

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Penelitian sebelumnya pada tahun 2019 yang berjudul “Analisa Pengaruh Perhitungan Jumlah Muatan Tongkang SS2409 Tanpa Memperlihatkan Variabel Koreksi Berdasarkan Kaidah *Draft survey*” Penelitian ini membahas tentang pengaruh variabel koreksi terhadap pelaksanaan *draft survey* pada kapal Tongkang SS2409, pada penelitian ini membuktikan bahwa variabel koreksi sangat berpengaruh terhadap perhitungan *draft survey* [1].

Penelitian lainnya pada tahun 2020 yang berjudul “Komparasi Perhitungan *Draft survey* Antara Metode Konvensional Dengan Metode Microsoft Excel Dalam Menghitung Berat Muatan Pada Kapal.” Penelitian diatas membahas terkait perbandingan antara perhitungan *draft survey* dengan metode konvensional atau manual dengan perhitungan *draft survey* dengan metode microsoft excel. Pada penelitian diatas membuktikan bahwa pekerjaan yang dilakukan dengan menggunakan microsoft excel lebih baik dibandingkan dengan metode konvensional [2].

Kemudian penelitian lainnya pada tahun 2020 yang berjudul “Manajemen *Draft survey* di MV.Sri Wandari Indah” Pada penelitian tersebut membahas tentang manajemen *draft survey* pada kapal MV.Sri Wandari Indah. Hal ini membuktikan bahwa penggunaan *draft survey* dapat meringankan beban pekerjaan pada kapal MV.Sri Wandari Indah [3].

Selanjutnya penelitian pada tahun 2019 yang berjudul “Evaluasi Draught Survey Batubara di Atas Tongkang dan Vessel PT Adaro Indonesia Site Kelanis” Pada penelitian diatas membahas tentang evaluasi untuk *draft survey* batubara di kapal tongkang dan vessel PT Adaro Indonesia Site Kelanis. Hal ini membuktikan bahwa perlunya ketelitian dalam mengerjakan *draft survey* [4].

Berdasarkan dari observasi yang dilakukan dan dari perbandingan dari penelitian lain, maka pada penelitian ini dimaksudkan untuk meneliti lebih dalam tentang perhitungan *draft survey* dengan mengembangkan sebuah aplikasi perhitungan *draft survey* berbasis website. Aplikasi yang dikembangkan terkait dengan aplikasi berbasis website sebagai alat untuk

menghitung *draft survey* yang dimana pada aplikasi nantinya akan dapat menghitung perhitungan *draft survey* dengan cepat dan tepat tanpa harus menuliskan kembali rumus yang dibutuhkan untuk perhitungan *draft survey*. Pada penelitian ini dibuatkanlah terobosan terbaru jika dibandingkan dengan penelitian sebelumnya, yaitu mengimplementasikan aplikasi perhitungan muatan kapal kedalam sebuah website.

## **2.2. Landasan Teori**

Penelitian ini diperlukan adanya teori-teori yang mendasar untuk menunjang proses penelitian ini. Teori-teori yang digunakan dalam penelitian ini adalah:

### **2.2.1. Sistem Informasi**

Menurut Turban, Rainer dan Potter sistem informasi adalah Proses yang menjalankan fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan tertentu. Sistem informasi tidak harus terkomputerisasi, walaupun kebanyakan memang terkomputerisasi [5].

Untuk dapat lebih memahami secara detail, membangun dan mengembangkan suatu sistem yang baik, maka perlu memiliki karakteristik seperti berikut :

#### a. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

#### b. Batasan Sistem (boundary)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

#### c. Lingkungan Luar Sistem (environment)

Lingkungan luar sistem adalah diluar batas sistem dari sistem yang mempengaruhi operasi sistem.

#### d. Penghubung Sistem (interface)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya.

#### e. Masukkan Sistem (input)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (maintenance input), dan masukkan sinyal (signal input). Maintenance input adalah energi yang dimasukkan agar

sistem dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran.

f. Keluaran Sistem (output)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (goal) atau sasaran (objective). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.

Dari uraian teori diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan prosedur organisasi yang saling berinteraksi dan merupakan gabungan empat bagian utama yaitu perangkat lunak, perangkat keras, infrastruktur dan sumber daya manusia yang terlatih yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan untuk mencapai suatu tujuan.

