

## Implementasi Metode *Waterfall* Pada Sistem Informasi Kapasitas Pengoperasian Kapal

<sup>1</sup>Dewangga Rivaldy Pratama, <sup>2</sup>Kunto Eko Susilo, <sup>3</sup>Aryo Nugroho  
<sup>1,2,3</sup> Universitas Narotama Surabaya

Alamat Surat

Email: [dewangga.rivaldy@gmail.com](mailto:dewangga.rivaldy@gmail.com)

### Article History:

**Diajukan:** 27 Maret 2021; **Direvisi:** 15 April 2022; **Diterima:** 25 April 2022

### ABSTRAK

Tujuan dari penelitian ini adalah untuk memudahkan *customer* dalam menyewa kapal dan melihat kapasitas kapal yang akan beroperasi. Industri pelayaran ini berfokus pada kegiatan bisnis pengiriman, pengangkutan barang, penumpang komersial dan penyewaan kapal. Perusahaan yang bergerak pada bidang ini kebanyakan berfokus pada bisnis logistik yang memastikan barang dapat terkirim sampai tujuan dengan tepat waktu walaupun menempuh jarak yang sangat jauh bahkan impor. Pada PT. LAS yang juga terjun pada bisnis ini belum memiliki sistem yang mampu menarik para *customer* secara luas. Untuk itu dibuatkanlah suatu sistem informasi yang mampu menarik banyak *customer* dengan menonjolkan fitur-fitur yang menarik dan memudahkan *customer* untuk melakukan pengiriman barang, bahkan menyewa kapal. Metode yang diterapkan dalam penelitian ini adalah metode *waterfall*. Hasil akhir dalam penggunaan metode ini berupa rancangan sistem informasi mulai dari UML (*Unified Modeling Language*), *database*, dan desain *interface* pada PT. LAS.

**Kata kunci:** Sistem Informasi, UML, *Waterfall*

### ABSTRACT

*The purpose of this study is to make it easier for customers rent a ship and ensure capacity of the ship that will operate. In shipping company focus for delivery activity, freight transport, commercial passenger, and rent of shipping. Companies engaged in this job mostly focused on logistic business to ensure that delivery on time even though it's a very long distance. PT. LAS also operate in this business don't have a system to make a easier operational in this company, it can be said this company is far behind competitors. For that reason, an information system is made overcome that problems. The method applied in this research is waterfall method. The final result for this research is information system design including a unified modeling language, database, and interface design of PT. LAS.*

**Keywords:** Information System, UML, *Waterfall*

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Seiring berkembangnya teknologi informasi, mendorong munculnya inovasi perusahaan-perusahaan untuk memanfaatkan teknologi dalam berbagai aspek. Industri pelayaran di Indonesia saat ini memiliki kemauan yang cukup signifikan. Saat ini di bidang industri pelayaran sudah berdiri banyak perusahaan yang menjadikan persaingan pada bidang ini sangat kompetitif. Di Indonesia sendiri sudah banyak perusahaan yang beroperasi dan siap bersaing untuk mencari kepuasan konsumen untuk menjadikan perusahaan tersebut menjadi yang terbaik.

Ketergantungan terhadap impor membuat industri perkapalan harus dilirik dikarenakan progres yang menjanjikan (Bachtiar et al., 2021).

Didalam industri perkapalan terdapat beberapa jenis kapal yang dioperasikan, mulai dari kapal kontener, feri, pesiar, *tanker*, *tugboat*, LCT (*Landing Craft Tank*) dan lain sebagainya. PT. Lestari Abadi Sakti adalah perusahaan perkapalan yang bergerak dalam bidang pengiriman alat berat menggunakan kapal jenis LCT. Perusahaan ini masih menyediakan informasi operasional kapal dengan cara manual yang mengharuskan konsumen datang langsung ke perusahaan atau harus melakukan pemesanan melalui perangkat telepon terlebih dahulu. Dikarenakan belum tersedianya sistem informasi menjadikan informasi seputar pengoperasian kapal di perusahaan ini menjadi terhambat yang mengakibatkan banyak konsumen yang telat mendapatkan informasi.

Dikarenakan hal tersebut, peneliti akan merancang sebuah sistem informasi berbasis web untuk memudahkan konsumen mendapatkan informasi. Tujuan dibuatkan rancangan sistem informasi ini yaitu agar konsumen dapat memonitoring seluruh operasional yang disediakan pada fitur-fitur sistem informasi tersebut. Diantaranya konsumen dapat mudah memonitoring tujuan atau rute kapal dan mengetahui berapa banyak muatan yang tersedia pada geledak kapal tersebut.

Berikut merupakan tahapan dari model *waterfall*; pertama, analisa kebutuhan adalah proses pengumpulan kebutuhan secara rinci untuk menganalisis kebutuhan apa saja yang dibutuhkan oleh peneliti untuk membuat sistem informasi. Kedua, desain adalah proses multi langkah yang hanya fokus pada desain perancangan sistem meliputi *user interface*, *flowchart*, relasi antar sistem, dan prosedur pengkodean. Ketiga, kode program adalah proses desain pada tahap sebelumnya yang diimplementasikan kedalam bentuk program, hasil dari tahap ini adalah program sistem sesuai dengan desain yang sudah dirancang. Keempat, pengujian adalah merupakan tahapan yang terakhir dari model *waterfall* ini. Tahapan ini memastikan bahwa sistem sudah siap digunakan oleh pengguna dan juga mencari sebanyak-banyaknya kesalahan pada sistem yang bertujuan untuk memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan (Nasir, 2019).

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka permasalahan yang dapat diidentifikasi dari penelitian ini adalah bagaimana sistem informasi memonitoring atau menunjukkan kepada *customer* agar dapat mengetahui aktifitas atau pengoperasian kapal terbaru dari segi kapasitas kapal maupun unit kapal mana yang akan dioperasikan, dan bagaimana metode *waterfall* dapat merancang suatu sistem informasi kapasitas pengoperasian kapal

## 1.3. Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah untuk membuat suatu sistem informasi yang dapat memonitoring kapasitas pengoperasian kapal sehingga dapat memudahkan *customer* untuk *tracking* secara *real time*.

