

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Pengertian sistem menurut Anastasia Diana, dkk (2011 : 3) adalah serangkaian bagian yang saling tergantung dan bekerjasama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerjasama untuk mencapai tujuan. Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri.

#### **II.2. Pengertian Informasi**

Menurut Agus Mulyanto (2009 : 12), “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata”.

#### **II.3. Pengertian Sistem Informasi**

Sistem Informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen baik manual maupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut (Anastasia Diana, dkk ; 2011 : 4).

#### II.4. Pengertian Sistem Informasi Akuntansi

Maretha, dkk., (2008: 1), menyatakan sebagian besar akuntansi menggunakan informasi kuantitatif dimana kegiatannya terdiri dari pengumpulan dan pengolahan data keuangan suatu unit organisasi dan mengkomunikasikan hasil olahannya kepada pihak yang berkepentingan dalam mengambil keputusan yang ekonomik. Sedangkan informasi kualitatif dalam akuntansi biasanya berupa catatan atas laporan keuangan. Akuntansi yaitu sistem informasi yang mengukur aktivitas bisnis, memroses data menjadi laporan, dan mengomunikasikan hasilnya kepada para pengambil keputusan. Akuntansi saat ini sering diartikan sebagai “bahasa bisnis” atau lebih tepat jika disebut “bahasa pengambilan keputusan”. Semakin kita menguasai bahasa ini akan semakin baik pula kita menangani berbagai aspek keuangan dalam kehidupan ini. Apapun peranan kita dalam masyarakat, pasti kita pernah mengambil keputusan yang berhubungan dengan aspek keuangan, baik sebagai manajer, investor, politisi, kepala rumah tangga, atau mahasiswa (Horngren dan Harrison, 2007: 4).

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu *input* Sistem Informasi Akuntansi pada sebuah toko baju adalah transaksi penjualan. Kita memproses transaksi dengan mencatat penjualan tersebut ke dalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian secara periodik Sistem

Informasi Akuntansi akan menghasilkan *output* berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi (Anastasia Diana, dkk ; 2011 : 4 : 5).

## II.5. Pengertian Akumulasi Penyusutan

Akumulasi penyusutan adalah jumlah total akumulasi penyusutan yang diakui oleh organisasi atau perusahaan, dari sejak pembelian. Penyusutan (*deprisasi*) adalah proses penyusutan aktiva/aset dapat dihitung dengan cara menghitungnya berdasarkan umur ekonomis barang tersebut dapat menggunakan metode garis lurus ( Eko Hari Atmoko ; 2013).

## II.6. Pengertian Metode Garis Lurus (*Straight Line Method*)

Metode Depresiasi Garis Lurus adalah metode depresiasi dimana depresiasi berupa bagian-bagian yang sama besar selama masa manfaat yang ditetapkan bagi harta tersebut (Anastasia Diana, dkk ; 2011: 241). Formula untuk menghitung depresiasi garis lurus adalah sebagai berikut :

$$\text{Beban depresiasi per tahun} = \frac{\text{Harga Perolehan Aktiva Tetap} - \text{Nilai Residu}}{\text{Umur Ekonomis}}$$

Umur ekonomis adalah estimasi umur aktiva. Jadi kira-kira berapa lama aktiva tersebut akan bermanfaat bagi perusahaan secara efisien. Sedangkan nilai residu adalah estimasi harga jual aktiva setelah umur ekonomis aktiva tersebut habis (Anastasia Diana, dkk ; 2011 : 242).

## II.7. Pengertian Visual Basic 2010

Program *Visual Basic* 2010 merupakan salah satu paket bahasa pemrograman dari *Visual Studio* 2010. Banyak fasilitas yang akan kita dapatkan melalui rilis *Visual Basic* versi ini. *Visual Studio* 2010 merupakan bahasa pemrograman generasi ke tiga dari *Microsoft* dengan *IDE (Integrated Development Environment)* atau pemrograman pengembangan terpadu, *Visual Basic* dibuat dan dirancang untuk mudah digunakan baik oleh programmer pemula sekalipun (Eko Hari Atmoko ; 2013).

