

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.(Asbon Hendra, 2012, Hal : 157).

##### **II.1.1. Karakteristik Sistem**

Ada beberapa karakteristik yang membentuk sebuah system yaitu:

###### **1. Komponen Sistem (*Component*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen tersebut dapat berupa suatu bentuk subsistem atau bagian-bagian dari sistem.

###### **2. Batasan Sistem (*Boundary*)**

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan, karena dengan batas system ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu system menunjukkan ruang lingkup (*scope*) dari system tersebut.

### 3. Lingkungan Luar Sistem (*Environment*)

*Environment* merupakan segala sesuatu diluar batas sistem yang memengaruhi operasi dari suatu sistem. Lingkungan luar system ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

### 4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

### 5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa perawatan (*maintenance input*) adalah energi yang dimasukkan supaya system tersebut dapat beroperasi. Masukan sinyal (*signal input*) adalah energi yang diproses untuk didapatkan keluaran.

### 6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh system, meliputi *output* yang berguna.

#### 7. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan menjadi keluaran yang diinginkan.

#### 8. Tujuan Sistem (*Goal*)

Setiap system pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. (Asbon Hendra, 2012, Hal : 158).

### II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandangannya antara lain :

#### a. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.

#### b. Sistem Fisik (*Physical System*)

Merupakan system yang ada secara fisik sehingga setiap makhluk dapat melihatnya.

c. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat, seperti system tata surya, system galaxy, system reproduksi, dan lain-lain.

d. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *Human Machine System*.

e. Sistem Tertentu (*Deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari system dapat diramalkan.

f. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

g. Sistem Tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan system luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teori, system tersebut ada, tetapi kenyataannya tidak ada system yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup).

h. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan system terotomasi, yang merupakan bagian dari system buatan manusia dan berinteraksi dengan

control oleh satu atau lebih computer sebagai bagian dari system yang digunakan dalam masyarakat modern. (Asbon Hendra, 2012, Hal : 160).

### **II.3. Informasi**

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi = *input-proses-output*.

### **II.4. Sistem Informasi**

“Suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya.”

“Sebuah system terintegrasi atau system manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi.”

“Sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/mengambil, mengolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan.”

“Sekelompok orang, prosedur, *input*, *output*, dan pengolahannya secara bersama-sama menghasilkan informasi yang akurat, tepat waktu, dan relevan bagi penggunanya”

Menurut Robert A. Leitch, system informasi adalah suatu system di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari

suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Ada empat operasi dasar dari sistem informasi, yaitu mengumpulkan, mengolah, menyimpan, dan menyebarkan informasi. Informasi mungkin dikumpulkan dari lingkungan dalam atau luar dan memungkinkan didistribusikan kedalam atau keluar organisasi.

## **II.5. Data**

Data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relatif, tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu informasi bagi level manajemen tertentu bias menjadi data bagi manajemen level di atasnya, atau sebaliknya. (Asbun Hendra, 2012).

## **II.6. Persediaan**

Persediaan dapat didefinisikan sebagai bahan yang disimpan dalam gudang untuk kemudian digunakan atau dijual. Persediaan dapat berupa bahan baku untuk keperluan proses, barang-barang yang masih dalam pengolahan dan barang jadi yang disimpan untuk penjualan. Persediaan adalah hal yang pokok sebagai fungsi yang tepat dari suatu usaha pengolahan/pembuatan.

Persediaan merupakan sejumlah bahan-bahan, bagian-bagian yang disediakan dan bahan-bahan dalam proses yang terdapat dalam perusahaan untuk proses produksi, serta barang-barang jadi/produk yang disediakan untuk memenuhi permintaan dari konsumen atau langganan setiap waktu.

Mengendalikan persediaan yang tepat bukan hal yang mudah, apabila jumlah persediaan terlalu besar mengakibatkan timbulnya dana menganggur yang besar, meningkatnya biaya penyimpanan, dan resiko kerusakan barang yang lebih besar. Namun jika persediaan terlalu sedikit mengakibatkan resiko terjadinya kekurangan persediaan karena seringkali baha/barang tidak dapat didatangkan secara mendadak dan sebesar yang dibutuhkan, yang menyebabkan terhentinya proses produksi, tertundanya penjualan, bahkan hilangnya pelanggan. (Meilani dan Saputra, 2013 : 327).

