

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Informasi

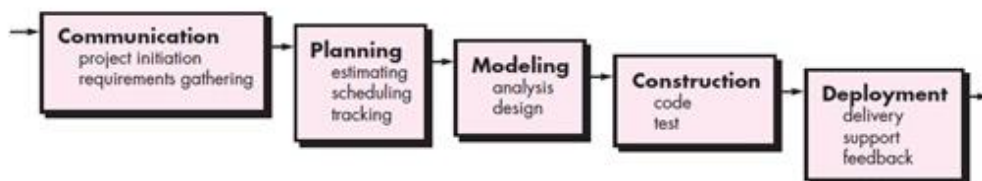
Secara umum Sistem Informasi bisa di artikan suatu sistem yang saling terintegrasi yang dapat menyediakan informasi yang bermanfaat bagi penggunanya. Sistem Informasi memiliki komponen – komponen dasar : perangkat keras komputer, perangkat lunak komputer, basis data, jaringan, prosedur, pengguna untuk pengelolaan operasi. Pengelola sistem Informasi memiliki tingkatan manajemen yang telah terstruktur. dapat disimpulkan Sistem informasi adalah suatu kombinasi modul yang terorganisir yang berasal dari komponen-komponen yang terkait dengan hardware, software, people dan network berdasarkan seperangkat komputer yang saling berhubungan atau berinteraksi untuk melakukan pengolahan data menjadi informasi untuk mencapai tujuan. [7]

#### 2.2 Metode Penelitian

Secara Umum, metode penelitian diartikan sebagai cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu.

#### 2.3 Waterfall Model

Waterfall Model adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Waterfall Model terdiri dari 5 tahapan untuk pengembangan. Berikut adalah penjelasan dari tahap – tahap yang dilakukan di dalam model ini menurut Pressman:



Gambar 1 waterfall model

#### 1. Communication (Project Initiation & Requirements Gathering)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi dari aplikasi. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, paper dan internet.

#### 2. Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

#### 3. Modeling (Analysis & Design)

Tahapan ini adalah tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur software, tampilan interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

#### 4. Construction (Code & Test)

Tahapan ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

#### 5. Deployment (Delivery, Support, Feedback)

Tahapan terakhir ini merupakan tahapan implementasi software ke customer, perbaikan software, evaluasi software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

## 2.4 Flowmap

Flowmap adalah gabungan antara peta dan flow chart, dengan menunjukkan pergerakan benda dari satu tahap ke tahap lain, seperti jumlah paket dalam jaringan, jumlah barang yang diperjualbelikan, ataupun jumlah orang dalam migrasi. Flowmap dapat menolong analis maupun programmer untuk memecahkan masalah ke dalam kelompok – kelompok yang lebih kecil dan dapat menolong dalam menganalisis alternatif-alternatif lain dalam pengoprasian sebuah sistem.

Jika seorang analis maupun programmer akan membuat flowmap, ada beberapa pedoman dasar yang harus diperhatikan, seperti :

1. Flowmap digambarkan dari halaman atas ke bawah dan kiri ke kanan dapat berbentuk vertical maupun horizontal.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati serta definisi aktivitas harus dapat di mengerti oleh pembacanya.
3. Tentukan secara jelas kapan aktivitas dimulai dan berakhir.
4. Setiap Langkah dari aktivitas harus berada pada urutan yang sesuai
5. Lingkup dari aktifitas yang sedang digambarkan harus di telusuri dengan hati-hati dan teliti
6. Agar mudah dimengerti maka gunakan symbol-simbol flowchart yang standar [9]

## 2.5 Use Case Diagram

### 1. Pengertian Use Case Diagram

Beberapa definisi use case diagram menurut beberapa ahli, sebagai berikut :

- a. Menurut (sukamto & Shalahuddin, 2018), “ Use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang dibuat”. Usecase mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang dibuat.
- b. Menurut (Tohari, 2014) “ Use case adalah rangkaian atau urutan sekelompok yang saling terkait dan membentuk sistem secara teratur

yang dilakukan atau diawasi oleh sebuah actor”. Use case digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.




