

## **BAB II**

### **LANDASAN TEORI**

Dalam penyusunan skripsi ini dan secara garis besar akan dijelaskan mengenai pengertian dan konsep-konsep yang digunakan dalam perancangan sistem yang akan dibuat dalam skripsi ini.

#### **2.1 Konsep Dasar Sistem**

##### **2.2.1 Definisi Sistem**

Terdapat banyak sekali definisi sistem yang dikemukakan oleh para ahli. Dalam hal ini penulis mencoba untuk mengemukakan definisi sistem yang cukup bisa diterima secara logis, yaitu: Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain.

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan untuk menyelesaikan suatu sistem tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen, menurut Tata Sutabri mendefinisikan sistem yaitu kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu <sup>[1]</sup>.

Dari definisi sistem diatas maka penulis mengambil sebuah kesimpulan bahwa, sistem adalah suatu elemen-elemen yang saling berhubungan satu sama lain untuk mencapai suatu tujuan tertentu. Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*Output*), pengolahan (*process*) dan sasaran (*Objectives*) atau tujuan (*goal*).

## **2.2 Konsep Dasar Informasi**

### **2.2.1 Pengertian Informasi**

Menurut Edhy Sutanta menyatakan, bahwa informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan <sup>[2]</sup>. Dari pengertian informasi di atas kita dapat memahami bahwa adanya informasi yang diterima telah melalui pengolahan terlebih dahulu agar tersusun dengan baik dan memiliki arti bagi penerima, yang akan menambah pengetahuan penerima untuk mengambil keputusan.

Sedangkan pengertian informasi lainnya menurut Tata Sutabri adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan <sup>[1]</sup>. Sepertinya pengertian pertama dan kedua tidak jauh berbeda dan mengandung pengertian informasi yang sama. Pada intinya menurut Raymond Mc. Jeod dan Tata Sutabri, informasi adalah data yang telah diolah yang digunakan untuk mengambil keputusan.

Menurut Jogiyanto H.M., fungsi informasi adalah untuk menambah pengetahuan dan mengurangi ketidakpastian pemakai informasi. Fungsi informasi tidak mengarahkan pengambilan keputusan mengenai apa yang harus dilakukan, tetapi untuk mengurangi keanekaragaman dan ketidakpastian yang menyebabkan diambilnya suatu keputusan yang baik<sup>[3]</sup>.

### **2.2.2 Kriteria kualitas informasi**

Suatu informasi dapat dikatakan memiliki manfaat dalam proses pengambilan keputusan apabila informasi tersebut mempunyai kualitas dan nilai. Kriteria kualitas informasi adalah:

1. Akurat

Yang berarti informasi harus tidak bias atau menyesatkan dan bebas dari kesalahan.

2. Tepat waktu

Yang berarti informasi yang sampai kepada penerima tidak boleh terlambat. Mahalnya informasi saat ini adalah karena harus cepatnya informasi tersebut

didapatkan, sehingga diperlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkannya.

### 3. Relevan

Yang berarti informasi harus mempunyai manfaat bagi pihak yang menerimanya.

Beberapa hal yang menyebabkan informasi tidak berkualitas atau tidak bernilai adalah :

- a. Metode pengukuran dan pengumpulan data yang salah.
- b. Tidak mengikuti prosedur pengolahan data yang benar.
- c. Data hilang atau tidak diolah.
- d. Kesalahan mencatat atau mengoreksi data.
- e. *File* historis/induk yang salah (atau keliru memilih *file* historis).
- f. Kesalahan dalam prosedur pengolahan (misal kesalahan program komputer).
- g. Kesalahan yang disengaja.

### 2.3 Sistem Informasi

Menurut Indrajani, Sistem Informasi adalah sekelompok elemen yang saling berhubungan atau berinteraksi hingga membentuk satu-kesatuan<sup>[4]</sup>.

Menurut Jogiyanto H.M., tujuan dari sistem informasi adalah menghasilkan informasi (*Information*) dari bentuk data yang diolah menjadi bentuk yang berguna bagi para pemakainya<sup>[3]</sup>.

