

## **BAB II**

### **LANDASAN TEORI**

#### **II.1 Penelitian Terdahulu**

Marisa Eka Putra, dkk(2014) mempublikasikan penelitian mengenai Perancangan Aplikasi Pemesanan Tiket Mobil Berbasis *Android* Pada Agen Travel. Penelitian ini telah menghasilkan sebuah sistem aplikasi yang memiliki kelebihan untuk proses bisnis yang efektif dan efisien dan membahas layanan berbasis *mobile* untuk mempermudah hubungan antara pengguna dan agen travel supaya proses bisnis dapat lebih efektif dan efisien.

Robby Rachmatullah, dkk(2015) mempublikasikan penelitian mengenai Aplikasi Perancangan Sistem Pemesanan Tiket Bus Online Berbasis Web. Aplikasi yang dibangun bertujuan untuk mempromosikan perusahaan sekaligus dapat memberikan pelayanan kepada pelanggan mengenai pemesanan tiket secara *on-line* karena Saat ini pemakaian jasa internet sebagai sarana untuk memperoleh informasi semakin banyak digunakan dan jangkauannya yang luas maka internet sangat ideal bila digunakan sebagai sarana promosi dan sarana untuk mempublikasikan informasi.

Fajar Fani Hartono, dkk(2014) mempublikasikan penelitian mengenai Aplikasi Reservasi Tiket Bus pada Handphone *Android* menggunakan Web Service Studi Kasus: PO. Rosalia Indah, dari hasil penelitian dan pengujian yang telah dilakukan. Aplikasi reservasi tiket bus berbasis *Android* dapat diterapkan pada PO. Rosalia Indah untuk penjualan tiket seluruh trayek. Aplikasi *mobile*.

dengan tampilan yang sederhana dan user input yang tidak rumit dapat digunakan dalam melakukan reservasi tiket secara online. Aplikasi ini dapat membantu user menghemat waktu, biaya, dan tenaga dalam pembelian suatu produk, karena menggunakan *mobile device* yang bersifat *ubiquity* sehingga transaksi bisa dilakukan kapan saja dan dimana saja.

Surya Wirawan, dkk(2014) mempublikasikan mengenai Aplikasi Sistem Reservasi Tiket Bus Di Terminal Arjosari Malang. Pada aplikasi ini menggunakan SMS Gateway untuk mengirimkan *report* kepada pembeli setelah melakukan pemesanan tiket bus secara *online*. SMS Gateway pada aplikasi ini berfungsi sebagai penghubung yang melakukan *delay* SMS antara *External Short Message Entity* (ESME) dan *Short Message Service Center* (SMSC) serta sebaliknya. Komunikasi antara ESME dan SMS Gateway dapat menggunakan protokol *Short Message Peer to Peer* (SMPP) atau dengan HTTP, sementara ke SMSC menggunakan SMPP.

Nurul Huda (2015) mempublikasikan mengenai Aplikasi Pemesanan Tiket Bus Online dan Penyajian Informasi Angkutan Bus di Terminal Tingkir Salatiga Berbasis Website. Berdasarkan hasil pembuatan aplikasi pembelian tiket online dan penyajian informasi angkutan bus di Terminal Tingkir Salatiga berbasis website, bahwa sistem yang dibuat dapat membantu calon penumpang untuk mengetahui informasi umum suatu angkutan bus dan dapat membantu dalam hal pembelian tiket bus. Selain itu, rute yang dibuat juga dapat membantu calon penumpang untuk mengetahui rute yang dilalui oleh suatu angkutan bus.

## **II.2 Uraian Teoritis**

Adapun teori – teori yang digunakan sebagai landasan dalam penulisan skripsi ini adalah sebagai berikut :