- c. Menurut (Fowler,2005),” Use case adalah Teknik untuk merelam persyaratan fungsional sebuah sistem”. Usecase mendeskripsikan interakasitipikal antara pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Skenario adalah rangkaian Langkah-langkah yang menjabarkan sebuah interaksi antar seorang pengguna dengan sebuah sistem

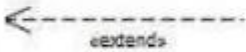

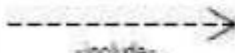
Sehingga dapat disimpulkan bahwa use case adalah urutan transaksi dari actor pada sistem. Use case dapat menggambarkan apa yang sistem lakukan dari perpektif user. Use case merupakan sekumpulan scenario yang menjadi satu untuk tujuan user.

Berikut beberapa keuntungan dari penggunaan Use case diagram :

- a. Menangkap kebutuhan fungsional dari perspektif user
- b. Memberi deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan sistem.
- c. Dasar membentuk tes sistem
- d. Menyediakan kemampuan untuk melacak kebutuhan fungsional ke dalam class actual dan operasi sistem
- e. Berfungsi sebagai unit estimasi
- f. Unit terkecil dari delivery.

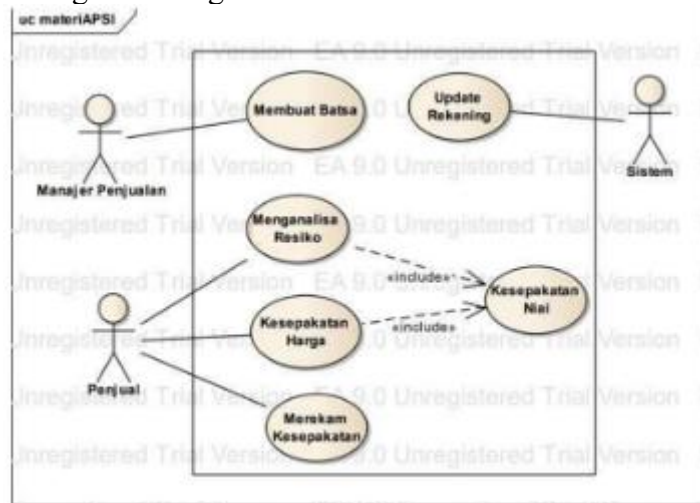
## 2. Simbol Use Case Diagram

Simbol	Keterangan
<p><i>Usecase</i></p> 	<p>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 <i>usecase</i>.</p>
<p><i>Actor/aktor</i></p> 	<p>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 <i>actor</i>.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>

<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>usecase</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; 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.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>usecase</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p><i>Include</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>usecase</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>.</p>

Gambar 2 Gambar Simbol Usecase

Contoh Use Case Diagram Sebagai Berikut :