## II.8. Pengertian SQL Server

*SQL (Structured Query Language)* adalah sebuah *relasional database management* sistem buatan *Microsoft* yang dirancang untuk mendukung program dengan arsitektur *client/server*, dimana *database* diletakkan pada komputer pusat yang disebut *server*, dan informasi digunakan bersama-sama oleh pengguna yang menjalankan program di dalam komputer yang disebut *client* (Eko Hari Atmoko ; 2013).

## II.9. Konsep Dasar Perancangan Sistem

Konsep dasar perancangan sistem adalah analisis dari siklus pengembangan sistem, pendefinisian dari kebutuhan-kebutuhan fungsional, persiapan untuk rancang bangun implementasi, menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah serta menkonfigurasi dari

### II.9.1. Basis Data

Basis Data adalah sebuah metode untuk menyimpan dan mengambil lagi data tersebut saat dibutuhkan. Data adalah bagian dari informasi. Data yang tidak teratur tidak mempunyai fungsi apa-apa. Sebuah basis data menyediakan cara untuk mengatur data-data dengan aturan tertentu tidak ada bedanya dengan rak buku perpustakaan yang mengatur buku berdasarkan nomor bukunya.

Sistem basis data mempunyai banyak model, salah satunya adalah basis data relasional. Basis Data relasional memilah-milah data dalam bentuk tabel dua dimensi. Tiap tabel terdiri dari lajur mendatar disebut dengan Row/Record dan lajur vertikal yang disebut Coloum/Field. Pada tiap pertemuan lajur-lajur data tersebut item-item data ditempatkan. Tabel merupakan bentuk alamiah untuk menyatakan fakta yang sering kita gunakan sehingga bentuk inilah yang sering kita gunakan. (Yuni Sugiarti ; 2013 : 24).

#### II.9.1.1. Kamus Data

Kamus data atau data *dictionary* adalah catalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Selama penyusunan suatu sistem informasi, kamus data digunakan sebagai alat unuk mendefinisikan aliran data yang mengalir di sistem, merancang *input*, merancang laporan-laporan dan merancang *database*.

Kamus data dibuat berdasarkan arus data yang ada di *Data Flow Diagram*. Struktur dari suatu arus data di *Data Flow Diagram* dapat dilihat secara lebih terinci di kamus data. ((Yuni Sugiarti ; 2013 : 25).

### II.9.1.2. Normalisasi

Normalisasi adalah suatu proses formal untuk menentukan atribut-atribut yang seharusnya dikelompokkan secara bersama-sama dalam suatu relasi. Bentuk normalisasi adalah suatu aturan yang dikenakan pada relasi-relasi dalam basis data dan harus dipenuhi oleh relasi-relasi tersebut pada langkah-langkah normalisasi.