Kegiatan sistem informasi meliputi:

1. *Input*, menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
2. *Proses*, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi bernilai tambah.
3. *Output*, suatu kegiatan untuk menghasilkan laporan dari kegiatan proses.
4. Penyimpanan, suatu kegiatan untuk memelihara dan penyimpanan data.
5. *Control*, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

Fokus utama sistem informasi berbasis komputer adalah untuk aplikasi perkantoran (*Office Automation*), dimana komputer memiliki porsi semakin berarti untuk aplikasi Sistem Informasi Akuntansi (SIA), Sistem Informasi Manajemen (SIM), dan *Decision Support System* (DSS).

## 2.4 Perancangan Basis Data

Menurut Tata Sutabri, perencanaan sistem informasi yang merupakan terjemahan dari *Information System Planning* (ISP), menjelaskan bagaimana menerapkan pengetahuan tentang sistem informasi ke dalam organisasi agar dapat terus maju dan eksis bila organisasi berkembang sesuai dengan teknologi dan teori organisasi *modern*. Namun, hal ini tidak berarti bahwa sistem informasi dan teknologi informasi merupakan suatu hal yang kaku. Sistem informasi dapat dibentuk sesuai dengan kebutuhan organisasi<sup>[1]</sup>.

Oleh karena itu, untuk dapat menerapkan sistem yang efektif dan efisien diperlukan perencanaan, pelaksanaan, pengaturan dan evaluasi sesuai keinginan dan nilai masing-masing organisasi. Tujuan dari sistem yang efektif dan efisien tidak lain adalah untuk mendapatkan keunggulan dalam kompetisi.

Alat-alat yang digunakan dalam tahap perancangan basis data adalah sebagai berikut:

### 2.5.1 *Entity Relationship Diagram* (ERD)

Menurut Edi Sutanta "*Entity Relationship Diagram* (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek." *Entity Relationship Diagram* (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut<sup>[2]</sup>.

1. *Entity* (entitas), menurut Edi Sutanta adalah Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data<sup>[2]</sup>.

2. *Relationship* (hubungan/ relasi), Menurut Edi Sutanta, Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda <sup>[2]</sup>.
3. *Attribute*, Menurut Edi Sutanta, Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas <sup>[2]</sup>. Setiap entitas harus memiliki atribut yang unik untuk pengidentifikasian, atribut yang dimaksud disebut dengan *key*, Jenis-jenis *key* adalah:
  - a. *Primary Key*, adalah yaitu *key* yang paling umum digunakan untuk mengidentifikasi secara unik setiap instansi dari entitas.
  - b. *Secondary Key*, adalah yaitu suatu *key* yang tidak terpilih untuk dijadikan *primary key*. Juga disebut sebagai *alternate key*.
  - c. *Composite key*, adalah suatu kelompok atribut yang dapat dengan unik mengidentifikasi suatu instansi dari suatu entitas.
  - d. *Foreign key*, adalah suatu *primary key* dari suatu entitas yang digunakan di entitas yang lain untuk mengidentifikasi instansi dari suatu hubungan (*relationship*).
4. *Cardinality*, adalah jumlah kejadian minimum dan maksimum dari satu entitas yang dihubungkan dengan kejadian yang tunggal dari entitas lain. Ada 3 (tiga) kemungkinan hubungan yang ada yaitu:
  - a. *One to one* (1:1), Jumlah kejadian adalah satu ke satu antara entitas yang saling berhubungan.
  - b. *One to many* (1:M), Jumlah kejadian adalah satu ke banyak dari satu entitas ke entitas lain yang berhubungan.
  - c. *Many to many* (M : N), adalah Hubungan banyak *entity* dengan banyak *entity*.

### 2.5.2 Transformasi ERD ke LRS

Transformasi ERD ke LRS merupakan suatu kegiatan untuk membentuk data data dari diagram hubungan entitas ke suatu LRS Diagram ERD di atas dapat ditransformasikan ke bentuk LRS.



## 1. Transformasi

Setiap *entity* akan di ubah ke bentuk sebuah kotak dengan nama *entity* berada di luar kotak dan atribut berada di dalam kotak Cardinality (Tingkat Hubungan) *entity* tersebut diantaranya adalah :

### a. *One-to-one (1:1)*

Hubungan relasi *one to one* yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas H.

