

## **BAB II**

### **LANDASAN TEORI**

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

Menurut Romney dan Steinbart (2016:4) Sistem adalah “Suatu rangkaian dari dua atau lebih komponen yang berinteraksi untuk mencapai suatu tujuan”.

Menurut Abdussomad, dkk (2016:113) “Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Menurut Anggraeni dan Irviasi (2017:1) mendefinisikan bahwa, “Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah sekumpulan elemen yang saling berkaitan guna untuk mencapai tujuan tertentu.

##### **2.1.1. Karakteristik Sistem**

Menurut Fauzi (2017:3) karakteristik sistem adalah sebagai berikut:

1. Komponen Sistem

Suatu sistem terjadi dikarenakan adanya sejumlah komponen yang melakukan interaksi. Suatu sistem yang sekecil apapun akan selalu mengandung komponen-komponen.

## 2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antar suatu sistem dengan sistem yang lain atau dengan lingkungan luar.

## 3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah daerah di luar batas dari suatu sistem yang memengaruhi operasi sistem.

## 4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lainnya. Keluaran dari subsistem menjadi masukan untuk subsistem lainnya.

## 5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem.

## 6. Keluaran Sistem

Keluaran sistem adalah hasil dari energi yang di olah dan diklasifikasikan menjadi keluaran yang berguna dan sisi pembuangan.

## 7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolahyang akan mengubah masukan menjadi keluaran.

## 8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran maka sistem tersebut tidak berguna.

### 2.1.2. Klasifikasi Sistem

Menurut Sutabri (2016:11) “Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut”. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya:

#### 1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara pemikiran hubungan antara manusia dengan Tuhan sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

#### 2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui alam, tidak dibuat oleh manusia misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut *human machine system*. Sistem informasi berbasis komputer merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

### 3. Sistem Deterministik dan Sistem Probabilistik

Sistem yang beroperasi dengan tingkah laku dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

### 4. Sistem Tertutup dan Sistem Terbuka

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

#### 2.1.3. Konsep Dasar Informasi

Menurut Anggraeni dan Irviani (2017:1) mendefinisikan bahwa, “informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya, serta untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan”.

Menurut Hanafiah (2017:108) “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.

Menurut TMBooks (2017:4) “Informasi adalah data yang telah di organisir dan diproses sehingga bermanfaat bagi proses pengambilan keputusan”.

Berdasarkan tiga definisi informasi tersebut penulis dapat menyimpulkan bahwa informasi adalah serangkaian data yang diolah, dibentuk atau dimanipulasi menjadi data yang mempunyai sifat sementara tergantung dengan waktu yang memiliki kegunaan dan manfaat bagi orang yang menerimanya.

#### 2.1.4. Konsep Dasar Sistem Informasi

Sistem informasi menurut Leitch dan Davis dalam (Hanafiah, 2017:108) mengemukakan bahwa “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi-strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang di perlukan”.

Menurut Puspitasari (2016:228) mengemukakan bahwa “Sistem informasi merupakan sebuah sistem didalam suatu organisasi yang mempertemukan berbagai kebutuhan proses pengolahan transaksi harian, membantu dan mendukung seluruh kegiatan operasi, bersifat manajerial dari suatu organisasi dan membantu memperlancar penyediaan laporan yang dibutuhkan”.

### 2.1.5. Konsep Dasar Akuntansi

Menurut Tresnawati, dkk (2017:1164) “Akuntansi dapat didefinisikan sebagai sistem informasi yang menyediakan laporan untuk para pemangku kepentingan mengenai aktifitas ekonomi dan kondisi perusahaan”.

Menurut Sudrajat, dkk (2016:1249) “Akuntansi adalah seni mencatat, mengelompokkan, mengikhtiarkan menurut cara yang berarti, serta kejadian yang sekurang-kurangnya bersifat finansial, kemudian dari catatan itu bisa ditafsirkan hasilnya”.

Berdasarkan definisi di atas dapat ditarik kesimpulan bahwa akuntansi merupakan proses yang terdiri dari identifikasi, pengukuran dan pelaporan informasi ekonomi. Informasi ekonomi yang dihasilkan oleh akuntansi diharapkan berguna dalam pengolahan keputusan mengenai kesatuan usaha yang bersangkutan. Informatika.

### 2.1.6. Konsep Dasar Sistem Informasi Akuntansi

Sistem Informasi Akuntansi menurut Romney dan Steinbart (2016:10) mengemukakan bahwa, “Sistem informasi akuntansi adalah suatu sistem yang mengumpulkan, mencatat, menyimpan, dan mengolah data untuk menghasilkan informasi bagi pengambil keputusan. Sistem ini meliputi orang, prosedur dan intruksi, data, perangkat lunak, infrastruktur teknologi informasi, serta pengendalian internal dan ukuran keamanan”.