### **II.2.1 *Reservasi (Pemesanan)***

Pemesanan dalam bahasa Inggris adalah Reservation yang berasal dari kata “to reserve” yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan reservation yaitu pemesanan suatu tempat fasilitas. Pengertian reservasi adalah sebuah proses perjanjian berupa pemesanan sebuah produk baik barang maupun jasa dimana pada saat itu telah terdapat kesepakatan antara konsumen dengan produsen mengenai produk tersebut namun belum ditutup oleh sebuah transaksi jual beli. Pada saat reservasi berlangsung biasanya ditandai dengan adanya proses tukar-menukar informasi antara konsumen dan produsen agar kesepakatan mengenai produk dapat terwujud. Alasan reservasi menjadi sebuah media yang sangat efektif baik bagi produsen maupun bagi konsumen adalah produsen akan dapat melakukan evaluasi terhadap produk yang akan mereka jual melalui tingkat tinggi rendahnya jumlah reservasi jauh sebelum produk tersebut dijual (barang) ataupun diselenggarakan (jasa), dimana hasil evaluasi tersebut akan membantu produsen untuk menentukan langkah pemasaran yang akan diambil terhadap produk yang akan dijual tersebut. Sedangkan bagi konsumen melalui media reservasi dapat menimbang terlebih dahulu sebelum membeli produk sampai dengan waktu yang telah ditentukan (*time limit*). Sampai dengan *time limit* yang telah ditentukan produk yang telah dipesan tersebut tidak boleh dijual kepada konsumen lain, karena secara tertulis maupun tidak sampai

dengan *time limit* produk tersebut telah diprioritaskan kepada konsumen yang telah melakukan reservasi. Hal ini akan sangat menguntungkan konsumen karena sampai dengan *time limit* yang ditentukan, konsumen dapat melakukan perbandingan dengan produk lain. Selain itu, jika terjadi sesuatu dan lain hal yang mengharuskan konsumen untuk batal membeli produk tersebut, konsumen yang bersangkutan tidak harus membeli produk tersebut. Layaknya perjanjian lainnya, reservasi akan dinyatakan batal apabila terjadi beberapa hal berikut :

1. Terjadi sesuatu di luar kendali manusia (bencana alam, perang, dll) sehingga produk tidak dapat dibuat atau diselenggarakan oleh produsen,
2. Konsumen melakukan pembatalan sebelum batas waktu,
3. Terdapat kesalahan pengisian data penting mengenai konsumen pada saat proses reservasi sehingga produsen harus membatalkan reservasi (catatan: tergantung kebijakan masing-masing perusahaan),
4. Tidak dilakukan transaksi terhadap produk yang dipesan sampai dengan *time limit* yang telah ditentukan,
5. Baik produsen maupun konsumen melakukan hal-hal yang menyebabkan reservasi harus dibatalkan, tergantung perjanjian yang dibuat/ditentukan (William Christanto, 2012).

## **II.2.2 Pengertian Tiket**

Tiket merupakan suatu alat/media yang digunakan oleh perusahaan tertentu sebagai pengganti uang langsung. Tiket biasanya berupa kertas yang didalamnya terdapat item-item tertentu yang menunjukkan suatu nilai.

Menurut Kamus Besar Bahasa Indonesia yang dimaksud dengan tiket adalah “sesuatu yang dianggap sebagai alat pembayaran yang digunakan oleh suatu alat transportasi yang ada.

### **II.2.3. Pengertian Tiket Bus**

Tiket bus adalah merupakan karcis yang harus dimiliki sebelum melakukan perjalanan dengan menggunakan alat transportasi berupa bus. Didalam tiket bus tersebut kita dapat melihat jurusan, jam keberangkatan, jam tiba di tempat tujuan dan nomor tempat duduk. Tiket bus bisa diperoleh dengan cara memesan sebelum hari keberangkatan atau bisa juga dengan cara membeli langsung pada agen penjualan tiket bus pada hari keberangkatan.

### **II.2.4 Pengertian Aplikasi**

Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi terdapat yang dirancang untuk menjalankan tugas tertentu.

Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rahmatillah ; 2011: 3).