### **2.2.2. Rekayasa Perangkat Lunak**

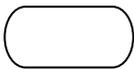
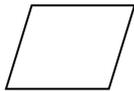
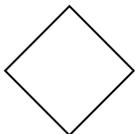
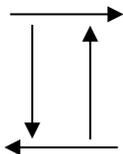
Rekayasa perangkat lunak adalah sebuah studi dan aplikasi dengan menggunakan pendekatan yang bersifat kuantifiabel, disiplin, dan sistematis kepada pengembangnya, memiliki operasi dan pemeliharaan perangkat lunak yang merupakan aplikasi. Pengimplementasian rekayasa perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut :

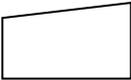
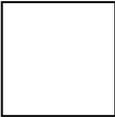
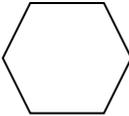
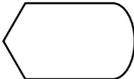
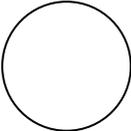
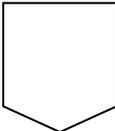
a. *Flowchart*

*Flowchart* adalah sebuah skema dengan simbol-simbol tertentu yang menjabarkan rangkaian dari proses dengan detail serta jalinan antara suatu proses (instruksi) dengan proses yang lainnya dalam sebuah program. Diagram alur bisa menggambarkan dengan jelas, jalannya sebuah algoritma yakni bagaimana menjalankan sebuah rangkaian aktivitas dengan logis dan sistematis [11].

Seperti yang telah disebutkan, *flowchart* memiliki simbol-simbol tertentu untuk menjabarkan rangkaian. Berikut adalah simbol-simbol pada *flowchart* seperti pada Tabel 2.1.

**Tabel 2. 1** Simbol-simbol *flowchart* sistem

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program.
2.		<i>Input</i> <i>Output</i>	Memasukkan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya.
3.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
4.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. simbol arus ini sering disebut dengan <i>connecting line</i> .
5.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
6.		<i>Manual</i> <i>Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.

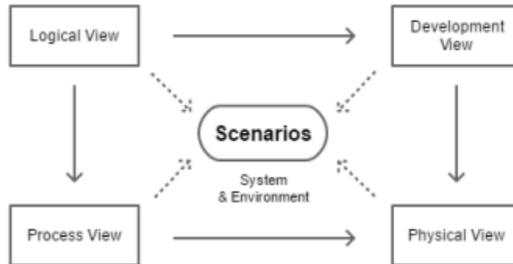
No.	Simbol	Nama	Keterangan
7.		<i>Manual Input</i>	Memasukkan data secara manual <i>on-line keyboard</i> .
8.		<i>Process</i>	Simbol yang menyatakan suatu proses yang dilakukan komputer.
9.		<i>Preparation</i>	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.
10.		<i>Display</i>	Simbol yang menyatakan peralatan output yang digunakan.
11.		<i>On-Page Refenece</i>	Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.
12.		<i>Off-Page Reference</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda.

No.	Simbol	Nama	Keterangan
13.		<i>Storage Data</i>	Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.

b. *Unified Modelling Language (UML)*

Menurut Munawar *Unified Modelling Language (UML)* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem berorientasi object. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lainnya. UML dibangun atas model 4+1 view. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam 5 view dimana salah satu diantaranya scenario. Scenario ini memegang peran khusus untuk mengintegrasikan content ke view yang lain. Kelima view tersebut berhubungan dengan diagram yang dideskripsikan di UML. Setiap view berhubungan dengan perspektif tertentu dimana sistem akan diuji. View yang berbeda akan menekankan pada aspek yang berbeda dari sistem yang mewakili ketertarikan sekelompok stakeholder tertentu [13].

Penjelasan mengenai model 4+1 view UML dapat dilihat pada gambar dibawah.



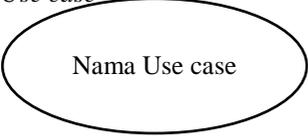
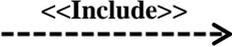
**Gambar 2. 1** Model 4+1 View

Dalam implementasinya digunakan 3 view karena menggunakan apa yg dipelajari semasa perkuliahan, yaitu *Scenario ( Use Case )*, *Logical View ( Class Diagram )* dan *Process view (Sequence Diagram)*.

- *Use Case Diagram*

Menurut Munawar *Use case diagram* adalah salah satu jenis diagram pada UML yang menggambarkan interaksi antara *sistem* dan *actor*. *Use case diagram* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user* (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai [13]. Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan use case diagram dapat dilihat pada Tabel 2.2.

