

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep dasar Sistem**

Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling ketergantungan satu sama lain dan terpadu.

##### **2.1.1 Pengertian Sistem**

Al fatta (2007:3) menjelaskan sistem merupakan suatu kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung sama lain.

Sujarweni (2015:1) juga menjelaskan bahwa sistem dilihat dari elemen-elemennya sistem adalah sekumpulan elemen yang saling berkaitan dan bekerja sama dalam melakukan kegiatan untuk mencapai suatu tujuan, sistem dilihat dari masukan dan keluaranya adalah suatu rangkaian yang berfungsi menerima input (masukan), mengelolah input dan menghasilkan output (keluaran).

Dari defenisi-defenisi sistem tersebut diatas dapat disimpulkan bahwa sistem terdiri dari unsur-unsur elemen-elemen yang merupakan bagian yang saling terpadu dan saling berhubungan secara erat antara satu dengan yang lainnya dengan sifat serta kerja samanya antar unsur atau elemen tersebut mempunyai bentuk yang tertentu dan saling mempengaruhi untuk mencapai satu atau berapa maksud secara

bersama-sama berkeinginan untuk mencapai tujuan yang sama pula. Dapat pula dikatakan bahwa suatu sistem terdiri dari struktur dan proses.

Struktur merupakan unsur-unsur yang secara terintegrasi membentuk sistem tersebut. Sedangkan proses merupakan penjelasan prosedur atau tata urutan kerja dari suatu sistem untuk mencapai tujuan tertentu. Agar suatu sistem dapat bekerja secara efisien dan efektif maka setiap struktur dan proses tersebut harus saling berkaitan antara satu dengan yang lainnya agar terciptanya sistem yang baik dan berguna.

### 2.1.2 Klasifikasi sistem

Pengklasifikasian sistem menurut Azhar Susanto dalam Novianti (2012:9) adalah seperti yang terlihat dalam tabel berikut:

**Tabel II.1.**  
**Klasifikasi Sistem**

KRITERIA	KLASIFIKASI	
Lingkungan	Sistem Terbuka	Sistem Tertutup
Asal Pembuatan	Buatan Manusia	Buatan Allah/alamiah
Keberadaannya	Sistem berjalan	Sistem Konsep
Kesulitan	Sulit/Kompleks	Sederhana
Output/Kinerjanya	Dapat dipastikan	Tidak dapat dipastikan
Waktu keberadaannya	Sementara	Selamanya
Wujudnya	Abstrak	Ada secara fisik
Tingkatannya	Sub sistm/Sistem	Super sistem
Flesibilitas	Bisa berinteraksi	Tidak dapat beradaptasi

Sumber: Azhar Susanto dalam Novianti (2012:9)

1. Sistem Terbuka dan Sistem Tertutup

Sebuah sistem dikatakan terbuka bila aktifitas di dalam sistem tersebut dipengaruhi oleh lingkungannya, sedangkan bila tidak terpengaruh oleh perubahan yang terjadi di lingkungannya, maka disebut sistem tertutup.

2. Sistem Buatan Manusia dan Sistem Buatan Allah SWT /Alamiah

Suatu sistem bila diklasifikasikan berdasarkan asalnya, maka ada sistem buatan manusia seperti organisasi perusahaan dan lain-lain. Dan sistem buatan Allah SWT (alamiah) seperti manusia, pohon-pohon dan lain-lain.

3. Sistem Berjalan dan Sistem Konsep

Suatu sistem yang belum diterapkan disebut sistem konseptual/konsep. Sedangkan sistem konseptual yang dapat diterima oleh penggunanya untuk menunjang operasi sehari-hari maka disebut sistem berjalan.

4. Sistem Sederhana dan Kompleks

Sistem sederhana merupakan sebuah sistem yang terbentuk dari sedikit tingkatan dan komponen atau sub sistem serta hubungan antara mereka sangat sederhana. Sistem kompleks adalah sebuah sistem yang terdiri dari banyak komponen dan tingkatan yang dihubungkan dalam berbagai cara yang berbeda.