## 2. METODE

Perencanaan alur penelitian ini merupakan serangkaian tahapan-tahapan untuk menyelesaikan penelitian ini. Pada tahapan ini metode yang digunakan adalah metode *waterfall*. Terdapat 4 tahapan yaitu analisis, desain, pengkodean, dan pengujian. Berikut merupakan tahapan-tahapan metode *waterfall*, sebagai berikut (Ario Yustin et al., 2016) :



Gambar 1. Metode *Waterfall*

### a. Analisis

Analisa kebutuhan adalah proses pengumpulan kebutuhan secara rinci untuk menganalisis kebutuhan apa saja yang dibutuhkan oleh peneliti untuk membuat sistem informasi. Analisa

kebutuhan adalah proses pengumpulan kebutuhan secara rinci untuk menganalisis kebutuhan apa saja yang dibutuhkan oleh peneliti untuk membuat sistem informasi.

b. Desain

Desain adalah proses multi langkah yang hanya fokus pada desain perancangan sistem meliputi *user interface*, *flowchart*, relasi antar sistem, dan prosedur pengkodean.

c. Pengkodean

Kode program adalah proses desain pada tahap sebelumnya yang diimplementasikan kedalam bentuk program, hasil dari tahap ini adalah program sistem sesuai dengan desain yang sudah dirancang.

d. Pengujian

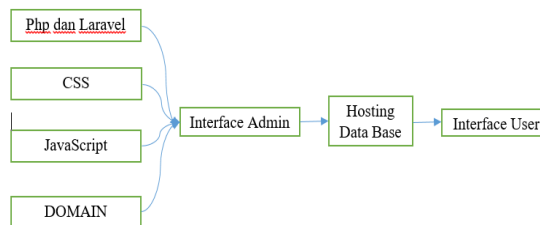
Pengujian adalah merupakan tahapan yang terakhir dari model waterfall ini. Tahapan ini memastikan bahwa sistem sudah siap digunakan oleh pengguna dan juga mencari sebanyak-banyaknya kesalahan pada sistem yang bertujuan untuk memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 2.1 Pengumpulan Data

Pada tahap ini pengumpulan data dilakukan dengan cara observasi langsung di kantor pusat PT LAS. Data yang sudah dikumpulkan menjadi suatu *requirements* yang akan dijadikan acuan pembuatan sistem. Data yang diambil untuk dijadikan *requirements* meliputi fungsional sistem kepada *customer*.

## 2.2 Diagram Blok Sistem

Diagram blok (Mulyana & Wijaya, 2018) merupakan representasi bergambar singkatan dari suatu hubungan sebab dan akibat atau akibat *input* dan *output* dari sistem fisik. Diagram blok ini merupakan sistem yang menjelaskan antar sub sistem. Didalam blok terdapat suatu sistem yang akan dibuat dan sistem lain terdapat diluarannya.



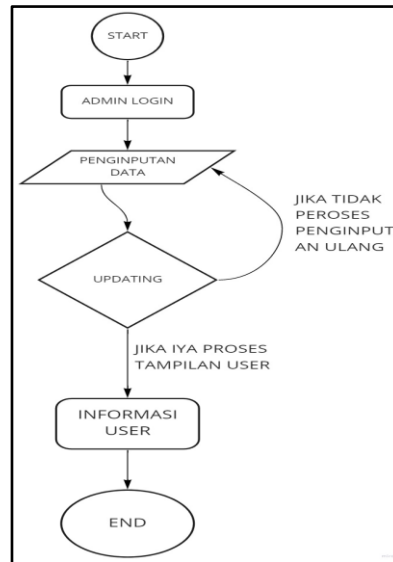
Gambar 2. Diagram Blok Sistem

Pada gambar 2 merupakan diagram blok sistem *interface*. Admin sebagai penginputan data pada *server* menggunakan bahasa pemrograman PHP dan Laravel dan di desain menggunakan bahasa CSS agar tampilan lebih menarik terhadap *user*. Kemudian data di inputkan kedalam *server database*. *User* dapat melihat tampilan yang sudah didesain dan juga diperbarui kepada *server* dan dapat dipantau melalui domain yang sudah disediakan, *user* disini hanya dapat melihat dan juga menerima notifikasi pembaruan yang sudah dirubah oleh admin.

## 3. PERANCANGAN SISTEM

Perancangan sistem (Fadlurrohimi et al., 2020) yang akan digunakan dalam penelitian ini adalah menggunakan pendekatan diagram berbasis obyek berupa diagram alir, *flowchart*, *Unified Modeling Language* (UML) yaitu berupa *usecase diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*. Diagram alir aplikasi merupakan gambaran dari langkah-langkah penelitian. Terdapat lima tahapan yang ada pada diagram alir aplikasi ini. Dimulai dari *login*, penginputan data, pengolahan data, dan informasi *user*. Tahap-tahap ini dibutuhkan peneliti agar hasil akhir dari penelitian ini sesuai dengan kebutuhan pengguna. Berikut merupakan tahap-tahap diagram alir aplikasi yang dapat dilihat pada Gambar 3. Perancangan sistem yang akan digunakan dalam penelitian ini adalah menggunakan

pendekatan diagram berbasis obyek berupa diagram alir, *flowchart*, *Unified Modeling Language* (UML) yaitu berupa *usecase diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*. Diagram alir aplikasi merupakan gambaran dari langkah-langkah penelitian. Terdapat lima tahapan yang ada pada diagram alir aplikasi ini. Dimulai dari *login*, penginputan data, pengolahan data, dan informasi *user*. Tahap-tahap ini dibutuhkan peneliti agar hasil akhir dari penelitian ini sesuai dengan kebutuhan pengguna. Berikut merupakan tahap-tahap diagram alir aplikasi yang dapat dilihat pada Gambar 3.