**Tabel 2. 2** Simbol Use Case [13].

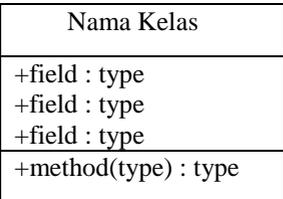
Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama use case.</p>
<p><i>Actor</i></p> 	<p>Merupakan abstraksi dari orang dan system yang lain yang mengaktifkan fungsi dari target system.</p>
<p><i>Association</i></p> 	<p>Merupakan garis antara obyek satu dengan yang lainnya.</p>
<p><i>Extend</i></p> 	<p>Merupakan suatu garis yang berfungsi menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case yang lainnya jika kondisi atau syarat tertentu dipenuhi.</p>
<p><i>Generalization</i></p> 	<p>Merupakan suatu garis yang berfungsi untuk mewariskan struktur data dan obyek induk kepada obyek anak yang dituju.</p>
<p><i>Include</i></p> 	<p>Merupakan suatu garis yang berfungsi menggambarkan suatu use case yang merupakan fungsionalitas dari use case yang lainnya.</p>
<p><i>System</i></p> 	<p>Merupakan bagian dari usecase yang menspesifikasikan sistem secara terbatas.</p>

- *Class Diagram*

Menurut Munawar *Class diagram* adalah salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. *Class diagram* menggambarkan *atribut*, *operation* dan juga *constraint* yang terjadi pada sistem. *Class diagram* banyak digunakan dalam pemodelan sistem *Object Oriented* karena mereka adalah satu-satunya diagram UML yang dapat dipetakan langsung dengan bahasa berorientasi obyek [13].

Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan *class diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2. 3** Simbol Class Diagram [13].

Simbol	Deskripsi
<p><i>Class</i></p> 	Merupakan kumpulan obyek yang memiliki attribute dan operation yang sama. Attribute, merupakan property dari sebuah class. Operation merupakan sesuatu yang bisa dilakukan oleh sebuah class.
<p><i>Composition</i></p> 	Merupakan sebuah tipe agregasi yang kuat dimana bagian dari obyek tergantung pada keseluruhan obyek.
<p><i>Directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity

<p><i>Generalization</i></p> 	<p>Merupakan suatu garis yang berfungsi untuk mewariskan stuktur data dan obyek induk kepada obyek anak yang dituju.</p>
<p><i>Dependency</i></p> 	<p>Merupakan sebuah garis yang berfungsi menunjukkan operasi pada suatu class yang menggunakan class yang lain.</p>
<p><i>Aggregation</i></p> 	<p>Merupakan suatu garis yang menangani obyek dimana salah satunya adalah bagian dari yang lain.</p>

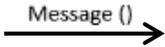
- *Sequence Diagram*

Menurut Munawar *Sequence diagram* adalah diagram dalam UML yang digunakan untuk menggambarkan perilaku pada suatu *scenario*. Diagram ini menunjukkan contoh obyek dan pesan yang diletakkan antar obyek di dalam use case [13].

Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan sequence diagram dapat dilihat pada Tabel 2.4.

**Tabel 2. 4** Simbol Sequence Diagram [13].

Simbol	Keterangan
<p><i>LifeLine</i></p> 	<p>Objek <i>entity</i>, antarmuka yang saling berinteraksi.</p>
<p><i>Actor</i></p> 	<p>Menggambarkan <i>user</i> atau pengguna.</p>

<p><i>Message</i></p> 	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi tentang aktivitas yang terjadi.
<p>Boundary</p> 	Menggambarkan sebuah <i>form</i> .
<p><i>Control Class</i></p> 	Menghubungkan <i>boundary</i> dengan Tabel.
<p><i>Entity Class</i></p> 	Menggambarkan hubungan kegiatan yang akan dilakukan

c. *Prototyping*

*Prototyping* perangkat lunak adalah metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*). Tujuannya adalah mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biayanya menjadi lebih rendah. Ciri dari metode ini adalah pengembang dan pelanggan dapat melihat dan melakukan pengerjaan dengan bagian dari sistem komputer dari sejak awal proses pengembangan. Adapun fase-fase dalam *prototyping* adalah sebagai berikut :

1. Analisa Kebutuhan

Di tahap ini dilakukan identifikasi *software* dan semua kebutuhan sistem yang akan dibuat.

2. Membangun *prototyping*

Pembangunan *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian *input* dan format *output*.