### b. *One-to-many (1:M)*

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

### c. *Many-to-many (M:M)*

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak pada himpunan entitas B. *Entity* ini sangat berpengaruh pada transformasi. Karena satu sama lain akan saling berhubungan.

## 2.5.3 LRS (Logical Record Structure)

LRS dibentuk dengan nomor dari tipe *record* di gambarkan oleh kotak empat persegi panjang dengan nama yang unik. LRS juga terdiri dari hubungan antara tipe *record*. Dua metode yang dapat dikonversikan ke LRS. Metode ini dimulai dengan ERD dan langsung dikonversikan ke LRS. *Logical Record Structure* dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda LRS dengan *diagram ER* nama tipe *record* berada diluar kotak *field* tipe *record* ditempatkan. *LogicalRecord Stucture* terdiri dari link-link diantara tipe *record*. *Link* ini menunjukkan arah dari satu tipe *record* ke tipe *record* lainnya. Banyak link dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua link tipe *record*. Penggambaran LRS mulai dengan penggambaran model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan Diagram ER yang langsung dikonversikan ke LRS. LRS (*Logical Record*

*Structure*) setelah ERD ditransformasikan ke bentuk LRS, maka hasil akhir dari proses transformasi adalah sebuah diagram yang sudah dapat menggambarkan basis data yang digunakan. LRS terdiri dari *report tipe record*, yang berupa sebuah persegi dengan *field* yang di butuhkan di dalamnya LRS terdiri juga dari hubungan antar tipe *record* tersebut.

#### **2.5.4 Tabel / Relasi**

Secara umum tabel adalah koleksi objek yang terdiri dari sekumpulan elemen yang diorganisasi secara kontinue, artinya memori yang dialokasi antara satu elemen dengan elemen yang lainnya mempunyai *address* yang berurutan.

Pada tabel, pengertian perlu dipahami adalah :

1. Keseluruhan tabel (sebagai koleksi) adalah kontainer yang menampung seluruh elemen.
2. Indek tabel yang menunjukkan *address* dari sebuah elemen.
3. Elemen tabel yang dapat dipacu melalui indeknya bertipe tertentu yang sudah terdefinisi.

Seluruh elemen tabel bertipe sama. Dengan catatan, Beberapa bahasa pemrograman memungkinkan pendefinisian tabel dengan elemengenerik, tapi pada saat diinstansiasi harus diinstansiasi dengan tipe sama.

#### **2.5 Analisa dan Perancangan Berorientasi Obyek**

Analisa berorientasi objek adalah cara baru dalam memikirkan sebuah masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan penggabungan antar struktur data dan perilaku dalam sebuah entitas. Analisa berorientasi objek dimulai dengan menyatakan suatu masalah, analisis menggambarkan model situasi dari dunia nyata, menggambarkan sifat yang penting.

Model analisa adalah abstraksi yang ringkas dan tepat dari apa yang harus dilakukan oleh sistem dan bagaimana cara melakukannya. Dalam pemrograman yang berorientasi objek, objek-objek diciptakan yang tidak hanya

mencakup kode tentang data, namun juga instruksi tentang operasi yang harus ditampilkan atasnya.

Tujuan dari analisa berorientasi objek yaitu untuk menentukan kebutuhan pemakaian secara akurat. Pendekatan-pendekatan yang dipakai dalam analisa berorientasi objek antara lain :

1. Pendekatan *Top Down* yaitu memecahkan masalah ke dalam bagian-bagian terkecil atau *perlevel* sehingga mudah diselesaikan.
2. Pendekatan modul yaitu membagi sistem ke dalam modul-modul yang dapat beroperasi tanpa ketergantungan.
3. Penggunaan alat-alat bantu dalam bentuk grafik dan teks sehingga mudah untuk dimengerti serta dikoreksi apabila terjadi kesalahan.

Pendekatan dalam berorientasi objek dilengkapi dengan alat-alat dan teknik-teknik yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang dapat terdefinisi dengan baik dan jelas. Maka analisa berorientasi objek akan dilengkapi dengan alat dan teknik didalam mengembangkan sistem.