### **II.2.5 *Android Studio***

*Android Studio* adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment (IDE)* untuk pengembangan aplikasi *Android*, berdasarkan *IntelliJ IDEA*. Konsep yang dimiliki *Android Inc* ternyata menggugah minat *Google* untuk memilikinya. Pada bulan Agustus 2005, Akhirnya *Android Inc* diakuisisi oleh *Google Inc*. seluruh sahamnya dibeli oleh *Google*. Banyak yang memperkirakan nilai pembelian *Android Inc* Oleh *Google* adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini adalah langkah awal yang dilakukan *Google* untuk masuk kepasar *mobile phone*. Maka dari situlah banyak pengembang sistem maupun software berlomba lomba untuk membuat atau merancang sistem *Android* menggunakan software – software yang support dengan *Android*, dan sebagai contoh disini kita pengenalan *Android Studio*. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi *Android*, misalnya:

1. Sistem versi berbasis *Gradle* yang *fleksibel*
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat *Android*.

4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. Template kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif.
7. Alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

Laman ini berisi pengantar dasar fitur-fitur *Android Studio*. Untuk memperoleh rangkuman perubahan terbaru, lihat Catatan Rilis *Android Studio* (*Google developer* ;2016).

## II.2.6 *Firebase*

*Firebase* merupakan sebuah layanan *infrastruktur backend-as-a-service (BaaS)* yang diakuisisi oleh *Google* pada Oktober 2014 silam. *Firebase* menawarkan kemudahan kepada para pengembang perangkat lunak dalam membangun aplikasi yang lebih baik serta mengembangkan bisnis yang sukses melalui seluruh fitur komplementernya. Saat ini, *Firebase* sudah memiliki fitur *Analytics, Cloud Messaging, Authentication, Realtime Database, Storage, Hosting, Test Lab, Crash Reporting, Notifications, Remote Config, App Indexing, Dynamic Links, Invites, AdWords, dan AdMob*. Seluruh fitur tersebut dikemas

dalam sebuah SDK *Firestore* tunggal sehingga dengan kemudahan yang ditawarkan para pengembang perangkat lunak dapat fokus untuk memecahkan masalah *customer* melalui perangkat lunak yang dibuatnya dan tidak menghabiskan banyak waktu dalam membangun infrastruktur yang kompleks (Leo Tiofan Justicia, 2017).

### **II.2.7. *Firestore User Authentication***

*Firestore User Authentication* merupakan sebuah layanan siap pakai yang dimiliki oleh *Firestore SDK*. *Firestore User Authentication* memungkinkan aplikasi untuk melakukan *user authentication* menggunakan berbagai macam metode authentication seperti *email and password based authentication*, *federated identity provider integration* (authentication menggunakan akun *Google*, *Facebook*, *Twitter* atau *Github*), *custom authentication system integration* hingga *anonymous authentication*. *Firestore User Authentication* ini bekerja dengan cara mengirimkan *server response* dari *Firestore Server* berdasarkan *credential* yang dikirimkan oleh *client* ke *Firestore Server*. *Credential* tersebut dapat berupa alamat *email* dan *password* ataupun sebuah token *OAuth* dari sebuah *federated identity provider*. Melalui *server response* yang diterima dari *Firestore Server*, aplikasi dapat mengakses informasi dasar profil pengguna dan mengontrol akses pengguna terhadap produk atau layanan *Firestore* yang terdapat pada aplikasi (Leo Tiofan Justicia, 2017).

### II.2.8 *Firestore Realtime Database*

*Firestore Realtime Database* merupakan sebuah layanan *NoSQL cloud-hosted database* yang dimiliki oleh *Firestore SDK*. Layanan ini menawarkan penyimpanan data yang dapat disinkronisasikan secara *realtime* terhadap seluruh *client* yang terhubung. Layanan ini memiliki 3 kemampuan inti yaitu *realtime*, *offline* dan *accessible from client devices* (Google, 2017). Maksud dari *realtime* adalah jika terdapat perubahan pada data pada *database*, maka seluruh *client* yang terhubung akan secara otomatis mendapatkan perubahannya dalam hitungan milidetik. Kemudian *offline*, yaitu aplikasi yang menggunakan *Firestore Realtime Database* akan tetap responsif bahkan saat *offline*. Hal ini disebabkan karena *Firestore SDK* dapat mempertahankan data dan perubahannya pada media penyimpanan *client*. Pada saat *client* terhubung ke jaringan internet, maka *Firestore SDK* akan melakukan penyesuaian otomatis atas catatan perubahan data yang disimpan pada media penyimpanan *client* dengan kondisi terkini dari *Firestore server*. Kemampuan inti yang terakhir adalah *accessible from client devices*. Layanan ini menawarkan kemudahan untuk mengakses *Firestore Realtime Database* secara langsung dari sebuah perangkat *mobile* atau sebuah peramban web tanpa membutuhkan *server application* (Leo Tiofan Justicia, 2017).