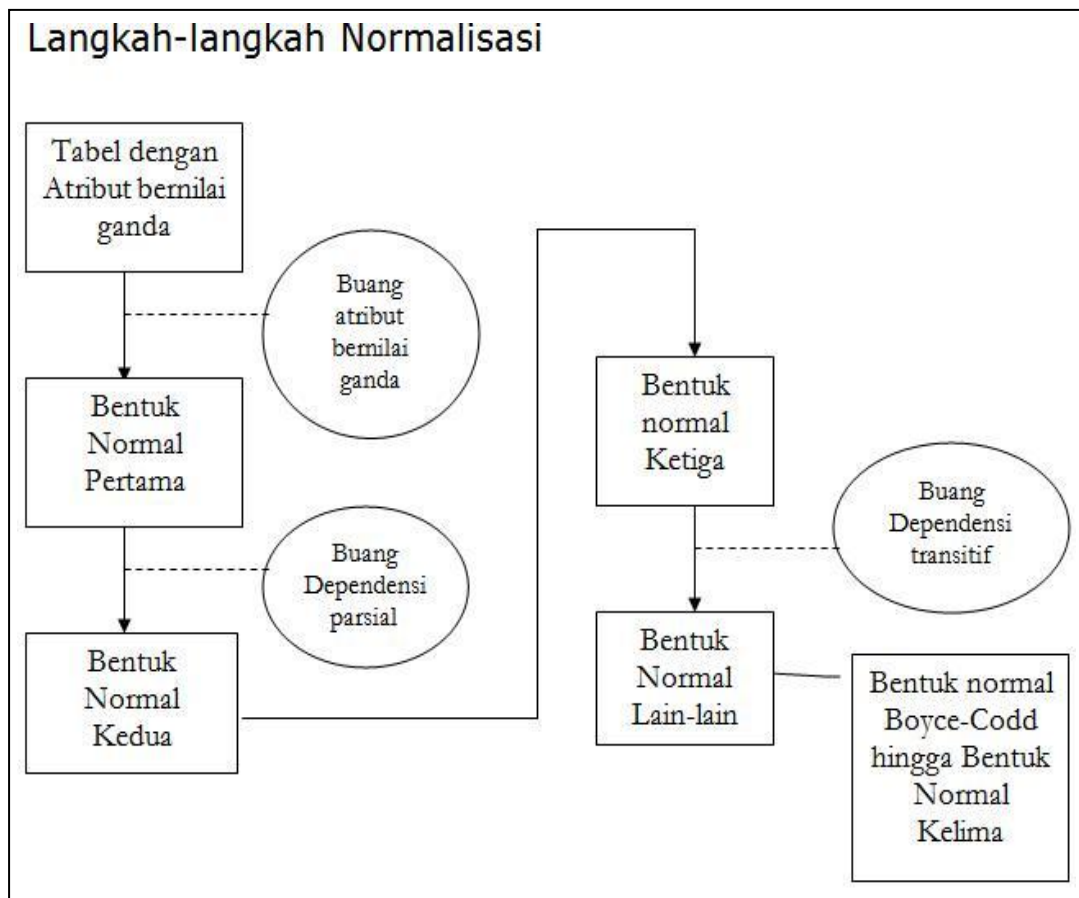
Alasan utama diperlukannya normalisasi karena permasalahan redundansi adalah :

1. Pemborosan ruang penyimpanan
2. Anomali pada saat insert (simpan), Update (edit/pembaharuan), Delete (hapus)
3. Dapat menyebabkan *inkonsistensi*

Ada 6 (enam) tingkat bentuk dalam normalisasi :

1. Bentuk Normal pertama (1NF)
2. Bentuk Normal Kedua (2NF)
3. Bentuk Normal Ketiga (3NF)
4. Bentuk Normal Boyce-Codd (BCNF)
5. Bentuk Normal Keempat (4NF)
6. Bentuk Normal Kelima (5NF)

Langkah-langkah dalam normalisasi dapat dilihat pada gambar II.1



**Gambar II.1. Langkah-Langkah Normalisasi**






**Sumber : Yuni Sugiarti (2013 : 47)**

### II.9.1.3. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi entitas yang menjelaskan data dan hubungan-hubungan antara data.

Simbol-simbol dalam *Entity Relationship Diagram* (ERD) dapat dilihat pada tabel II.1

**Tabel II.1. Simbol-Simbol *Entity Relationship Diagram* (ERD)**

Notasi	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak, dan banyak ke banyak.
	Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasi.
	Input/output data, yaitu proses input/output data, parameter, informasi.

**Sumber : Yuni Sugiarti (2013 : 54)**

*Entity Relationship Diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *Entity Relationship Diagram* (ERD) untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data, yaitu :

1. Satu ke satu (*One to one*)

Hubungan relasi satu kesatu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2. Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan B.

### **II.9.2. Pengertian UML (*Unified Modelling Language*)**

Sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku (Yuni Sugiarti ; 2013 : 34).

*UML* berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan

menjadi diagram. *UML* mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun.