5. Sistem yang kinerjanya dapat dan tidak dapat dipastikan

Sistem yang dapat dipastikan kinerjanya artinya dapat ditentukan pada saat sistem tersebut akan dan sedang dibuat misalnya sistem listrik dimana kita tinggal. Sedangkan sistem yang tidak dapat dipastikan artinya tidak dapat ditentukan dari awal, yakni tergantung kepada situasi yang dihadapi misalnya organisasi perusahaan.

6. Sistem Sementara dan Selamanya

Sistem sementara artinya sistem yang digunakan hanya dalam periode tertentu misalnya sistem pemilu. Sedangkan sistem selamanya artinya sistem tersebut digunakan untuk waktu yang tidak ditentukan misalnya sistem lalu lintas.

7. Sistem yang ada secara Fisik dan Abstrak

Sistem yang ada secara fisik artinya kita dapat menyentuhnya atau merasakannya. Sedangkan sistem abstrak sebaliknya, yakni tidak dapat disentuh.

8. Sistem, sub sistem dan super sistem

Berdasarkan tingkatannya/hirarki sebuah sistem bisa merupakan komponen dari sistem yang lebih besar. Sub sistem adalah sistem yang lebih kecil yang ada dalam sebuah sistem. Super sistem adalah sistem yang sangat besar dan sangat kompleks.

9. Sistem yang bisa dan tidak bisa beradaptasi

Sistem yang bisa beradaptasi adalah sistem yang memiliki kemampuan untuk beradaptasi terhadap setiap pengaruh yang diakibatkan oleh perubahan yang terjadi di lingkungannya. Kebalikannya disebut sistem yang tidak bisa beradaptasi.

### **2.1.3 Karakteristik sistem**

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem yang lainya. Fatta (2007:5) :

1. Batasan (*boundary*) : penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang diluar sistem.
2. Lingkungan (*environment*) : segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*) : sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanupulasi oleh suatu sistem.
4. Keluaran (*output*) : sumber daya atau produk (infomasi, laporan, dokumen, tampilan layer computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*) : kegiatan-kegiatan atau prosees dalam suatu sistem yang menginformasikan input menjadi bentuk setengah jadi (output). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*) : tempat dimana komponen atau sistem dan lingkungan nya bertemu atau berinteraksi.
7. Penyimpanan (*storage*) : area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagai nya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berada dari berbagai data yang sama.

#### **2.1.4 Pengertian Informasi**

Berbicara informasi tidak akan lepas dengan data dan teknologi. Menurut Pratama (2014:9) Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memiliki arti dan manfaat.

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.(Fatta 2007:9).

Berdasarkan Pengertian informasi menurut para ahli yang telah disebutkan diatas, dapat disimpulkan bahwa informasi adalah sekumpulan fakta-fakta yang telah diolah menjadi bentuk data, sehingga dapat menjadi lebih berguna dan dapat digunakan oleh siapa saja yang membutuhkan data-data tersebut sebagai pengetahuan ataupun dapat digunakan dalam pengambilan keputusan.

### **2.1.5 Pengertian Sistem Informasi**

Menurut Fatta (2009:9) Sistem informasi merupakan suatu perkumpulan data yang terorganisasi beserta tatacara penggunaannya yang mencangkup lebih jauh dari pada sekedar penyajian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tatacara penggunaannya. Keberhasilan suatu sistem informasi yang diukur berdasarkan maksud pembuatannya tergantung pada tiga faktor utama, yaitu : keserasian dan mutu data, pengorganisasian data, dan tatacara penggunaannya. Untuk memenuhi permintaan penggunaan tertentu, maka struktur dan cara kerja sistem informasi berbeda-beda bergantung pada macam keperluan atau macam permintaan yang harus dipenuhi. Suatu persamaan yang menonjol ialah suatu sistem informasi menggabungkan berbagai ragam data yang dikumpulkan dari berbagai sumber. Untuk dapat menggabungkan data yang berasal dari berbagai sumber suatu sistem alih rupa (*transformation*) data sehingga jadi tergabungkan (*compatible*). Berapa pun ukurannya dan apapun ruang lingkungannya suatu sistem informasi perlu memiliki ketergabungan (*compatibility*) data yang disimpannya.