### II.2.9 *Android*

*Android* adalah suatu sistem operasi yang berbasis *Linux* dengan perangkat *Mobile device*, *Android* menyediakan *platform* terbuka bagi para pengembang

untuk menciptakan aplikasi-aplikasi baru untuk membantu kebutuhan manusia. Awal terdirinya *android* dimulai dari *Google Inc* yang membeli *Android inc* pendatang bat dari perangkat lunak untuk ponsel / *smartphone*. Semakin banyak *smartphone* dan *PC Tablet* menggunakan sistem operasi dengan versi yang berbdeda. Semakin tinggi versinya maka fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream* yang dirilis pada tanggal 22 Oktober 2008. Perkembangan sistem *Android* berkembang sesuai versi keluarnya adapun versi *Android* dimulai dari (Muhamad Zaelani ; 2017: 5) :

**Tabel. II.1 Versi Android**

Versi	Nama	Rillis	Catatan
1.0	<i>Android 1.0</i>	23 September 2008	<i>Android</i> pertama hanya untuk <i>smartphone</i>
1.1	<i>Android 1.1</i>	9 Februari 2008	
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama
1.6	<i>Donut</i>	15 September 2009	
2.0- 2.1	<i>Éclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	
2.2	<i>Froyo</i>	20 Mei 2010	
2.3	<i>Gingerbread</i>	6 Desember 2010	Digunakan pada <i>smartphone</i> jenis lama
3.0- 3.2	<i>Honeycomb</i>	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	<i>ICS (Ice Cream Sandwich)</i>	19 Oktober 2011	<i>Smartphone</i> dan tablet
4.1- 4.3	<i>Jelly Bean</i>	9 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	<i>Update</i> untuk memperbaiki dan menambah <i>fitur</i> pada <i>ICS</i>
4.4	<i>Kit kat</i>	3 September 2013	

**Tabel. II.2 Versi Android (Lanjutan)**

Versi	Nama	Rilis	Catatan
5.0	<i>Lollipop</i>	12 november 2014 (5.0) 9 Maret 2015 (5.1)	
6.0	<i>Marshmallow</i>	5 oktober 015	Terdapat <i>daze mode</i> , <i>Do Not Disturb mode</i> , mendukung <i>USB tipe C</i> , mendukung pembacaan <i>fingerprint</i> .
7.0	<i>Nougat</i>	22 Agustus 2016	Multitasking

Sumber: Muhamad Zaelani (2017: 5)

### II.2.10 UML (Unified Modelling Language)

*UML* merupakan kependekan dari *Unified Modeling Language* yaitu diagram dan metode standar untuk memodelkan dan merepresentasikan *object oriented software* dan sistem bisnis. (Mulyani, 2016 : 243).

Beberapa fungsi dan kegunaan dari *UML* yaitu (Mulyani, 2016 : 244) :

1. *Visualizing*

*Visualizing* adalah sebagai alat komunikasi konseptual model antara tim pengembang sistem (sistem analis dengan programmer)

2. *Specifying*

*Specifying* adalah sebagai *tools* yang digunakan untuk memodelkan sistem secara tepat dan jelas.

3. *Constructing*

*Constructing* adalah *UML* sebagai bahasa grafis mampu melakukan *mapping* dan konseptual model kedalam bahasa pemrograman.

#### 4. Documenting

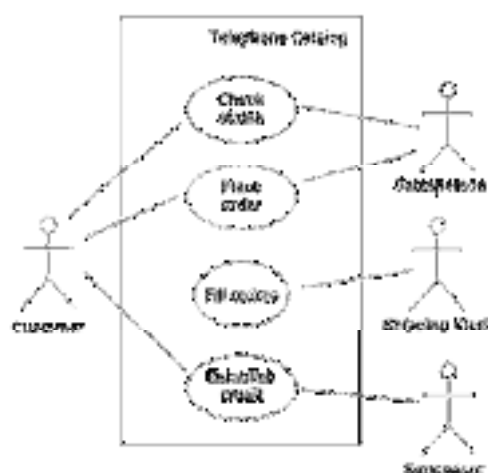
*Documenting* adalah *UML* digunakan sebagai *tools* untuk melakukan dokumentasi teknis sebuah sistem.