Gambar 3. Diagram Alir Aplikasi

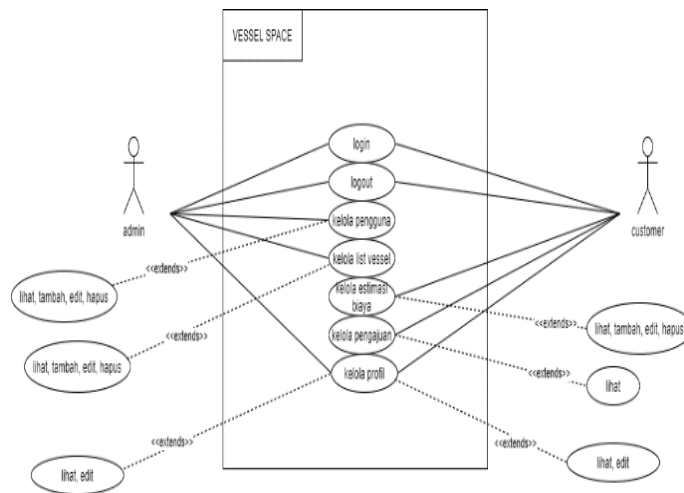
Pada Gambar 3 merupakan diagram alir aplikasi pada web Sistem informasi muatan kapal untuk memberi informasi kapasitas kapal menggunakan metode *waterfall*. Admin melakukan *login* akun lalu menginput data muatan *user* kemudian diproses oleh sistem. Kemudian diinformasikan kepada *user* untuk dapat dipantau muatan yang akan dibawa kapal dari perjalanan hingga muatan yang akan diturunkan oleh kapal. Pada Gambar 3 merupakan diagram alir aplikasi pada web Sistem informasi muatan kapal untuk memberi informasi kapasitas kapal menggunakan metode *waterfall*. Admin melakukan *login* akun lalu menginput data muatan *user* kemudian diproses oleh sistem. Kemudian diinformasikan kepada *user* untuk dapat dipantau muatan yang akan dibawa kapal dari perjalanan hingga muatan yang akan diturunkan oleh kapal.

## 4. HASIL DAN PEMBAHASAN

### 4.1. *Unifield Modeling Language* (UML)

*Unifield Modeling Language* (UML)(M Teguh Prihandoyo, 2018) merupakan salah satu permodelan visual yang digunakan pada penelitian ini. UML merupakan model perancangan visual pembuatan *software* yang berorientasi pada objek. Perancangan model UML pada penelitian ini menggunakan 4 diagram. Pada model UML ini terdapat sebuah rancangan atau *blueprint* dimana terdapat sebuah *usecase diagram*, *activity diagram*, *sequence diagram*, *class diagram*. *Unifield Modeling Language* (UML) merupakan salah satu permodelan visual yang digunakan pada penelitian ini. UML merupakan model perancangan visual pembuatan *software* yang berorientasi pada objek. Perancangan model UML pada penelitian ini menggunakan 4 diagram. Pada model UML ini terdapat sebuah rancangan atau *blueprint* dimana terdapat sebuah *usecase diagram*, *activity diagram*, *sequence diagram*, *class diagram*.

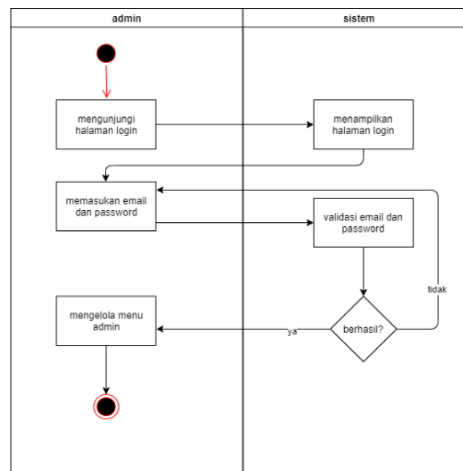
a. *Usecase Diagram*



Gambar 4. *Usecase Diagram*

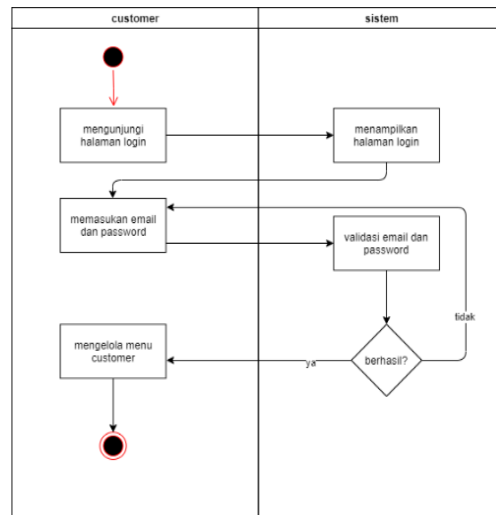
Pada Gambar 4 merupakan rancangan desain sebuah *usecase diagram*. Pada *usecase diagram* tersebut terdapat 2 aktor yaitu admin dan customer. Masing-masing dari actor yang ada mempunyai hak akses tersendiri. Dijelaskan bahwa admin dapat mengelola data, sedangkan actor customer hanya dapat login, logout, kelola estimasi biaya, kelola pengajuan sewa, dan kelola profil.

b. *Activity Diagram*



Gambar 5. *Activity Diagram Admin*

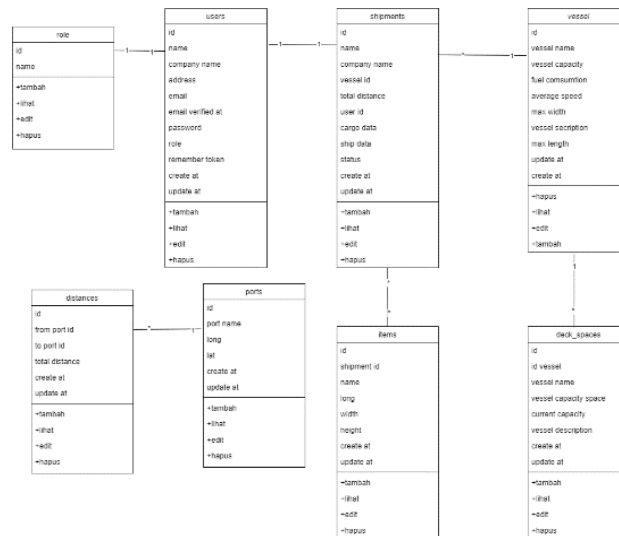
Pada Gambar 5 merupakan rancangan desain *activity diagram* admin. Langkah awal yang dijelaskan yaitu admin dapat login dengan mengunjungi halaman login, setelah itu admin memasukkan id dan juga password. Tahap selanjutnya yaitu validasi email dan password. Jika email dan password tervalidasi maka admin dapat masuk ke halaman utama dan bisa melakukan operasional sistem. Admin dapat mengelola berbagai bentuk data pada sistem ini, mulai dari edit data kapal terbaru, stok kapasitas kapal, stok tiket penumpang komersial, stok kapal yang dapat dipesan. Admin juga mendapatkan hak akses dalam melihat dan edit profil customer.