### **2.1.6 Pengertian Sistem Informasi Akuntansi**

Menurut Mujilan (2012:3) Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi.

SIA juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi. Saat ini, digital dan informasi online semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai faktor yang terkait ketika mengatur sistem informasi akuntansi.

SIA pada umumnya meliputi beberapa siklus pemrosesan transaksi:

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumberdaya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas.

### 2.1.7 Aktiva tetap

Sugiono (2009:199) Merupakan aktiva berwujud yang di peroleh dalam keadaan jadi atau dengan cara atau dengan cara di bangun terlebih dahulu dan digunakan untuk operasi normal perusahaan serta memiliki masa manfaat lebih dari 1 tahun.

Adapun jenis-jenis aktiva tetap sebagai berikut :

1. Aktiva tetap yang tidak dapat disusutkan

Yang termasuk dalam katagori ini adalah aktiva tetap yang mempunyai umur atau masa yang tidak terbatas, contoh : Tanah. Harga perolehan tanah tidak perlu disusutkan karena mempunyai umur dan masa tidak terbatas serta harga tanah tidak mengalami penurunan(dalam kondisi normal).

2. Aktiva tetap yang disusutkan

Yang termasuk dalam katagori ini adalah aktiva tetap yang umur dan masa manfaatnya yang terbatas. Dapat dibagi menjadi :

- a. Jika umur dan masa manfaat dari aktiva tetap tersebut sudah habis dan dapat diganti oleh aktiva sejenis maka harga perolehannya dapat dialokasikan dengan cara penyusutan. Contoh : mesin \_mesin pabrik, kendaraan angkut, peralatan kantor, dll.
- b. Jika umur dan masa manfaat aktiva tetap tersebut sudah habis tetapi tidak dapat digantikan oleh aktiva lain maka harga perolehannya dapat dialokasikan dengan cara depleksi (*depletion*). Contoh : lahan pertambangan.

Berhubungan dengan adanya biaya-biaya yang timbul untuk memperoleh aktiva tetap tersebut, maka dapat dibedakan menjadi :

1. Pengeluaran modal, merupakan pengeluaran yang berkaitan dengan aktiva tetap yang mempunyai manfaat lebih dari 1 priode akuntansi. Pengeluaran ini harus ditambah dalam nilai perolehan.
2. Pengeluaran pendapatan , merupakan pengeluaran yang berkaitan dengan aktiva tetap yang mempunyai manfaat dalam 1 priode akuntansi.

Pengeluaran ini di bebaskan sebagai biaya dalam priode bersangkutan.

Jurnal untuk mencatat pembelian aktiva tetap :

**Tabel II.2.**  
**Jurnal perolehan aktiva tetap**

Jan 00	1	Aktiva tetap Kas		Rp.xxx	Rp. Xxx
--------	---	---------------------	--	--------	---------

Sumber: Sugiono (2009:211)

Penyusutan merupakan suatu proses biaya dari harga perolehan aktiva tetap yang diterapkan secara sistematis sesuai dengan umur ekonomisnya. Sedangkan pada sumber dayanya alam disebut deplesi.