### 2.1.7. Konsep Dasar Sistem Pembelian

Menurut Hanafiah dan Solihin (2017:108) mengemukakan bahwa, “ Pembelian adalah sebagai salah satu fungsi dari pembelanjaan atau merupakan kegiatan dari pembelanjaan. Pembelian sama pentingnya dengan penjualan, yaitu untuk memenuhi kebutuhan setiap perusahaan, seperti kebutuhan peralatan kantor, gedung, peralatan produksi, dan lain sebagainya ” .

Transaksi Pembelian dicatat dengan jurnal sebagai berikut:

1. Jika pembelian secara tunai

Pembelian	xxx
Kas	xxx

2. Jika pembelian secara kredit

Pembelian	xxx
Hutang Dagang	xxx



## 2.2. Peralatan Pendukung

### 2.2.1. *Unified Modeling Language (UML)*

Menurut ( Sukamto dan Shalahudin, 2016:133 ) “UML adalah standar bahasa yang digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain serta menggambarkan *arsitey* dalam pemrograman berorientasi objek”.

UML di deskripsikan oleh beberapa diagram diantaranya (Sukamto dan Shalahudin):

#### 1. *Use case Diagram*

*Use Case diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan *use case* diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem.



**Tabel II.1.**  
**Use Case Diagram**

Simbol	Deskripsi
	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
 Top Package::nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi/association 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
<i>Ekstensi/extend</i> «extends» 	Relasi use case tambahan sebuah use case dimana use case yang di tambahan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan
<i>Generalisasi/generalization</i> «uses» 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini

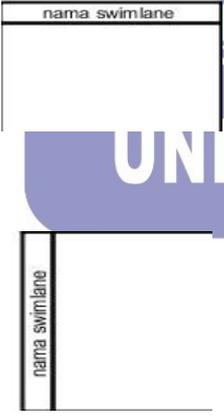
Sumber : Sukamto dan Shalahudin

### 1. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*.



**Tabel II.2.**  
**Activity Diagram**

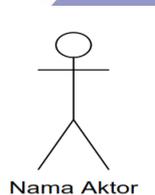
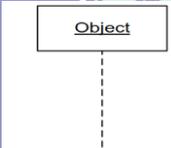
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> Atau 	Memisahkan organisasi bisnis, yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Sukamto dan Shalahudin

## 2. Sequence Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

**Tabel II.3.**  
**Sequence Diagram**

Simbol	Deskripsi
<p>Aktor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>
<p>Object Lifeline</p> 	<p>Menyatakan kehidupan suatu objek, Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinterkasi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya menyatakan objek yang berinteraksi pesan</p>
<p>Pesan tipe create / call</p> <p>Message1</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. menyatakan suatu objek memanggil operasi atau metode yang adapada objek lain atau dirinya sendiri</p>
<p>Pesan tipe send</p> <p>Message2</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek lain yang dikirimin</p>

<p>Pesan tipe return</p> <p style="text-align: center;">Message3</p> <p style="text-align: center;">←-----</p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
--	--

Sumber : Sukamto dan Shalahudin

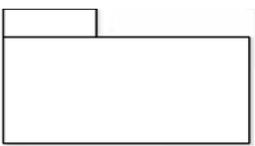
### 3. Deployment Diagram

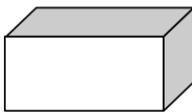
Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. Sistem tambahan ( *embedded system*) yang menggambarkan rancangan
- b. *device, node, dan hardware.*
- c. Sistem *client/server*
- d. Sistem terdistribusi murni
- e. Rekayasa ulang aplikasi



**Tabel II.4.**  
*Deployment Diagram*

Simbol	Deskripsi
<p><i>Package</i></p> 	<p><i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i>.</p>

<p><i>Node</i></p> 	<p>Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Kebergantungan antar node, arah panah mengarah pada <i>node</i> yang dipakai</p>
<p><i>Link</i></p> 	<p>Relasi antar <i>node</i></p>

Sumber : Sukamto dan Shalahudin

### 2.2.2. Entity Relationship Diagram (ERD)

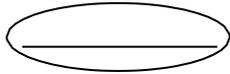
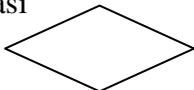
Salah satu cara permodelan data adalah dengan *Entity Relationship Diagram* (ERD). Menurut Fatta dalam (Hutapea & Muningsih, 2017:12) menjelaskan bahwa “ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan dan digunakan dalam sistem bisnis”. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data, yang akhirnya ERD bisa juga digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang akan dibangun. Beberapa poin yang harus diperhatikan dalam menggunakan ERD untuk menunjukkan aturan bisnis yaitu (Hutapea & Muningsih) :