Gambar 3 Contoh Use Case Diagram







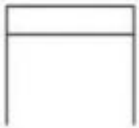
## 2.6 Activity Diagram

### 1. Pengertian Activity Diagram

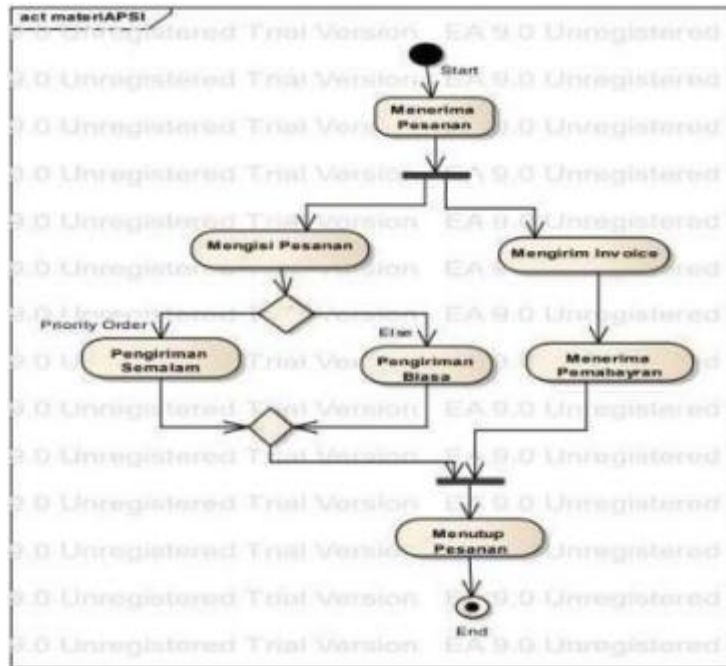
Beberapa pendapat mengenai Activity Diagram, Sebagai berikut :

- a. Menurut (Gata & Gata 2013) Activity diagram adalah “Menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis”.
- b. Menurut (Fowler,2005) Activity diagram adalah “ Teknik untuk menggambarkan logika procedural, proses bisnis”. Dalam beberapa hal mirip dengan sebuah diagram alir. Perbedaan antar kedua diagram tersebut adalah prinsip antara diagram activity dan no notasi diagram alir adalah diagram activity mendukung behavior paraller.

## 2. Simbol Activity Diagram

Simbol	Fungsi
	<i>Start point</i> , diletakan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activities</i> menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (percabangan), digunakan untuk menunjukan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>Rake</i> , digunakan untuk menunjukan adanya dekomposisi.
	<i>Decisions points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukan siapa melakukan apa.

Gambar 4 Simbol Activity Diagram



Gambar 5 Contoh Activity Diagram



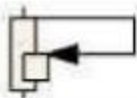


## 2.7 Sequence Diagram

### 1. Pengertian Sequence Diagram

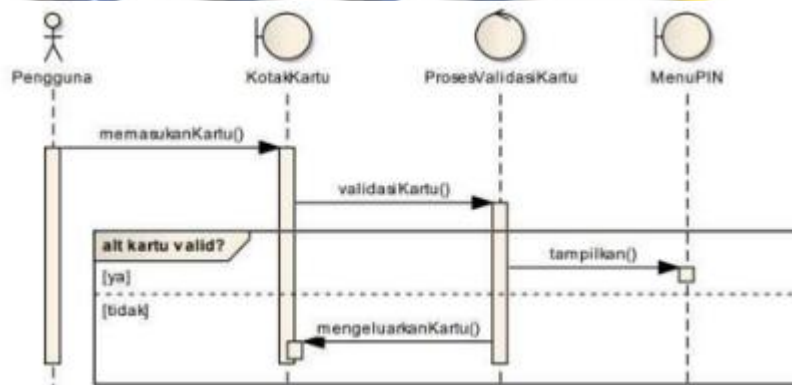
Menurut (Gata & Gata, 2013) “Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek”.

### 2. Simbol Sequence Diagram

Simbol	Fungsi
	<i>Entity class</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 <i>formentry</i> dan <i>form</i> cetak.

	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. <i>Control object</i> mengkoordinir pesan antara <i>boundary</i> dengan entitas.
	<i>Message</i> , simbol mengirim pesan antar class.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Gambar 6 Simbol Sequence Diagram



Gambar 7 Contoh Sequence Diagram

## 2.8 Class Diagram

Class Diagram dibuat setelah usecase dibuat terlebih dahulu. Pada diagram ini dijelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuklah suatu sistem aplikasi. Class Diagram ini adalah struktur database yang sedang dibuat seperti table login, table user, table katageri, table pemesanan\_item, table alamat, table keranjang dan table konfirmasi dapat dilihat pada gambar dibawah ini : [10]



Gambar 8 Class Diagram