*Unified Modeling Language* adalah satu set peragaan konvensi yang digunakan untuk menetapkan atau menguraikan suatu sistem perangkat lunak dalam bentuk obyek. UML merupakan salah satu alat bantu yang sangat handal di dunia analisa dan perancangan sistem informasi yang berorientasi obyek (*Object Oriented*). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru (*Blue Print*) atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain .

## **2.6 Analisa Berorientasi Obyek**

Menurut Adi Nugroho, secara spesifik pengertian “berorientasi objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya



berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, polymorphism (polimorfisme), dan inheritance (perwarisan) <sup>[6]</sup>.

Analisa berorientasi objek (*Object-Oriented Analysis*) adalah sebuah pendekatan yang digunakan untuk:

1. Mempelajari obyek – obyek yang ada untuk mengetahui apakah obyek tersebut dapat digunakan berulang kali atau dapat disesuaikan untuk keperluan yang baru.
2. Menggambarkan obyek yang baru atau memodifikasi obyek, yang akan dikombinasi dengan obyek – obyek yang sudah ada ke dalam sebuah aplikasi bisnis komputer yang bermanfaat.

Alat bantu yang digunakan dalam analisa berorientasi obyek dengan UML antara lain adalah:

#### **2.7.1.1 Use Case Diagram, terdiri dari:**

1. *Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Penamaan use case sesuai dengan tujuan yang dicapai dari hasil interaksinya dengan actor. *Use case* biasanya menggunakan kata kerja.
2. *Actor* adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi actor, harus ditentukan pembagian tenaga kerja dan tugas – tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan *use case*, tetapi tidak memiliki kontrol terhadap *use case*.
3. Relasi (relationship) antara *actor* dengan *use case* pada *use case diagram* digambarkan dalam bentuk garis. Relasi antara *actor* dengan *use case* disebut

dengan asosiasi, asosiasi adalah sebuah relasi antara actor dengan *use case* dimana sebuah interaksi terjadi diantara mereka .

#### 4. Relasi *Use Case*

Selain terdapat relasi-relasi antara *actor* dan *use case*, juga terdapat relasi-relasi antara *use case* - *use case*. Ada beberapa jenis relasi antara *use case* – *use case*, yaitu:

- a. *Include*, digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.
- b. *Extend*, digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu yang dipenuhi.

#### 2.7.1.2 Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja. Simbol – simbol yang sering digunakan pada saat pembuatan activity diagram adalah sebagai berikut:

1. *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
2. *End Point*, akhir aktifitas.
3. *Activity*, menggambarkan suatu proses/kegiatan bisnis.
4. *Fork* (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5. *Join* (penggabungan) atau *Rake*, menunjukkan adanya dekomposisi.
6. *Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, true atau false.
7. *Swimlane*, pembagian *activity diagram* untuk menunjukkan siapa melakukan apa.

## 2.7.2 Perancangan Berorientasi Objek

Perancangan berorientasi obyek merupakan tahap lanjutan setelah analisa berorientasi obyek, perancangan berorientasi obyek adalah suatu pendekatan yang digunakan untuk menspesifikasi kebutuhan-kebutuhan sistem dengan mengkolaborasikan obyek-obyek, atribut-atribut, dan *method-method* yang ada.

Alat-alat yang digunakan dalam tahap perancangan sistem adalah sebagai berikut:

### 1. Diagram Kelas (*Class Diagram*)

Diagram kelas (*class diagram*) adalah suatu diagram yang melukiskan kelas yang sesuai dengan komponen-komponen perangkat lunak yang digunakan untuk membangun aplikasi perangkat lunak. Class diagram secara khas meliputi di bawah ini:

- a. Kelas (*Class*)
- b. Relasi *Associations*, *Generalization* dan *Aggregation*
- c. Atribut (*Attributes*)
- d. Operasi (*Operations/Method*)
- e. *Visibility*, tingkat akses obyek eksternal kepada suatu operasi atau atribut

### 2. *Sequence Diagram* (Diagram Urutan)

Setelah kita menentukan tanggung jawab dan perilaku dari obyek, kita dapat menciptakan suatu model yang terperinci dari bagaimana objek tersebut akan saling berhubungan satu sama lain untuk menyediakan kemampuan atau fungsi yang ditetapkan pada setiap use case yang telah didesain sebelumnya. UML menyediakan dua jenis diagram untuk melukiskan interaksi tersebut dengan nyata yaitu *Sequence Diagram* dan *collaboration diagram*.