## **II.7. PT. Sinar Mas**

PT. Sinar Mas Agro Resources and Technology (SMART), Tbk adalah sebuah perusahaan agribisnis dengan komoditi utama berupa kelapa sawit yang berkantor pusat di jalan M. H. Thamrin No. 51 Kav. 22, Jakarta 10350, dengan presiden direktur Muktar Widjaja. Perusahaan ini dapat dihubungi pada nomer berikut : Phone : 021 318 1388 (hunting system) - Fax : 021 318 1389, Email investor@smart-tbk.com. PT. SMART, Tbk merupakan bagian dari Golden Agri- Resources Limited (GAR), perusahaan komoditi kelapa sawit terbesar di dunia yang berada di Singapore. PT. SMART, Tbk pertama kali didirikan tahun 1962, pada saat itu bernama PT Maskapai Perkebunan Sumcoma Padang Halaban, kemudian baru pada tahun 1991 berganti nama menjadi PT. SMART Corporation. Pada tahun 1992, pihak manajemen mengambil kebijakan untuk *go public* dengan mendaftarkan sahamnya ke bursa saham Jakarta dan Surabaya. Dengan *listingnya* perusahaan ini di pasar bursa, nama perusahaan berubah menjadi PT. SMART, Tbk pada tahun 1999

berdasarkan peraturan pemerintah no. 26 tahun 1998. (Rani dan Rahmawati, 2014 : 3).

## **II.8. Visual Basic 2010**

*Visual basic* dibuat oleh Microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi *database*. Visual basic merupakan bahasa pemrograman *event drive*, di mana program aplikasi yang dapat berupa kejadian atau *event*, misalnya ketika *user* mengklik tombol atau menekan enter. (Prayogi, dkk, 2015 : 3).

## **II.9. SQL Server 2008**

SQL Server 2008 adalah sebuah *Relational Database Management System* (RDBMS) yang sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, SQL Server 2008 memiliki banyak fitur yang bisa diandalkan untuk meningkatkan performa *database*.

SQL Server 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari hari berkaitan dengan *database*, seperti menulis T-SQL, melakukan *backup* dan *restore database*, melakukan *security database* terhadap aplikasi, dan sebagainya. Pada GUI tersebut kita bisa melakukan settingan terhadap SQL Server untuk berkerja lebih optimal. *Settingan* juga bisa dilakukan menggunakan *script* untuk

memudahkan *developer* mengubah *Setting Options* pada *SQL Server 2008*. (Ruslan, 2013 : 39).

## II.10. Normalisasi

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses mendekomposisikan sebuah tabel yang masih memiliki beberapa anomali atau ketidakwajaran sehingga menghasilkan tabel yang lebih sederhana dan struktur yang bagus, yaitu sebuah tabel yang tidak memiliki *data redundancy* dan memungkinkan *user* untuk melakukan *insert*, *delete*, dan *update* pada baris (*record*) tanpa menyebabkan inkonsistensi data. (Triyono, 2012 : 19).

### 1. *First Normal Form* (1 NF)

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal. Atribut *multivalued*, *composite*, *derive* tidak tunggal. Setiap nilai dari atribut hanya mempunyai nilai tunggal.

### 2. *Second Normal Form* (2 NF)

Untuk menjadikan tabel normal tingkat ke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *functional* tergantung pada semua atribut pembentuk *primary key*.

### 3. *Third Normal Form* (3 NF)

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung

pada *primary key* dan atribut tersebut juga tergantung pada atribut lain yang bukan *primary key*.

#### 4. *Boyce-codd Normal Form* (BCNF)

Tabel dalam BCNF jika sudah 3NF dan semua *determinants* adalah *candidate keys*. Perbedaan 3NF dan BCNF adalah untuk *functional dependency*  $A \rightarrow B$ , 3NF memperbolehkan ketergantungan ada dalam relasi jika B adalah *Primary Key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menuntut untuk ketergantungan tetap ada dalam relasi, A harus menjadi *candidate key*.

#### 5. *Fourth Normal Form* (4 NF)

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalued dependency*.

#### 6. *Fifth Normal Form* (5 NF)

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula. Jika penyusunan ini tidak mungkin dilakukan dikatakan pada relasi itu terdapat *join dependencies* dan dikatakan bersifat *lossy join*. (Triyono, 2012 : 20).

### II.11. *Unified Modeling Language* (UML)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan Siregar, 2015: 93).

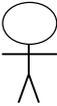
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

#### 1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini :

**Tabel II.2. Simbol *Use Case***

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use</i></p>

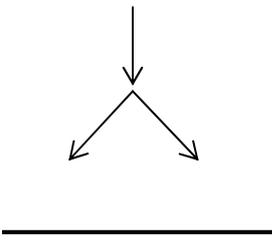
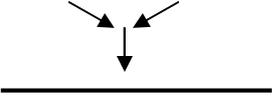
	<i>case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran.
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber:Urva dan Siregar; 2015: 94)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

**Tabel II.3. Simbol *Activity Diagram***

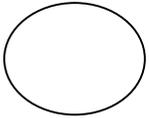
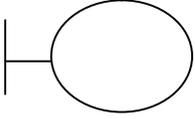
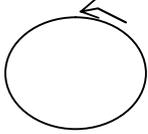
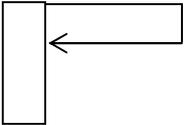
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

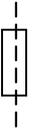
(Sumber : Urva dan Siregar; 2015: 94)

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

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

**Tabel II.4. Simbol *Sequence Diagram***

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar; 2015: 95)

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

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.5 dibawah ini:

**Tabel II.5. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih

0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

**(Sumber : Urva dan Siregar; 2015 : 95)**