Gambar 6. Activity Diagram Customer

Pada Gambar 6 merupakan rancangan desain *activity diagram customer*. Langkah awal yang dijelaskan yaitu customer dapat *login* dengan mengunjungi halaman *login*, setelah itu admin memasukkan *id* dan juga *password*. Tahap selanjutnya yaitu validasi *email* dan *password*. Jika *email* dan *password* tervalidasi maka *customer* dapat masuk ke halaman utama dan bisa melakukan kelola profil, pemesanan kapal, dan juga pemesanan penumpang komersial.

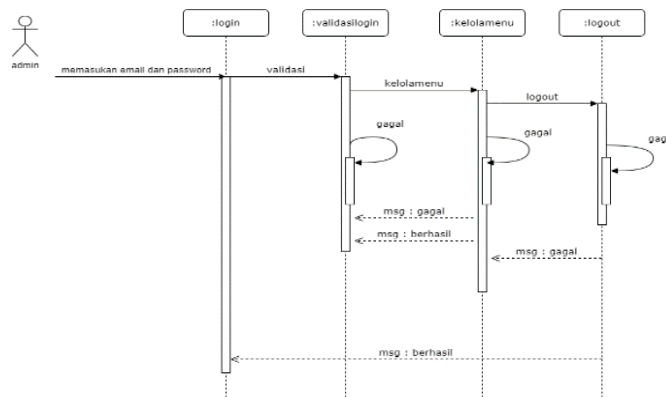
c. Class Diagram



Gambar 7. Class Diagram

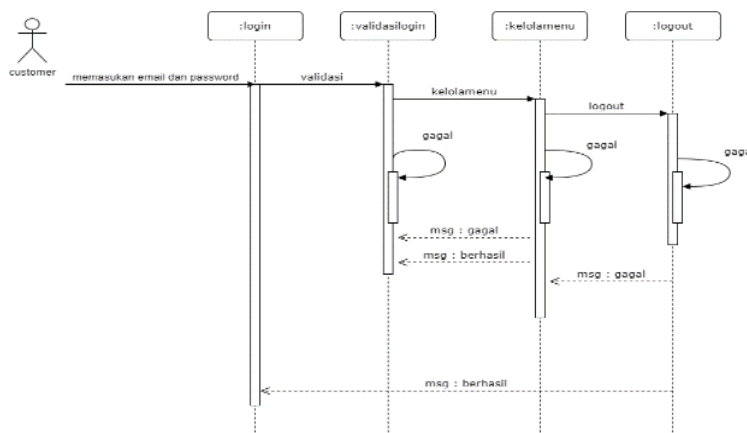
Pada Gambar 7 merupakan rancangan desain *class diagram*. *Class diagram* ini menggambarkan kelas-kelas yang terbentuk dalam kasus pembuatan sistem informasi kapasitas pengoperasionalan kapal ini. Seluruh desain rancangan *class diagram* akan menjadi acuan untuk mengimplementasikan ke dalam sebuah bahasa pemrograman.

d. *Sequence Diagram*



Gambar 8. *Sequence Diagram Admin*

Pada Gambar 8 merupakan rancangan desain *sequence diagram* admin. Dari rancangan diatas dapat dilihat bahwa yang menjadi aktor adalah admin. Terdapat 1 aktor dan 4 objek, yaitu admin sebagai *actor* dan *login*, *validasi login*, *kelola menu*, *logout* sebagai objek. Langkah pertama admin kan masuk ke layer *login* dengan menggunakan *id* dan *password*. Setelah melakukan *input id* dan *password* akan dilakukan *validasi login*. Setelah berhasil melewati *validasi login* admin dapat melakukan pengelolaan menu. Pengelolaan menu ini bertujuan untuk memberitahu *customer* apa saja yang disajikan pada sistem ini. Terdapat beberapa kasus jika admin gagal masuk pada pengelolaan menu yaitu gagalnya *validasi id* dan *password*. *Validasi* bertujuan untuk memberikan akses kepada admin untuk melakukan pengelolaan sistem.



Gambar 9. *Sequence Diagram Customer*

Pada Gambar 9 merupakan rancangan desain *sequence diagram customer*. Terdapat 1 aktor dan 4 objek yaitu *customer* sebagai *actor* dan *login*, *validasi login*, *kelola profil*, dan *logout*. Langkah pertama *customer* yaitu *login* dengan memasukkan *id* dan *password*, setelah itu dilakukan *validasi login*. Jika berhasil *login* maka *customer* dapat mengelola profil ataupun melakukan transaksi pemesanan. Dan jika *validasi* gagal maka perlu dilakukan pengolahan biodata pada fitur *kelola profil* yang ada pada tampilan *customer*.

**4.2. Rancangan Database**

*Database* (Syakti et al., 2019) merupakan proses permodelan dan pembuatan sebagai media penyimpanan. Pada pembangunan sistem diperlukan perancangan *database* guna menampung data pada tabel-tabel *database* yang bertujuan mengolah data menjadi informasi. Pembentukan *database* dilakukan sesuai dengan kebutuhan informasi dari hasil analisis kebutuhan. Manfaat dari

*database* sendiri yaitu penyimpanan data yang saling berkaitan dan perangkat lunak seharusnya berhubungan langsung dengan manajemen basis data.