### II.9.2.1. Use Case Diagram

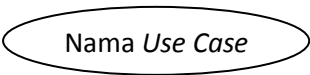
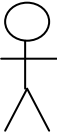

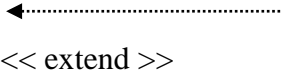
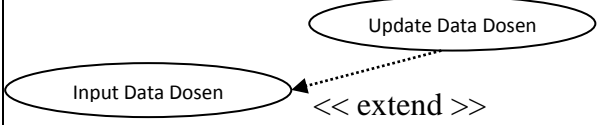
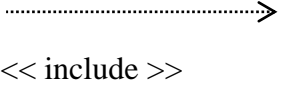
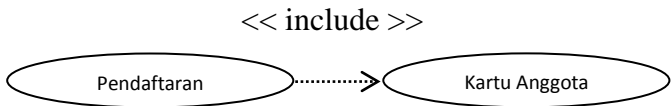
Menggambarkan sejumlah eksternal actor dan hubungannya ke *Use Case* yang diberikan oleh sistem. *Use Case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bagaimana fungsi yang ada didalam sistem.

#### 1. Simbol-simbol pada *Use Case Diagram*

Simbol-simbol *use case diagram* dapat dilihat pada tabel II.2 dibawah ini :

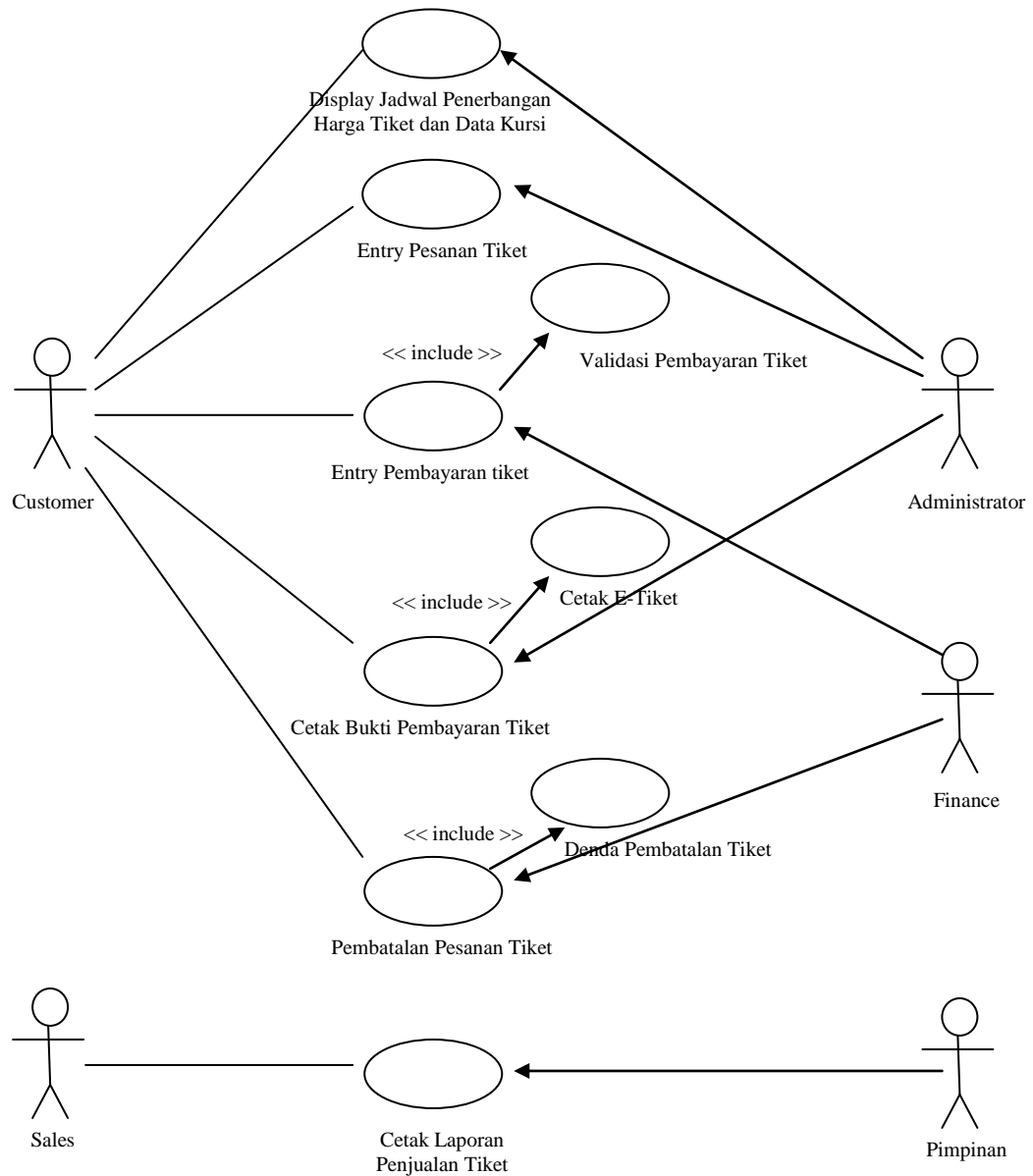
**Tabel II.2. Simbol-Simbol *Use Case Diagram***