3. Evaluasi *prototyping*

Evaluasi ini dilakukan untuk mengetahui apakah *prototyping* sudah sesuai dengan harapan pelanggan. Jika belum sesuai maka kembali ke fase analisa kebutuhan.

#### 4. Mengkodekan sistem

Pada tahap ini *prototyping* yang sudah disetujui akan diubah ke dalam bahasa pemrograman berupa *website* dengan *framework laravel*.

#### 5. Menguji sistem

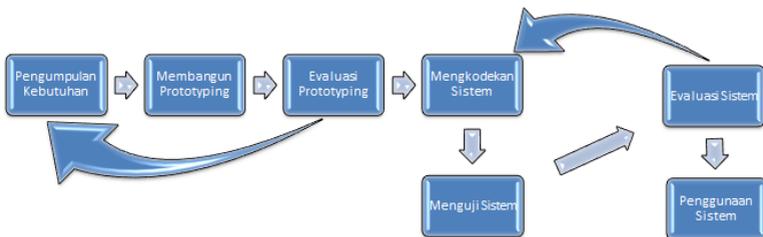
Di tahap ini dilakukan untuk menguji sistem perangkat lunak yang sudah dibuat.

#### 6. Evaluasi Sistem

Perangkat lunak yang sudah siap jadi akan dievaluasi oleh pelanggan untuk mengetahui apakah sistem sesuai dengan yang diharapkan. Jika belum sesuai dengan harapan, maka akan kembali ke fase mengkodekan sistem.

#### 7. Menggunakan sistem

Perangkat lunak yang sudah diuji dan disetujui oleh pelanggan siap digunakan [14].



**Gambar 2. 2** Metode Prototyping [14].

#### d. *Testing*

Menurut Hetzel (Revisi) *Testing* adalah tiap aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari program atau sistem dan menentukan apakah telah memenuhi

kebutuhan atau hasil yang diharapkan [17]. Testing yang digunakan pada penelitian ini adalah *blackbox testing*.

Menurut Jaya *blackbox testing* adalah *testing* yang berkonsentrasi dari sisi kesesuaian perangkat lunak yang dikembangkan dengan kebutuhan pengguna yang telah didefinisikan pada saat awal perancangan. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. *Blackbox testing* dilakukan sesuai dengan item uji yang telah dirancang. Adapun hasil *blackbox testing* adalah seluruh proses pada sistem telah berjalan dengan baik. Tak seperti *white box testing* yang dipakai pada awal proses *testing*. *Black box testing* digunakan pada tahap akhir dan berfokus pada domain informasi [18].

### **2.2.3. OOP (Object Oriented Programming)**

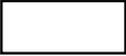
Menurut Shalahuddin Pemrograman Berorientasi Objek atau *Object Oriented Programming* (OOP) adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Pemahaman ini mengacu pada teknis dari pemrograman OOP, yakni menyusun, mengolah objek-objek berisi data dan operasi yang dibutuhkan melalui *Class* dan *Method* yang dibutuhkan dalam suatu perangkat lunak [10].

### **2.2.4. Entity Relationship Diagram (ERD)**

Dalam rekayasa perangkat lunak diperlukan adanya *Entity Relationship Diagram* (ERD). ERD merupakan model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Di dalam hubungan tersebut dinyatakan yang utama dari ERD adalah menunjuk objek data (*Entity*) dan hubungan (*Relationship*) [12].

Seperti yang telah disebutkan, ERD memiliki simbol-simbol tertentu untuk menjabarkan rangkaian. Simbol-simbol ERD dapat di lihat pada Tabel 2.5

**Tabel 2. 5** Simbol-simbol ERD

No.	Simbol	Keterangan
1.	Entitas 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer.
2.	Relasi 	Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.
3.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
4.	Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan.
5.	Atribut Multinilai 	<i>Field</i> yang butuh disimpan dalam entitas yang dapat memiliki nilai lebih dari satu.
6.	Alur 	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

### 2.2.5. *Data Base*

Menurut Hidayatullah *data base* adalah himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga dapat dimanfaatkan kembali dengan cepat dan mudah [15]. Penggunaan database sangat penting dalam sistem untuk mengorganisir informasi, data, atau file secara terpadu. Penggunaan database membantu meningkatkan efisiensi penyimpanan dan pengelolaan data. Dalam pengembangan situs web, database digunakan dalam bentuk tabel yang terdiri dari kolom dan baris yang menyimpan atribut dan nilai spesifik.