Diagram-diagram yang terdapat dalam *UML* sangat banyak, berikut ini beberapa diagram yang sering digunakan dalam pengembangan sistem yaitu :

##### 1. Use Case Model

*Use case model* merupakan kumpulan diagram dan text yang saling bekerja sama untuk mendokumentasikan bagaimana *user* (aktor) berinteraksi dengan sistem. *Use case model* terdiri dari beberapa diagram :

a) *Use case diagram* yaitu diagram yang menggambarkan dan merepresentasikan aktor, *use cases*, dan *dependencies* suatu proyek dimana tujuan dan diagram ini adalah untuk menjelaskan konsep hubungan antara sistem dengan dunia luar. *Use case diagram* dapat dilihat pada gambar II.1 seperti berikut.



**Gambar II.1 Use Case Diagram**  
(Sumber : Mulyani ; 2016)

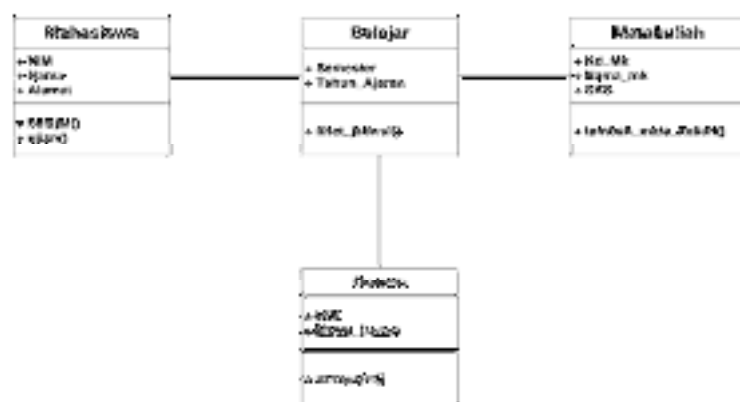
b) *Use case narrative* yaitu deskripsi yang menjelaskan use case diagram.

Pada *use case diagram* sistem hanya digambarkan secara sederhana menggunakan simbol *use case* yang berhubungan (*relationship*) dengan aktor, sehingga terkadang diperlukan deskripsi yang menjelaskan dan proses tersebut.

c) *Use case scenario* yaitu pemecahan kemungkinan logika pada *use case diagram*.

## 2. *Class Diagram*

*Class diagram* adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan antara masing-masing kelas serta mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan *property* dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. *UML* menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah kelas. *Class diagram* dapat dilihat pada gambar II.2 seperti berikut.



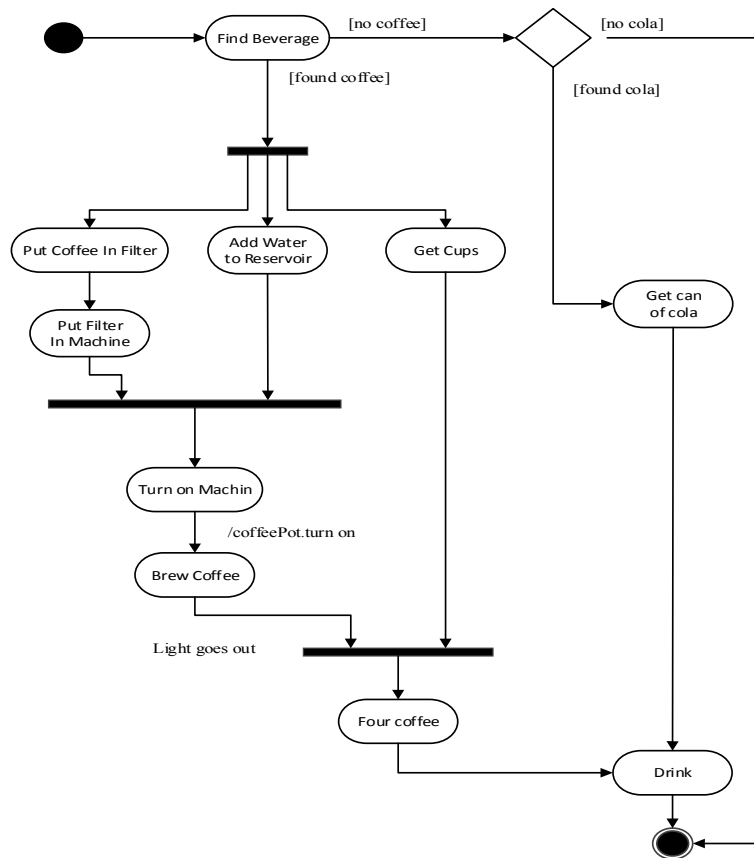
**Gambar II.2 *Class Diagram***  
(Sumber : Mulyani ; 2016)