Notasi	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak, dan banyak ke banyak.
	Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasi.
	Input/output data, yaitu proses input/output data, parameter, informasi.

Simbol	Deskripsi
<p data-bbox="288 412 427 448"><i>Use Case</i></p> 	<p data-bbox="663 412 1366 600">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>use case</i>.</p>
<p data-bbox="288 600 379 636">Aktor</p>  <p data-bbox="395 748 560 784">Nama Aktor</p>	<p data-bbox="663 600 1366 788">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.</p>
<p data-bbox="288 788 587 824">Asosiasi / <i>Assosiation</i></p> 	<p data-bbox="663 788 1366 931">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="288 931 395 967"><i>Extend</i></p> 	<p data-bbox="663 931 1366 1155">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, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, arah panah menunjuk pada <i>use case</i> yang dituju. Contoh :</p> 
<p data-bbox="288 1370 400 1406"><i>Include</i></p> 	<p data-bbox="663 1370 1366 1671">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>, <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. Contoh :</p> 

Sumber : Yuni Sugiarti (2013 : 42)

2. Contoh *use case diagram* pada sistem pemesanan tiket pesawat



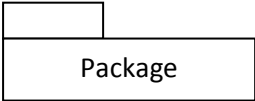
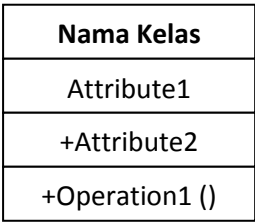