*Sequence Diagram* menunjukkan secara detil bagaimana objek saling berhubungan satu sama lain dari waktu ke waktu, dan *collaboration diagram* menunjukkan bagaimana objek bekerja sama dalam bentuk urutan pesan untuk memenuhi fungsionalitas dari suatu *use case*. Definisi dari *sequence diagram* adalah suatu diagram UML yang memodelkan logika dari

suatu *use case* dengan menggambarkan interaksi berupa pengiriman pesan (*message*) antar objek dalam urutan waktu.

Beberapa simbol yang umum digunakan pada sequence diagram, yaitu:

- a. *Entity Object*, suatu objek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan ke dalam suatu database.
- b. *Interface/Boundary Object*, sebuah obyek yang menjadi penghubung antara *user* dengan sistem. Contohnya *window*, *dialogue box* atau *screen* (tampilan layar).
- c. *Control Object*, suatu obyek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai obyek. *Control object* mengkoordinir pesan (*message*) antara boundary dengan entitas.
- d. *Simple Message*, simbol pengiriman pesan dari sebuah objek ke obyek lain.
- e. *Recursive* sebuah obyek yang mempunyai sebuah operation kepada dirinya sendiri.
- f. *Activation*, mewakili sebuah eksekusi operasi dari obyek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
- g. *Lifeline*, garis titik-titik yang terhubung dengan obyek, sepanjang lifeline terdapat activation.

## 2.7 Model *Freamwork for Aplication of System Thinking* (FAST)

Menurut Pressman, Roger (2001) Metodologi pengembangan sistem (*system development methodology*) adalah proses pengembangan sistem yang sangat formal dan akurat yang mendefinisikan sekumpulan aktivitas, metode, praktek-praktek terbaik, penyampaian, dan alat terotomasi yang digunakan oleh pengembang sistem dan manajer proyek untuk mengembangkan dan memelihara sistem dan *software* informasi. Metodologi penelitian yang dipergunakan dalam penelitian ini adalah menggunakan metodologi **FAST** (*Framework for the*



*Application of Systems Thinking*).<sup>[7]</sup> Model Fast terdiri dari beberapa fase tahapan sebagai berikut;<sup>[8]</sup>

### **1. Fase Penyelidikan Awal**

Pada fase ini, penulis akan melakukan observasi dan wawancara di PT. Gusti Glori Sukses Pangkalpinang. Observasi dan wawancara digunakan untuk memperoleh data dimana data tersebut akan digunakan untuk menentukan cakupan awal sistem. Sehingga penulis dapat membuat sistem yang sesuai dengan kebutuhan.

### **2. Fase Analisis Masalah**

Pada fase kedua, penulis mempelajari wilayah permasalahan yang ditemukan, menganalisis permasalahan dan atau kesempatan tersebut beserta proses bisnisnya. Penentuansasaran pengembangan sistem diperlukan pada fase ini untuk mengetahui batasan dari sistem yang akan dikembangkan. Rencana proyek yang telah dibuat pada fase pertama akan diupdate atau dilakukan perubahan rencana jika pada fase ini terdapat ketidaksesuaian atas rencana proyek pada fase pertama. Penulis juga akan mengestimasi waktu yang dibutuhkan untuk membuat sistem yang diusulkan.