### 3. *Object Diagram*

*Object diagram* adalah diagram yang digunakan untuk menggambarkan dan merepresentasikan objek dan hubungan antar objek tersebut, selain itu *object diagram* juga dapat digunakan sebagai diagram untuk menunjukkan sebuah konfigurasi contoh dan sebuah objek.

### 4. *Activity Diagram*

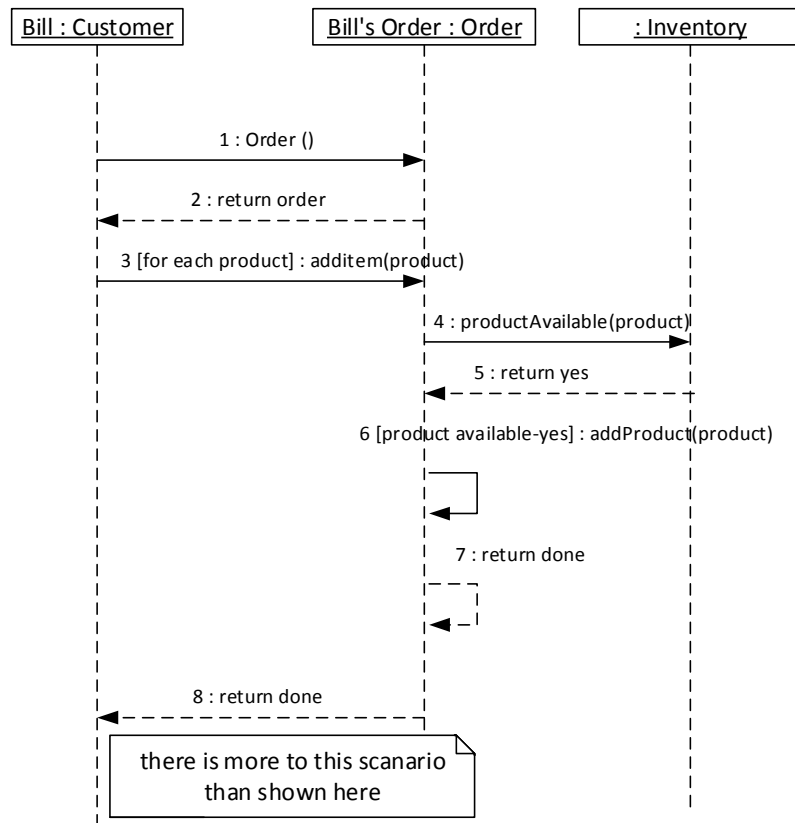
*Activity diagram* adalah diagram *UML* yang digunakan untuk menggambarkan alur aktivitas dari satu proses. *Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya. Hal ini penting untuk pemodelan bisnis karena proses-proses sering muncul secara paralel. Ini juga berguna pada algoritma yang bersamaan, dimana urutan-urutan independen dapat melakukan hal-hal secara paralel. *Activity diagram* dapat dilihat pada gambar II.3 seperti berikut.



**Gambar II.3 Activity Diagram**  
(Sumber : Mulyani ; 2016)

### 5. Sequence Diagram

*Sequence diagram* adalah diagram yang menggambarkan interaksi antar objek. *Sequence diagram* secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam sebuah *use case*.



**Gambar II.4 Sequence Diagram**  
*(Sumber : Mulyani ; 2016)*