Tabel 1. *Deck Space*

No	Field	Length	Type Data	Description
1	<i>Id</i>	10	<i>Integer</i>	<i>Primary Key</i>
2	<i>Id Vessel</i>	20	<i>Bigint</i>	<i>Foreign Key</i>
3	<i>Vessel name</i>	255	<i>Varchar</i>	
4	<i>Vessel capacity space</i>	20	<i>Bigint</i>	
5	<i>Current capacity</i>	20	<i>Bigint</i>	
6	<i>Vessel description</i>	255	<i>Varchar</i>	
7	<i>Create at</i>	-	<i>Timestamp</i>	
8	<i>Update at</i>	-	<i>Timestamp</i>	

Tabel 1 berisi perancangan *database* dimana terdapat 8 entitas yaitu *id*, *id vessel*, *vessel name*, *vessel capacity space*, *current capacity*, *vessel description*, *cerate at*, *update at*. Masing-masing entitas memiliki panjang *font* atau angka dan jenis tipe data yang berbeda-beda. Pada kolom *field* nomor 1, *id* memiliki ukuran panjang angka 10 dengan tipe data *integer* dan memiliki *primary key* pada kolom *description*. *Primary key* digunakan sebagai indentitas suatu baris data. *Foreign key* berguna sebagai relasi antar tabel, baik dua tabel maupun lebih. Secara sederhana memiliki hubungan antara tabel master dengan tabel turunannya.

Tabel 2. *Distances*

No	Field	Length	Type Data	Description
1	<i>Id</i>	20	<i>Bigint</i>	<i>Primary Key</i>
2	<i>From port id</i>	20	<i>Bigint</i>	<i>Foreign Key</i>
3	<i>To port id</i>	20	<i>Bigint</i>	<i>Foreign Key</i>
4	<i>Total distance</i>		<i>Double</i>	
5	<i>Create at</i>		<i>Timestamp</i>	
6	<i>Update at</i>		<i>Timestamp</i>	

Tabel 2 berisi perancangan *database* dimana terdapat 6 entitas yaitu *id*, *from port id*, *total distances*, *cerate at*, *update at*. Masing-masing entitas memiliki panjang *font* atau angka dan jenis tipe data yang berbeda-beda. Pada kolom *field* nomor 1, *id* memiliki ukuran panjang angka 20 dengan tipe data *bigint* dan memiliki *primary key* pada kolom deskripsi. *Primary key* digunakan sebagai indentitas suatu baris data.

Tabel 3. *Ports*

No	Field	Length	Type Data	Description
1	<i>Id</i>	20	<i>Bigint</i>	<i>Primary Key</i>
2	<i>Port name</i>	255	<i>Varchar</i>	
3	<i>Long</i>	255	<i>Varchar</i>	



4	<i>Lat</i>	255	<i>Varchar</i>	
5	<i>Create at</i>	-	<i>Timestamp</i>	
6	<i>Update at</i>	-	<i>Timestamp</i>	

Tabel 3 berisi perancangan *database* dimana terdapat 6 entitas yaitu *id*, *port name*, *long*, *lat*, *cerate at*, *update at*. Masing-masing entitas memiliki panjang *font* atau angka dan jenis tipe data yang berbeda-beda. Pada kolom field nomor 1, *id* memiliki ukuran panjang angka 20 dengan tipe data *bigint* dan memiliki *primary key* pada kolom deskripsi. *Primary key* digunakan sebagai indentitas. Terdapat 3 jenis tipe data yaitu *bigint*, *varchar*, dan *timestamp*. *Bigint* merupakan tipe data *integer* atau angka, *varchar* merupakan jenis data *string* dan nilai maksimalnya adalah 255 karakter.

Tabel 4. *Role*

No	Field	Length	Type Data	Description
1	<i>Id</i>	11	<i>Integer</i>	<i>Primary Key</i>
2	<i>Name</i>	255	<i>Varchar</i>	

Tabel 4 berisi perancangan *database* dimana terdapat 2 entitas yaitu *id* dan *name*. Pada kolom *field* nomor 1, *id* memiliki ukuran panjang angka 11 dengan tipe data *integer* dan memiliki *primary key* pada kolom deskripsi. *Primary key* digunakan sebagai indentitas suatu baris data. Sedangkan *name* pada nomor 2 memiliki panjang huruf 255 dengan tipe data *varchar*.

Tabel 5. *Shipments*

No	Field	Length	Type Data	Description
1	<i>Id</i>	10	<i>Integer</i>	<i>Primary Key</i>
2	<i>Company name</i>	255	<i>Varchar</i>	
3	<i>Vessel id</i>	20	<i>Bigint</i>	
4	<i>Total distance</i>	-	<i>Double</i>	
5	<i>User id</i>	-	<i>Bigint</i>	
6	<i>Cargo data</i>	-	<i>Longtext</i>	
7	<i>Ship data</i>	-	<i>Longtext</i>	
8	<i>Price</i>	-	<i>Double</i>	
9	<i>Status</i>	-	<i>Enum</i>	<i>Pending, approval</i>
10	<i>Create at</i>	-	<i>Timestamp</i>	
11	<i>Update at</i>	-	<i>Timestamp</i>	

Tabel 5 berisi perancangan *database shipments* dimana terdapat 11 entitas yaitu *id*, *company name*, *id vessel*, *total distance*, *user id*, *cargo data*, *ship data*, *price*, *status*, *cerate at*, *update at*. Masing-masing entitas memiliki panjang *font* atau angka dan jenis tipe data yang berbeda-beda. Pada kolom *field* nomor 1, *id* memiliki ukuran panjang angka 10 bertipe data *integer* dan memiliki *primary key* pada kolom deskripsi. *Primary key* digunakan sebagai indentitas suatu baris data. Tabel 5 berisi perancangan *database shipments* dimana terdapat 11 entitas yaitu *id*, *company name*, *id vessel*, *total distance*, *user id*, *cargo data*, *ship data*, *price*, *status*, *cerate at*, *update at*. Masing-masing entitas memiliki panjang *font* atau angka dan jenis tipe data yang berbeda-beda. Pada

kolom *field* nomor 1, *id* memiliki ukuran panjang angka 10 bertipe data *integer* dan memiliki *primary key* pada kolom deskripsi. *Primary key* digunakan sebagai identitas suatu baris data.