### **3. Fase Analisis Kebutuhan**

Pada fase ketiga, penulis akan mendefinisikan kebutuhan sistem yang dapat menjawab kebutuhan pengguna dengan apa yang dapat dihasilkan oleh sistem baru yang akan dibuat. Penulis juga akan menganalisis kebutuhan fungsional yang diharapkan dapat dipenuhi oleh sistem yang diusulkan, menelusuri dan melengkapi kebutuhan, memprioritaskan kebutuhan. Selain itu, penulis menganalisa teknologi yang cocok digunakan untuk permasalahan yang ada. Penulis juga mengestimasi waktu yang dibutuhkan untuk pembuatan sistem yang baru. Perubahan rencana proyek masih dilakukan pada fase ini jika terdapat ketidaksesuaian atas rencana proyek pada fase pertama dan fase kedua. Penulis juga akan mengestimasi waktu yang dibutuhkan untuk membuat sistem yang diusulkan.

#### **4. Fase Desain Logis**

Pada fase ini, hasil dari analisis kebutuhan akan diterjemahkan menjadi gambar-gambar yang disebut model sistem seperti *activity diagram*, *use case diagram*, *package diagram*, *classdiagram*, *deployment diagram*, *sequence diagram*, dan ERD. Penulis juga akan menentukan kebutuhan teknis untuk membantu penyelesaian sistem yang diusulkan. Model sistem yang dibuat diilustrasikan sebagai sistem yang terpisah dari sembarang solusi teknis yang mungkin. Dengan kata lain, pada fase ini akan menjawab pertanyaan-pertanyaan seputar teknologi (data, proses, antarmuka) yang menjamin kegunaan, kehandalan, kelengkapan, kinerja, dan kualitas yang akan dibangun di dalam sistem.

#### **5. Fase Analisis Keputusan**

Pada fase ini, penulis akan mengidentifikasi dan menganalisis kandidat-kandidat solusi perangkat lunak dan keras yang nantinya akan dipilih dan dipakai dalam implementasi sistem sebagai solusi atas masalah dan kebutuhan yang sudah didefinisikan pada tahapan sebelumnya, merekomendasikan sebuah target sistem (solusi) yang akan didesain, dibangun dan diimplementasikan. Penulis akan melakukan *update* rencana proyek jika terdapat ketidaksesuaian atas rencana proyek pada fase pertama dan fase yang berkaitan dengan fase ini. Penulis juga akan mengestimasi waktu yang dibutuhkan untuk membuat sistem yang diusulkan.

#### **6. Fase Konstruksi**

Setelah membuat desain logis dan menemukan solusi atas *software* dan *hardware* yang dibutuhkan, maka akan dilakukan konstruksi dan pengujian terhadap sistem yang memenuhi kebutuhan-kebutuhan bisnis dan spesifikasi desain yang dibuat pada tahap ini. Penulis akan membuat basis data, program aplikasi, rancangan antarmuka, dan kode program. Setelah pengujian terhadap keseluruhan sistem telah dilakukan, maka sistem siap untuk diimplementasikan.

### **2.8 Tinjauan Penelitian Terdahulu**

Tinjauan pustaka menguraikan hasil-hasil penelitian yang pernah dilakukan oleh peneliti sebelumnya yang mempunyai kaitan dengan penelitian :

1. Syantriawati (2017), **Rancang Bangun Sistem Informasi Pelayanan Bongkar Muat Barang Berbasis Web Pada PT. Nusantara Terminal Service Makassar, Fakultas Sains Dan Teknologi UIN Alaudin Makassar.**  
Proses pelayanan bongkar muat oleh bagian administrasi operasional masih belum terorganisir dan masih menggunakan cara konvensional. Disisi lain proses registrasi surat yang menjadi persyaratan kapal bisa sandar atau tidak masih membutuhkan waktu yang sangat lama. Perkembangan teknologi berkembang begitu pesat, hampir semua aspek kehidupan manusia bersinggungan dengan teknologi komputer. Tujuan dari penelitian ini adalah untuk merancang dan membangun sistem pelayanan bongkar muat barang pada PT. Nusantara Terminal Services Makassar yang dapat memudahkan para agen dalam melayani proses bongkar muat barang secara langsung dengan komputer. Jenis penelitian yang digunakan adalah penelitian kualitatif lapangan. Lokasi penelitian pada PT. Nusantara Terminal Services Makassar. Metode perancangan menggunakan metode *waterfall*. Teknik pengujian sistem menggunakan *BlackBox* dan *WhiteBox testing*. Penelitian ini menghasilkan sistem informasi Pelayanan Bongkar Muat Barang. Berdasarkan hasil pengujian, maka tujuan penelitian ini sudah tercapai, hal ini dibuktikan berdasarkan pengujian *WhiteBox*, *BlackBox*, dan hasil penyebaran kuesioner ke user target.
2. Simangunsong Andriyanus, Chandra Arvika, Desmardeni Henri (2013), **Sistem Pelayanan Kapal Dan Barang Pada PT. PELINDO II Cabang Palembang, STMIK Palcomtech Palembang.** PT PELINDO II Cabang Palembang merupakan perusahaan Badan Usaha Milik Negara (BUMN) yang bergerak dibidang kegiatan pelayanan jasa kepelabuhan. Pelayanan Kapal dan Barang merupakan tugas yang diemban pada lingkungan pelabuhan PT. PELINDO II Cabang Palembang. Teknologi informasi sangat dibutuhkan agar kegiatan pelayanan dapat tersaji secara cepat, tepat dan mudah. PT. PELINDO II Cabang Palembang telah menerapkan sistem layanan berbasis Information Communication Technology (ICT). Namun, tidak demikian pada divisi