Untuk menghitung besarnya nilai penyusutan adalah :

$$\frac{\text{Harga perolehan aktiva tetap} - \text{Nilai residu}}{\text{Umur ekonomis}}$$

**Tabel II.3.**  
**Jurnal Penyusutan aktiva tetap**

Tanggal	Rekening	Ref	Debit	Kredit
Des 01	Beban Penyusutan mesin Akumulasi Penyusutan mesin		Rp.9000.000	Rp.9000.000

## 2.2 Peralatan pendukung (*Tools System*)

Peralatan pendukung (*Tool System*) merupakan alat yang digunakan untuk menggambar logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, ataupun diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Fungsi dari peralatan pendukung (*Tool System*) adalah untuk menjelaskan kepada user bagaimana fungsi dari sistem informasi dapat bekerja dengan suatu bentuk *logika model* dan *physical model*.

### 2.2.1 Object Oriented Programming (OOP)

Shalahuddin dan Rosa (2015:100) Suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Keuntungan menggunakan metode berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lain yang melibatkan objek tersebut (reusable).

2. Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

#### 4. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

#### 5. Meningkatkan kualitas perangkat lunak

Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pengguna serta mempunyai sedikit kesalahan.

### 2.2.2 Karakteristik OOAD

Mathiassen dalam Andria (2016:21) Dalam pendekatan berorientasi objek ada 4 pilar utama yang harus dipahami dalam pendekatan berorientasi objek yaitu karakteristik. Karakteristik (ciri) suatu program termasuk OOAD/OOP, apabila terdapat abstraksi, pembungkusan (*encapsulation*), *polymorphisme*, dan turunan (*inheritance*).

#### 1. *Abstraction*

Kemampuan untuk menjadikan dalam bentuk yang lebih sederhana. Hal ini juga dikenal dalam metodologi pendekatan struktur yaitu dekomposisi seperti menyederhanakan suatu sistem dalam bentuk *Context Diagram*.

#### 2. *Encapsulation*

Merupakan suatu karakteristik OOAD dimana program terbungkus (jadi satu) data dan perilaku, artinya lebih memperhatikan aspek internal daripada aspek eksternal. Contoh: dalam program terdapat tombol *button close* didalamnya ada *method system.exit(0)* untuk keluar dari sistem java. Berbeda dengan metodologi terdahulu, metodologi ini menggabungkan atribut dan fungsi / proses kedalam suatu objek yang disebut dengan *encapsulation*. Setiap objek dapat “menyembunyikan”

kompleksitasnya dan berhubungan dengan objek lain dengan mengirim “pesan/message” yang dapat dikenal dan diproses oleh objek penerima. Contoh: Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan *rice cooker*, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep *information hiding*.

### 3. *Polymorphisme*

Dengan kata lain suatu mekanisme yang memungkinkan suatu objek memiliki semua atau sebagian definisi dari objek induk. Menurut Bambang *Polymorphism* berasal dari kata Poly yang artinya banyak dan morph yang artinya bentuk. Jadi *polymorphism* adalah kemampuan suatu atribut atau method dapat berubah dalam berbagai bentuk dalam implementasi. Contoh Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup/pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

### 4. *Inheritance*

Merupakan suatu karakteristik OOAD di mana suatu kelas (*parent/base class*) dapat diturunkan ke kelas lain (*child/derived class*), sehingga kelas anak dapat memiliki data atau perilaku kelas orang tuanya. Contoh dengan beberapa buah

mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dikatakan obyek induk (*parent*). Sedangkan minibus obyek anak (*child*), berarti semua operasi yang berlaku pada mobil berlaku pada minibus.

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

Shalahuddin dan rosa (2015:133) UML adalah Standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek.