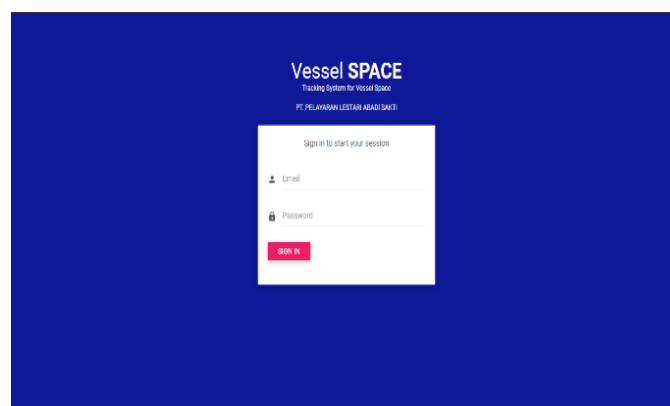
Tabel 6. *User*

No	Field	Length	Type Data	Description
1	<i>Id</i>	10	<i>Bigint</i>	<i>Primary Key</i>
2	<i>Name</i>	255	<i>Varchar</i>	
3	<i>Company name</i>	255	<i>Varchar</i>	
4	<i>Addres</i>	-	<i>Text</i>	
5	<i>Email</i>	255	<i>Varchart</i>	
6	<i>Email verified</i>	-	<i>Timestamp</i>	
7	<i>Password</i>	255	<i>Varchar</i>	
8	<i>Role</i>	11	<i>Integer</i>	<i>Foreign Key</i>
9	<i>Remember token</i>	100	<i>Varchar</i>	
10	<i>Create at</i>	-	<i>Timestamp</i>	
11	<i>Update at</i>	-	<i>Timestamp</i>	

Tabel 6 berisi perancangan *database user* dimana terdapat 11 entitas yaitu *id, name, company name, addres, email, email verified, password, role, remember token, cerate at, update at*. Pada kolom *field* nomor 1, *id* memiliki ukuran panjang angka 10 bertipe data *bigint*. Memiliki *primary key* pada kolom *description*. *Primary key* digunakan sebagai identitas suatu baris data.

### 4.3. Hasil Tampilan Aplikasi

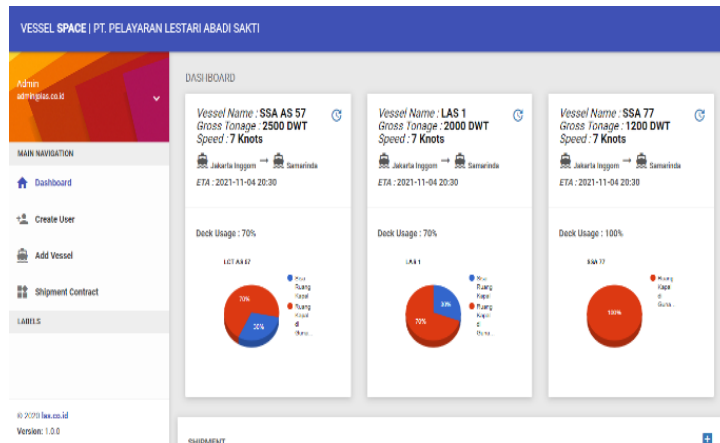
Setelah mendapatkan hasil desain rancangan sistem, tahap selanjutnya yaitu implementasi kedalam sistem. Hasil dari pembangunan sistem yang telah dirancang memiliki halaman-halaman yang siap digunakan. Berikut merupakan *interface* sistem yang dapat dilihat pada gambar-gambar di bawah ini. Setelah mendapatkan hasil desain rancangan sistem, tahap selanjutnya yaitu implementasi kedalam sistem. Hasil dari pembangunan sistem yang telah dirancang memiliki halaman-halaman yang siap digunakan. Berikut merupakan *interface* sistem yang dapat dilihat pada gambar-gambar di bawah ini.



Gambar 10. Tampilan *Login*

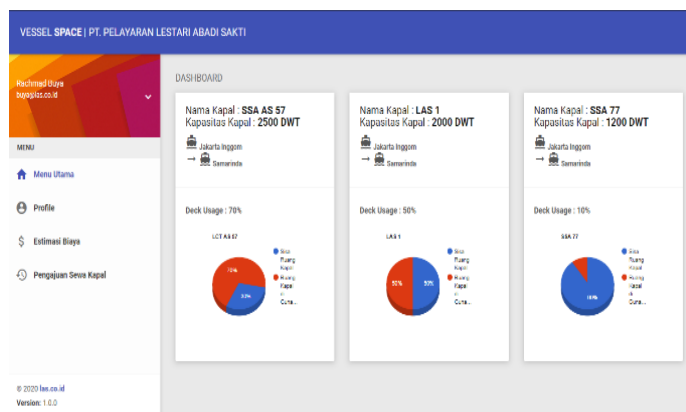
Gambar 10 merupakan tampilan *login* untuk pengguna dengan memasukkan *username* dan *password* agar dapat masuk ke tampilan selanjutnya. Hasil dari halaman *login* untuk memasukkan

email dan password admin dan juga konsumen yang nantinya untuk masuk ke halaman dashboard utama admin dan konsumen. Akan tetapi halaman dashboard admin dan konsumen akan berbeda.