Jumlah kolom dan baris dalam sebuah database tergantung pada jumlah kategori atau jenis informasi yang perlu disimpan.

Dari pendapat ahli diatas dapat disimpulkan bahwa *data base* merupakan sistem penyimpanan yang menyimpan kumpulan informasi yang disusun sehingga mudah untuk diakses.

Menurut Subagia MySQL merupakan *software data base open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa SQL. Menurut Winarno MySQL adalah sebuah *software data base*. MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-tabel yang saling berhubungan. Keuntungan menyimpan data di database adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk tabel. Menurut Priyanto dan Hidayatullah MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan para pemrogram aplikasi web. Kelebihan dari MySQL adalah gratis, handal, selalu di-update dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di bundling dengan *web server* sehingga proses instalasinya jadi lebih mudah [16].

Dapat ditarik kesimpulan bahwa MySQL merupakan aplikasi yang digunakan untuk mengolah basis data yang banyak digunakan untuk membangun aplikasi yang menggunakan *data base* [15].

a. *Data Definition Language* (DDL)

DDL dapat di gunakan untuk mengoperasikan suatu skema struktur pada sebuah *data base*, perintah utama yang dapat di gunakan pada DDL adalah seperti *CREATE*, *RENAME*, *ALTER*, *DROP*.

b. *Data Manipulation Language* (DML)

DML adalah sekumpulan perintah yang di gunakan untuk melakukan pengelolaan database seperti membuat baru, menambah, menghapus dan menampilkan, perintah utamanya adalah seperti *UPDATE*, *DELETE*, *INSERT*, *SELECT*.

### 2.2.6. *Draft survey*

Kunci dalam perhitungan *draft survey* adalah pembacaan *draft* kapal. *Draft* kapal adalah suatu seri angka-angka yang diterapkan atau

dilekatkan (dilasi/ welding atau hanya dicat) di lambung kapal sebelah kanan dan kiri pada bagian depan (forward), dibagian tengah (midship) dan dibagian belakang (after), dimana angka-angka tersebut menunjukkan kedalaman bagian kapal yang masuk ke dalam air laut atau sungai. *draft* kapal menggunakan sistem penerapan *Metric System* dengan penerapan angka latin [7].

Dari pendapat diatas dapat disimpulkan *draft* kapal adalah suatu angka dalam satuan metric system yang menjadi acuan surveyor untuk menghitung *draft survey*. Angka *draft* dapat dilihat pada lambung kapal sebelah kanan dan kiri pada bagian depan (forward), dibagian tengah (midship) dan dibagian belakang (after), dimana angka-angka tersebut menunjukkan kedalaman bagian kapal yang masuk ke dalam air laut atau sungai.

Sedangkan pengertian *draft survey* sendiri adalah survey 'sebelum dan sesudah', yang menentukan, dengan pengukuran, perpindahan kapal sebelum dan sesudah pemuatan atau pembongkaran. Perbedaan antara dua perpindahan ini adalah berat kargo yang dimuat atau dikeluarkan [7].

*Draft survey* adalah salah satu jenis pekerjaan dari Jasa Marine Survey yaitu sistem perhitungan muatan kapal berdasarkan pembacaan dan pengukuran *draft* mark kapal sebelum dan sesudah pemuatan ataupun pembongkaran dengan memperhtungkan juga perubahan berat barang-barang di atas kapal selain muatan yang mungkin terjadi selama proses operasi muat dan bongkar tersebut. Semua kapal kargo dan tongkang termasuk kapal besar seperti Mother Vessel yang memuat barang atau kargo curah seperti batubara, pasir, biji besi, nickel, steel scrap, bauksit, mangan, emas, tembaga dan barang tambang lainnya pasti menggunakan metode perhitungan *draft survey* kapal dalam transaksi bisnis dan perdagangan sebagaimana di isyaratkan oleh hukum peraturan perdagangan internasional maupun dinas perdagangan [8].

Dari pengertian diatas dapat disimpulkan, *draft survey* adalah sistem perhitungan muatan kapal yang berpacu kepada penunjukkan angka *draft* kapal pada waktu sebelum dilaksanakan kegiatan muat atau bongkar muatan dan penunjukkan *draft* kapal setelah selesai dilaksanakan kegiatan muat atau bongkar muatan, dengan diketahuinya *draft* kapal pada kedua waktu tersebut maka jumlah muatan dapat ditentukan dengan cara menghitung selisih diantara keduanya.