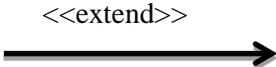
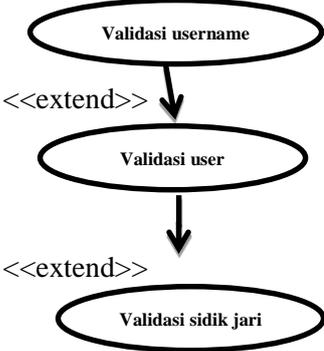
#### **2.2.3.1. Use Case Diagram**

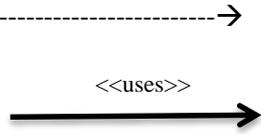
Shalahuddin dan rosa (2015:155) Use case atau diagram use case merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. use case mendeskripsikan sebuah interaksi antara satu atau lebih actor digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi apa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada use case adalah nama di definisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu mendefenisikan apa yang disebut actor dan use case.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

**Tabel II.4**  
**Simbol Use Case Diagram**

3. Simbol	Deskripsi
<i>Use case</i> 	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.
aktor/actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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 <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi/ extend 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal  arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

Simbol	Deskripsi
<p>Menggunakan/include/uses</p> <p>&lt;&lt;include&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none"> <li>• Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</li> <li>• Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</li> </ul>

Sumber : Shalahuddin dan rosa (2015:156)

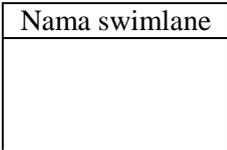
### 2.2.3.2. Activity Diagram

Shalahuddin dan rosa (2015:161) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal sebagai berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- Rancangan menu yang ditampilkan pada perangkat lunak.

**Tabel II.5**  
**Simbol 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.
Swimlane  atau 	Memisahkan sebuah organisme bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Shalahuddin dan rosa (2015:161)

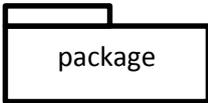
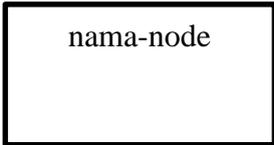
### 2.2.3.3. Deployment Diagram

Menurut Shalahuddin dan rosa (2015:154) Diagram *deployment* atau *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

- Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
- Sistem *client/server*

Berikut adalah simbol-simbol yang ada pada diagram deployment:

**Tabel II.6**  
**Simbol Deployment Diagram**

Simbol	Deskripsi
Package 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i> .
Node 	Biasanya mengacu pada perangkat keras( <i>hardware</i> ), perangkat lunak yang dibuat sendiri( <i>software</i> ), jika didalam node di sertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikuti sertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
Kebergantungan/ <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
<i>Link</i> 	Relasi antar <i>node</i> .

Sumber : Shalahuddin dan rosa (2015:161)

#### 2.2.3.4. Sequence Diagram

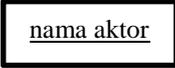
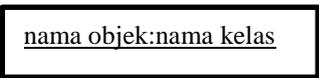
Shalahuddin dan rosa (2015:165) Diagram sequen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi

menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

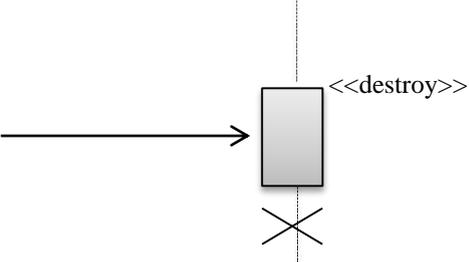
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

**Tabel II.7**  
**Simbol Sequence Diagram**

Simbol	Deskripsi
Aktor  <b>nama aktor</b> atau  <u>nama aktor</u> Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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.
Garis hidup/ lifeline 	Menyatakan kehidupan suatu objek
objek  <u>nama objek:nama kelas</u>	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe create 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.

Sumber : Shalahuddin dan rosa (2015:165)

Simbol	Deskripsi
Pesan tipe call l:nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarahkan pada objek yang memiliki operasi /metode maka operasi/metode yang dipanggil sesuai dengan kelas objek yang berinteraksi.
Pesan tipe send l:masukan   Pesan tipe return l:keluaran 	Menyatakan bahwa suatu objek yang mengirimkan data masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. 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.
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panak mngarah pada objek yang di akhiri,sebaiknya jika ada create maka ada distroy