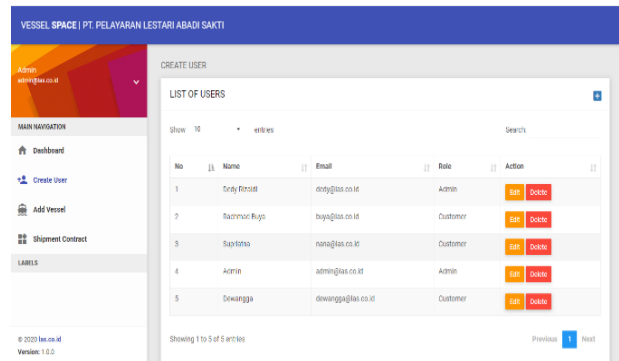
Gambar 11. Tampilan Dashboard

Gambar 11 merupakan tampilan dashboard admin. Terdapat beberapa fitur yaitu menu create user, add vessel, dan shipment contract. Tampilan ini merupakan halaman awal setelah melakukan login. Tampilan ini didesain untuk memenuhi kebutuhan admin. Gambar 11 merupakan tampilan dashboard admin. Terdapat beberapa fitur yaitu menu create user, add vessel, dan shipment contract. Tampilan ini merupakan halaman awal setelah melakukan login. Tampilan ini didesain untuk memenuhi kebutuhan admin.



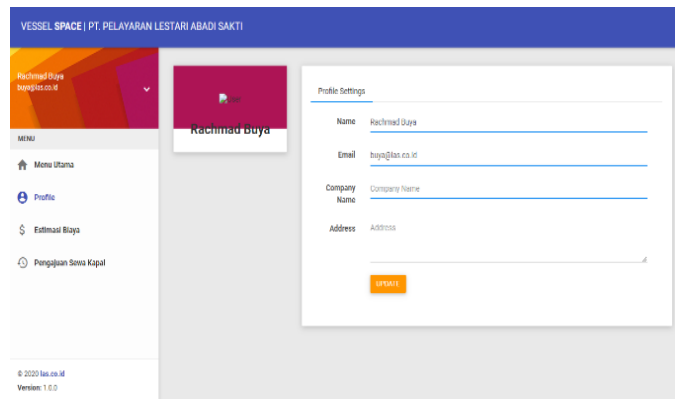
Gambar 12. Tampilan Dashboard Customer

Gambar 12 merupakan tampilan dashboard customer. Terdapat beberapa fitur yaitu menu profil, estimasi biaya, dan pengajuan sewa kapal. Tampilan ini merupakan halaman awal setelah melakukan login untuk customer. Tampilan ini didesain untuk memenuhi kebutuhan customer. Customer dapat mengelola profil atau biodata yang ada pada fitur kelola profil. Fitur-fitur lain seperti pemesanan kapal, stok kapasitas kapal juga dapat diakses oleh pengguna. Gambar 11 merupakan tampilan dashboard customer. Terdapat beberapa fitur yaitu menu profil, estimasi biaya, dan pengajuan sewa kapal. Tampilan ini merupakan halaman awal setelah melakukan login untuk customer. Tampilan ini didesain untuk memenuhi kebutuhan customer. Customer dapat mengelola profil atau biodata yang ada pada fitur kelola profil. Fitur-fitur lain seperti pemesanan kapal, stok kapasitas kapal juga dapat diakses oleh pengguna.



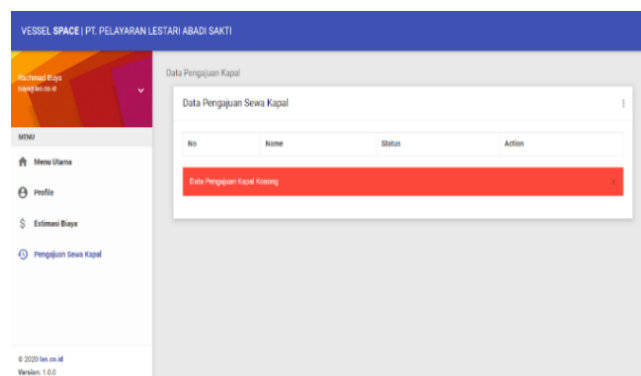
Gambar 13. Tampilan Data Pengguna

Gambar 13 merupakan tampilan data pengguna sistem. Terdapat beberapa fitur yaitu menu profil, estimasi biaya, dan pengajuan sewa kapal. Kelebihan pada tampilan ini yaitu pengguna bisa mengedit isi biodata maupun identitas pengguna. Tampilan ini merupakan halaman awal setelah melakukan *login*. Tampilan ini didesain untuk memenuhi kebutuhan admin.



Gambar 14. Tampilan Profil

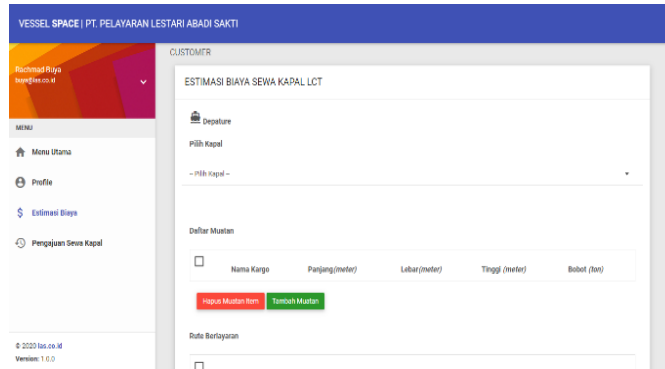
Gambar 14 merupakan tampilan profil. Pengguna dapat mengedit foto, email, alamat, dan perusahaan asal. Fitur yang ada sudah dirancang sesuai dengan kebutuhan *customer* sehingga memudahkan dalam mengakses sistem informasi ini. Tampilan yang mudah dipahami akan sangat berpengaruh untuk *customer* dari segi operasional. Hasil dari halaman profil yang telah didesain yang berguna untuk melengkapi data pengguna. Gambar 13 merupakan tampilan profil. Pengguna dapat mengedit foto, email, alamat, dan perusahaan asal. Fitur yang ada sudah dirancang sesuai dengan kebutuhan *customer* sehingga memudahkan dalam mengakses sistem informasi ini. Tampilan yang mudah dipahami akan sangat berpengaruh untuk *customer* dalam segi operasional. Hasil dari halaman profil yang telah didesain yang berguna untuk melengkapi data pengguna.



Gambar 15. Tampilan Sewa Kapal

Gambar 15 merupakan tampilan data untuk *customer* menyewa kapal. Pada tampilan sewa kapal ini *customer* dapat mengetahui status pemesanan dan juga *action* dari perusahaan yang

menyediakan kapal untuk disewakan. Tampilan tersebut adalah data yang sudah valid yang digunakan untuk mendapatkan validasi dari perusahaan penyedia sewa kapal.