Usteryang masih menggunakan cara manual. Hal ini mengakibatkan sistem berjalan dengan tidak efektif dan efisien, seperti sebagian dokumen penting yang masih ditulis satu per satu dan terkadang terjadinya kesalahan dalam pembuatan dokumen, penyimpanan berkas dokumen tidak dapat tersusun secara rapi dan data tidak dapat tersimpan secara aman karena banyaknya penumpukan data berupa arsip. Dengan adanya sistem informasi, kecepatan dan ketepatan pada pelayanan barang dapat terlaksana dengan baik dan kesalahan dapat diminimalisir, baik dalam pembuatan dokumen dan laporan maupun keamanan dan kemudahan dalam penyimpanan dan pelayanan barang. Hal ini lah yang menjadi landasan utama penulis untuk memberikan solusi guna memecahkan permasalahan yang ada dengan membuat sistem pelayanan berbasis dekstop, sehingga sistem pelayanan yang ada dapat berjalan secara cepat, tepat dan mudah. Dengan adanya sistem ini, penulis berharap agar permasalahan dan kendala yang ada pada PT. PELINDO II Cabang Palembang dapat terselesaikan dengan baik.

3. Syukron Farid (2014), **Aplikasi Pengelolaan Data Stuffing Pada PT. Pelabuhan Indonesia II (Persero) Cabang Palembang Berbasis Web, Politeknik Negeri Sriwijaya Palembang**. Divisi USTER yang terdapat di PT. Pelabuhan Indonesia II cabang Palembang merupakan divisi yang mengatur dan mengurus semua kegiatan bongkar muat barang khususnya *stuffing*. Pengolahan data *stuffing* di divisi USTER masih sering mengalami kesalahan dalam penambahan, pencarian dan penyimpanan datanya, maka dari itu tujuan dari penulisan laporan akhir ini adalah untuk mengetahui bagaimana membangun suatu aplikasi pengolahan data *stuffing* yang baik dalam menyajikan informasi data *stuffing* pada divisi USTER di PT. Pelabuhan Indonesia II cabang Palembang. Penelitian telah dilaksanakan dengan pengamatan langsung di divisi USTER PT. Pelabuhan Indonesia II Palembang dan hasil yang didapat yaitu divisi USTER pada PT. Pelabuhan Indonesia II Palembang membutuhkan suatu aplikasi yang memudahkan staff admin dalam pengolahan data *stuffing* untuk memberikan informasi



dan laporan yang cepat, tepat dan akurat kepada Kepala USTER. Dalam pembuatan aplikasi ini penulis menggunakan aplikasi pemrograman *Page Hypertext Preprocessor (PHP)* dengan database *MySQL*.