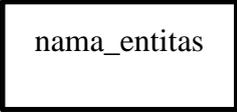
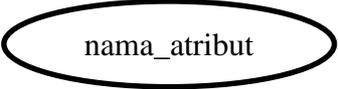
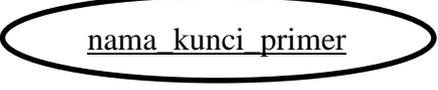
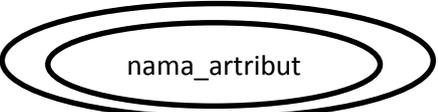
Sumber : Shalahuddin dan rosa (2015:165)

#### 2.2.4 Entity Relationship Diagram (EDR)

Shalahuddin dan rosa (2015:50) Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data rasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oelh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian

Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

**Tabel II.8**  
**Simbol Entity Relationship Diagram**

Simbol	Deskripsi
Entitas/ <i>entity</i> 	Entitas merupakan data 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.
Atribut 	<i>Field</i> atau kolom data yang dibutuhkan dalam suatu entitas
Atribut kunci primer 	<i>field</i> atau kolom data yang dibutuhkan disimpan 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).
Atribut multivalori/ <i>multivalued</i> 	<i>Field</i> atau kolom data yang dibutuhkan disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja

Simbol	Deskripsi
<p data-bbox="316 315 584 344">Asosiasi/<i>association</i></p> 	<p data-bbox="831 349 1350 495">Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> <p data-bbox="831 499 1350 645">Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas lain disebut dengan kardinalitas.</p>

Shalahudin dan Rosa (2015:50)

### 2.2.5. Logical Record Structure (LRS)

Menurut Frieyadie (2007:13) “LRS merupakan hasil dari pemodelan *Entity Relational Ship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”. Dalam pembuatan LRS terdapat 3 hal yang dapat mempengaruhi yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka di gabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

### **2.2.6. Microsoft Visio 2010**

Pascal (2010:171) Microsoft visio adalah software pembuat diagram dan flowchart berbasis vector yang awalnya dibuat oleh Visio Corporation. Pada tahun 1995 Visio 4.0 dirilis. Ini merupakan versi visio pertama yang dapat berjalan di Windows 95 dan sekaligus menjadi versi terakhir yang dibuat Visio Corporation. Pada bulan Januari 2000, Microsoft mengakuisisi Visio Corporation dengan membeli seluruh sahamnya. Ini merupakan akuisisi terbesar yang pernah Microsoft lakukan, kini Visio merupakan salah satu divisi Microsoft dan terus mengembangkan software dengan nama barunya Microsoft Office Visio. Visio mulai terintegrasi di dalam Microsoft Office sejak Office 2007. Sejumlah fitur baru ditambahkan terutama integrasi antara data dengan diagram dan otomatisasi dalam pembuatan bagan dan diagram. Di Office 2010, Visio mengalami perubahan antarmuka mengikuti standar aplikasi Office lainnya.

### **2.2.7. Netbeans IDE 7.1**

Nofriadi (2015:4) Netbeans merupakan sebuah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing dan banyak digunakan sekarang sebagai editor untuk berbagai bahasa pemrograman. Sampai sekarang, Netbeans sudah sampai ke versi 8.0. Pada Netbeans, kita bisa membuat bahasa pemrograman Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, Scala, Clojure. Swing merupakan teknologi Java untuk pengembangan aplikasi desktop yang bisa dijalankan di berbagai sistem operasi, seperti windows, linux, Mac OS X, dan Solaris.

### **2.2.8. MySql**

Menurut Wahana Komputer (2010:5) MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi, yaitu free software dan shareware.

### **2.2.9. XAMPP**

Menurut Wahana (2015:55) mengemukakan bahwa “XAMPP adalah tool yang menyediakan paket perangkat lunak dalam salah satu buah paket”.

### **2.2.10. JasperReport**

Miftakhul Huda (2010:191) JasperReport merupakan tool atau SDK (Software Development Kit) pembuat report (laporan) dalam lingkungan Java.