**Gambar II.2. Contoh Use Case Diagram**


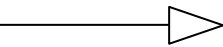

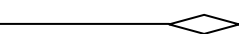
**Sumber : Yuni Sugiarti (2013 : 55)**

### II.9.2.2. Class Diagram

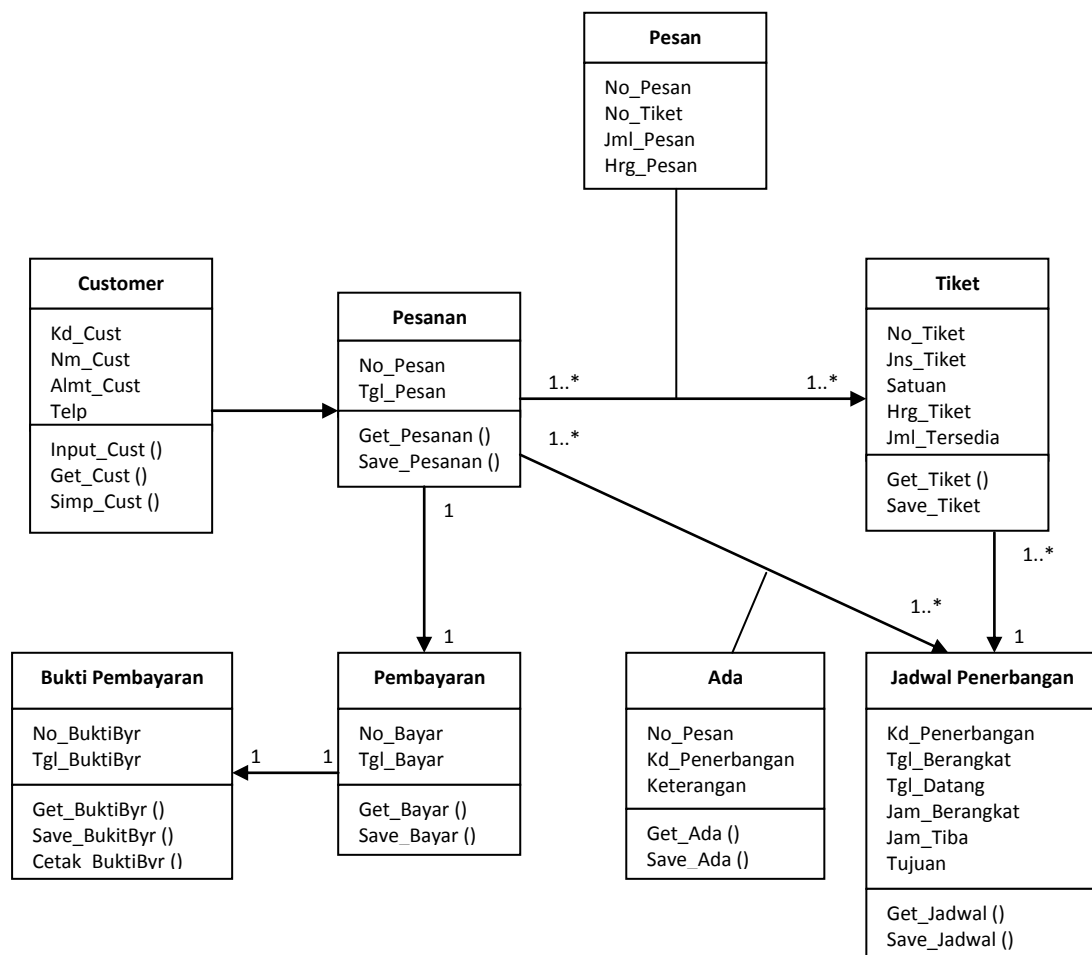
Menggambarakan struktur statis *class* dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat dihubungkan dengan lainnya melalui sejumlah cara : *assasiated* (terhubung satu dengan yang lain), *dependent* (satu *class* tergantung / menggunakan *class* yang lainnya), *specialiized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *packaged* (grup bersama sebagai suatu unit). Berikut adalah simbol-simbol yang ada pada *Class Diagram* :

**Tabel II.3. Simbol-simbol Class Diagram**

Simbol	Deskripsi
<p><i>Package</i></p>  <p>Package</p>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih kelas.
<p>Operasi</p>  <p>Nama Kelas</p> <p>Attribute1</p> <p>+Attribute2</p> <p>+Operation1 ()</p>	Kelas pada struktur sistem
<p>Antarmuka / <i>Interface</i></p>  <p>Interface</p>	Sama dengan konsep <i>Interface</i> dalam pemograman berorientasi objek.
<p>Asosiasi</p>  <p>1                      1..*</p>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>Directed</i> asosiasi</p>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya

	juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan/ <i>Defedency</i> 	Relasi antar kelas dengan makna bergantung antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (whole-part).