## **2.9 Teori Pendukung Bongkar Muat Kapal Isap Produksi**

### **2.9.1 Pengertian Bongkar Muat**

Pelabuhan adalah tempat yang terdiri atas daratan dan atau perairan dengan batas-batas tertentu sebagai tempat kegiatan pemerintahan dan kegiatan perusahaan yang dipergunakan sebagai tempat kapal bersandar, naik turun penumpang, dan atau bongkar muat barang, berupa terminal dan tempat berlabuhan kapal yang dilengkapi dengan fasilitas keselamatan dan keamanan pelayaran, kegiatan penunjang pelabuhan serta sebagai tempat perpindahan intra dan antar moda transportasi. (PP No. 61 tahun 2009).

Bongkar Muat adalah salah satu kegiatan yang dilakukan dalam proses *forwarding* barang. Yang dimaksud dengan kegiatan muat adalah proses memindahkan barang dari gudang, menaikkan lalu menumpuknya di atas kapal sedangkan kegiatan bongkar adalah proses menurunkan barang dari kapal lalu menyusunnya di dalam gudang di pelabuhan atau *Stock pile* atau *container yard*.

### **2.9.2 Kegiatan bongkar muat**

Alat-alat yang digunakan untuk aktivitas bongkar muat adalah:

1. *Grabs* adalah alat muat / bongkar yang sering digunakan untuk memuat/ membongkar barang jenis curah kering.
2. *Bucket* adalah sebuah bak dengan kapasitas tertentu yang digunakan untuk memuat barang curah atau bag.
3. *Crane* adalah suatu alat dengan kapasitas tertentu yang digunakan untuk menaikkan/ menurunkan barang dari/ke kapal.

4. *Sling* adalah jerat untuk muatan yang dibuat dari tali, termasuk tali kawat atau baja, gunanya untuk mengangkat atau menurunkan muatan dari/ke kapal.
5. *Forklift* adalah kendaraan roda empat yang berfungsi sebagai alat pemindah (transport) barang dari satu titik ke titik yang lain dengan jarak yang dekat. Operasional kendaraan ini banyak terdapat di lingkungan pabrik
6. *Loader* adalah mesin yang digunakan untuk meraup dan transportasi bahan dalam area kerja.
7. *Exchavator* adalah alat berat yang sering dipergunakan pada pekerjaan konstruksi, kehutanan dan industri pertambangan karena alat ini dapat melakukan berbagai macam pekerjaan.

Berdasarkan Pasal 2 ayat 1 Peraturan Menteri Perhubungan Republik Indonesia Nomor: PM 60 Tahun 2014 tentang Penyelenggaraan dan Pengusahaan Bongkar Muat Barang dari dan ke Kapal, kegiatan usaha bongkar muat adalah kegiatan usaha yang bergerak dalam bidang bongkar muat barang dari dan ke kapal di pelabuhan, yang meliputi kegiatan:

1. *Stevedoring*, adalah pekerjaan membongkar barang dari kapal ke dermaga/tongkang/truk atau memuat barang dari dermaga/tongkang/truk ke dalam kapal sampai dengan tersusun dalam palka kapal dengan menggunakan derek kapal atau derek darat.
2. *Cargodoring*, adalah pekerjaan melepaskan barang dari tali/jala-jala (ex tackle) di dermaga dan mengangkut dari dermaga ke gudang/lapangan penumpukan barang atau sebaliknya dan;
3. *Receiving/delivery*, adalah pekerjaan memindahkan barang dari timbunan/tempat penumpukan di gudang/lapangan penumpukan dan menyerahkan sampai tersusun diatas kendaraan di pintu gudang/lapangan penumpukan atau sebaliknya.

### **2.9.3 Mekanisme Pada Kegiatan Bongkar Muat**

Kegiatan bongkar muat adalah pekerjaan membongkar barang dari atas dek/palka kapal dan menempatkannya di atas dermaga atau ke dalam tongkang atau kebalikannya memuat dari atas dermaga atau dari dalam tongkang dan menempatkannya ke atas dek atau ke dalam palka kapal. Dalam pelaksanaannya kegiatan bongkar muat dilakukan oleh perusahaan bongkar muat atau dengan derek kapal apabila kapal tersebut memiliki derek sendiri. Perusahaan Bongkar Muat (PBM) adalah badan hukum Indonesia yang khusus didirikan untuk menyelenggarakan dan mengusahakan kegiatan bongkar muat barang.

