioner kepada admin agen kapal yang terdaftar di *database* BP Batam dan karyawan tetap BP Batam di lingkungan Bidang TIK IT Centre BP Batam yang bersedia menjadi responden dan mengisi kuesioner. Data tersebut didapatkan dari hasil interview dan dari hasil angket SUS yang dapat dilihat pada Tabel 2.2 yang sudah dibuat sesuai dengan kebutuhan penelitian yang terdapat pada Lampiran A.

Angket atau kuisisioner adalah sejumlah pertanyaan tertulis yang digunakan untuk memperoleh informasi dari responden dalam arti laporan tentang pengalaman pribadinya dari hal-hal yang ia ketahui selama menggunakan aplikasi B-SIMS. Sehingga dalam fokus utamanya untuk mengevaluasi aplikasi B-SIMS dengan menggunakan teknik SUS dan melibatkan responden sebagai pengguna yang berbagi pengalamannya menggunakan aplikasi ini, sehingga dapat diketahui tingkat kebergunaan dilihat dari aspek kemudahan, kecepatan, dan kepuasan terhadap aplikasi B-SIMS.

LAMPIRAN

Lampiran A. Kuesioner System Usability Scale

Terdapat 10 (sepuluh) pengajuan pertanyaan setelah aplikasi B-SIMS (BP Batam Seaport Informations Management System) dibuat. Dipaparkan beberapa pertanyaan mengenai nilai kebergunaan suatu website atau aplikasi. Seperti Tabel 2.2.

SYSTEM USABILITY SCALE

Nama Responden :

Jenis Kelamin :

Nama Perusahaan :

Mohon berikan tanda (✓) pada kolom alternatif jawaban yang saudara anggap paling tepat mencerminkan persepsi Bapak/Ibu. Terdapat lima (5) alternatif jawaban yang dapat digunakan untuk masing-masing pernyataan, yaitu: 1 = Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Netral, 4 = Setuju, 5 = Sangat Setuju.

No	Pernyataan	1	2	3	4	5
1	Saya berpikir akan menggunakan sistem ini lagi					
2	Saya merasa sistem ini rumit untuk digunakan (memuat banyak hal yang tidak perlu)					
3	Saya merasa sistem ini mudah digunakan					
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini					
5	Saya merasa fitur-fitur yang disediakan pada situs ini dirancang dan disiapkan dengan baik					
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)					
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat					
8	Saya merasa sistem ini membingungkan					
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini					

No	Pernyataan	1	2	3	4	5
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini					
<p>Berikan komentar tentang aplikasi B-SIMS ini. Berikan pula apa yang anda harapkan terkait dengan aplikasi B-SIMS:</p>						