Sumber : Yuni Sugiarti (2013 : 59)



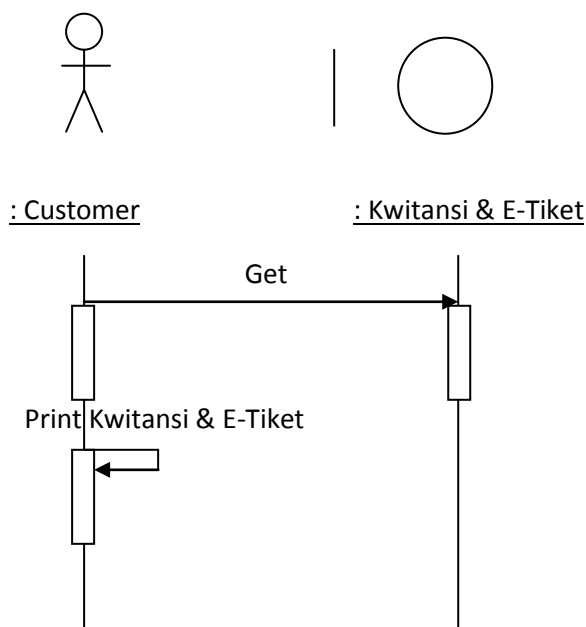
Gambar II.3. Contoh *Class Diagram*

Sumber : Yuni Sugiarti (2013 : 64)

### II.9.2.3 *Sequence Diagram*

Diagram *Sequence* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar Diagram *Sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya Diagram *Sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada Diagram *Sequence* sehingga semakin banyak *use case* yang didefinisikan maka Diagram *Sequence* yang harus dibuat juga semakin banyak (Yuni Sugiarti ; 2013 : 69).

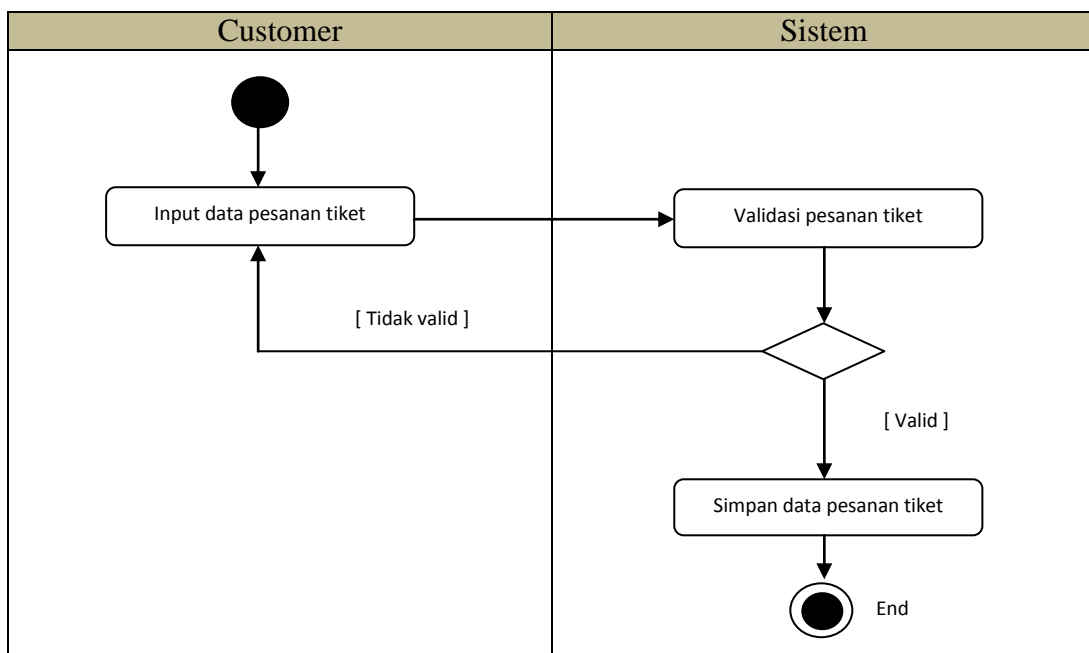


**Gambar II.4. Contoh Diagram *Sequence***

**Sumber : Yuni Sugiarti (2013 : 70)**

#### II.9.2.4. Activity Diagram

Diagram Aktivitas atau *Activity Diagram* menggambarkan *Workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.



**Gambar II.5. Contoh Activity Diagram**

**Sumber : Yuni Sugiarti (2013 : 78)**