Gambar 16. Tampilan Estimasi Biaya Sewa

Gambar 16 merupakan tampilan estimasi biaya sewa kapal. Tampilan ini memiliki fitur estimasi biaya yang memungkinkan *customer* memilih biaya yang sesuai. Dengan adanya fitur ini dapat memudahkan *customer* dalam hal pengelolaan biaya perusahaan dan juga meminimalisir kerugian akibat melonjaknya biaya sewa. Halaman estimasi biaya yang telah didesain yang berguna untuk mengetahui estimasi biaya yang nantinya juga akan dibuat untuk pengajuan penyewaan. Gambar 16 merupakan tampilan estimasi biaya sewa kapal. Tampilan ini memiliki fitur estimasi biaya yang memungkinkan *customer* memilih biaya yang sesuai. Dengan adanya fitur ini dapat memudahkan *customer* dalam hal pengelolaan biaya perusahaan dan juga meminimalisir kerugian akibat melonjaknya biaya sewa. Halaman estimasi biaya yang telah didesain yang berguna untuk mengetahui estimasi biaya yang nantinya juga akan dibuat untuk pengajuan penyewaan.

#### 4.4. Testing

Pada tahapan ini pengujian dilakukan sebanyak dua kali, yaitu pengujian *front end* dan *back end*. Pengujian *front end* merupakan antarmuka pengguna yang menyajikan suatu fungsi dari sebuah *website* serta berinteraksi secara langsung dengan pengguna. Pengujian *front end* (Chastro & Darmawan, 2020) merupakan antarmuka pengguna yang menyajikan suatu fungsi dari sebuah *website* serta berinteraksi secara langsung dengan pengguna. Pengujian *back end* merupakan yang berfokus pada pengkodean sistem yang memungkinkan suatu fitur sistem berjalan sesuai keinginan. Pada tahapan ini pengujian dilakukan sebanyak dua kali, yaitu pengujian *front end* dan *back end*. Pengujian *front end* merupakan antarmuka pengguna yang menyajikan suatu fungsi dari sebuah *website* serta berinteraksi secara langsung dengan pengguna. Pengujian *front end* merupakan antarmuka pengguna yang menyajikan suatu fungsi dari sebuah *website* serta berinteraksi secara langsung dengan pengguna. Pengujian *back end* merupakan yang berfokus pada pengkodean sistem yang memungkinkan suatu fitur sistem berjalan sesuai keinginan.

Tabel 7. Pengujian *Front End*

Partisipan	Registrasi	Login	Nilai Kesuksesan
1	Ok	Ok	100
2	Ok	Ok	100

Tabel 7 merupakan hasil pengujian *front end*. Hasil yang dijelaskan menunjukkan 2 partisipan. Terdapat 2 skenario yang akan diujikan pada testing *front end* kali ini, yaitu registrasi dan *login*. Hasil dari pengujian ini menunjukkan berhasil registrasi dan *login* dengan nilai 100. Pengujian *front end* ini bertujuan untuk menguji kesesuaian kinerja dari sistem. Sistem diuji

keseluruhan untuk memastikan bahwa tampilan *interface* sesuai dengan yang diinginkan. Dan jika terdapat kesalahan dapat segera diperbaiki agar dapat digunakan sebagaimana mestinya.

## 5. SIMPULAN DAN SARAN

Berdasarkan analisis dan perancangan sistem yang telah dilakukan oleh peneliti, sistem informasi yang dibangun memiliki menu pengelolaan data vessel, data akun, data item, dan data *distance* atau jarak. Dengan kebutuhan yang telah dianalisis menghasilkan sistem informasi yang dapat mengelola aktivitas dan transaksi data pada nama lokasi penelitian. Dengan adanya sistem ini diharapkan dapat membantu pihak-pihak yang berkepentingan pada nama lokasi penelitian.

## 6. DAFTAR PUSTAKA

- Ario Yustin, J., sujaini, H., & Azhar Irwansyah, M. (2016). Rancang Bangun Aplikasi *Game* Edukasi Pembelajaran Matematika Menggunakan Construct 2. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 1(1), 422–426. <https://jurnal.untan.ac.id/index.php/justin/article/view/16354>
- Bachtiar, A. I., Marimin, M., Adrianto, L., & Bura, R. O. (2021). Strategi Peningkatan Daya Saing Industri Perkapalan (*Shipbuilding Industry*). *Jurnal Aplikasi Bisnis Dan Manajemen*, 7(1), 121–134. <https://doi.org/10.17358/jabm.7.1.121>
- Chastro, C., & Darmawan, E. (2020). Perbandingan Pengembangan *Front End* Menggunakan *Blade Template* dan *Vue Js*. *Jurnal STRATEGI-Jurnal Maranatha*, 2(2), 302–313.
- Fadlurrohimi, I., Nulhaqim, S. A., & Sulastri, S. (2020). Implementasi Program Bantuan Pangan Non Tunai (Studi Kasus Di Kota Cimahi). *Share: Social Work Journal*, 9(2), 122. <https://doi.org/10.24198/share.v9i2.20326>
- M Teguh Prihandoyo. (2018). *Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web*. *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 126–129.
- Mulyana, A., & Wijaya, H. (2018). Perancangan *E-Payment System* pada *E-Wallet* Menggunakan Kode QR Berbasis Android. *Komputika: Jurnal Sistem Komputer*, 7(2), 63–69. <https://doi.org/10.34010/komputika.v7i2.1511>
- Nasir, M. (2019). Kinerja Karyawan Ditinjau dari Pelatihan dan Motivasi pada PT. Industri Kapal Indonesia (Persero) Makassar. *Jurnal Manajemen Bisnis*, 6(2), 85–94. <https://doi.org/10.33096/jmb.v6i2.401>
- Syakti, F., Hutrianto, & Ependi, U. (2019). Desain Dan Implementasi Pemodelan *Database*. *Desain Dan Implementasi Pemodelan Database Industri Kecil Menengah Kota Palembang*, 19, 70–78.