1. Aturan bisnis adalah batasan yang harus diikuti ketika sistem beroperasi

2. Simbol ERD hanya menunjukkan satu *instance* dari entitas harus ada sebelum *instance* lain dari suatu entitas. Sebagai contoh: seorang dokter harus ada sebelum perjanjian ketemu dengan dokter dibuat.
3. Simbol ERD dapat menunjukkan ketika salah satu *instance* dari suatu entitas dapat direlasikan dengan satu anggota atau lebih dari entitas lainnya. Sebagai contoh, satu dokter bisa memiliki banyak pasien, satu pasien bisa jadi hanya memiliki satu dokter utama.
4. Simbol ERD juga menunjukkan ketika eksistensi dari suatu *instance* dalam suatu *entity* adalah opsional untuk sebuah relasi dengan instance lain dari suatu entitas. Sebagai contoh, pasien mungkin memiliki atau mungkin tidak memiliki biaya asuransi. ERD terbagi atas tiga komponen, yaitu entitas (*entity*), atribut (*attribute*), dan relasi atau hubungan (*relation*). Atribut atau *field* berperan sebagai penjelas dari entitas, dan relasi atau hubungan menunjukkan hubungan yang terjadi antara dua entitas.

**Tabel II.5.**  
**Entity Relationship Diagram (ERD)**

No.	Simbol	Deskripsi
1	Entitas/ <i>Entity</i> 	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; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
2	Atribut 	<i>Field</i> atau kolom data yang butuh di simpan dalam suatu entitas

3	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh di simpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
4	Atribut multival/multival ue 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
5	Relasi 	Relasi yang menghubungkan antar entitas; Biasanya diawali dengan kata kerja
6	Asosiasi/association 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

### 2.2.3. Logical Relational Structure (LRS)

Menurut (Hutapea & Muningsih, 2017:13) menjelaskan bahwa LRS merupakan suatu hasil dari pemodelan *Entity Relationship* (ER) beserta dengan atributnya sehingga bisa terlihat hubungan-hubungan antar entitas yang ada. LRS bertujuan untuk menentukan kardinalitas, jumlah tabel dan *foreign key* (FK). Hal yang perlu diperhatikan untuk mengubah ERD kedalam bentuk LRS yaitu tingkat hubungan (*cardinality*) apakah 1:1, 1:M atau M:M.

### 2.2.4. Spesifikasi File

Menurut (Wijaya & Supriadi, 2015) menjelaskan bahwa : spesifikasi *file* adalah normalisasi data yang menunjukkan attitude (*field*) apa saja yang terdapat dalam sebuah *file*. Spesifikasi file memberikan rincian yang lebih

lengkap yang berisi kode *file*, organisasi, *primary key*, panjang *record*, dan deskripsi *field* (No., nama *field*, *type field*, keterangan)

### 2.2.5. *User Interface*

Menurut (Irawan, 2015) menyimpulkan bahwa *user* sama dengan pengguna, dan *interface* sama dengan antarmuka. *user interface* adalah suatu alat yang disediakan oleh sistem operasi (OS) sebagai sarana untuk berinteraksi dengan sistem operasi (OS) yang mana pengguna (*user*) memberikan perintah kepada sistem operasi (OS) yang mana pengguna (*user*) memberikan perintah kepada sistem operasi (OS) untuk kemudian diterima, diproses dan di tampilkan oleh sistem operasi (OS).

### 2.2.6. *Black Box*

Menurut (Setyawan, 2016) menyimpulkan bahwa *black box* adalah salah satu tipe testing yang memperlakukan perangkat lunak yang di ketahui kinerja internalnya.

### 2.2.7. *Code Generation*

Menurut (Monalisa dan Mahendra, 2017) menyimpulkan bahwa *code generation* (pengkodean) merupakan tahap penerjemah desain sistem yang telah dibuat ke dalam bentuk bahasa yang dapat di mengerti oleh komputer.

### 2.2.8. Spesifikasi *Software* dan *Hardware*

Menurut (Husaini dan Dwi, 2017) menyimpulkan bahwa spesifikasi *software* (perangkat lunak) adalah kebutuhan *software* atau perangkat lunak penting bagi terciptanya sistem yang sedang dirancang.

Spesifikasi *hardware* (perangkat keras) adalah *hardware* atau perangkat keras yang digunakan dalam pembuatan sistem informasi.

