

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Informasi**

##### **II.1.1. Konsep Dasar Sistem**

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh *Kenneth Boulding*, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem.

Selain itu, suatu sistem tidak bisa lepas dari lingkungan sekitarnya maka umpan balik atau *feedback* dapat berasal dari lingkungan sistem yang dimaksud. Organisasi dipandang sebagai suatu sistem yang tentunya akan memiliki semua unsur ini (Tata Sutabri ; 2012 : 9-10).

##### **II.1.2 Konsep Dasar Informasi**

Secara umum Informasi dapat didefinisikan sebagai data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerima nya. (Tata Sutabri ; 2012 : 29).

### **II.1.3 Konsep Dasar Sistem Informasi**

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem informasi akuntansi terdiri dari komponen- komponen yang disebut blok bangunan atau *building block*, yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran (Tata Sutabri ; 2012 : 46).

## **II.2. Sistem Informasi Akuntansi**

Analog dengan defenisi sebelumnya, sistem informasi akuntansi (SIA) adalah sistem berbasis komputer yang dirancang untuk mentransformasi data akuntansi menjadi informasi. Namun, dalam buku ini, istilah *sistem informasi akuntansi* memiliki cakupan yang sanga luas, yaitu mencakup juga siklus pemrosesan transaksi, pengguna teknologi informasi, dan pengembangan sistem informasi (George H. Bodnar ; 2006 : 8).

### **II.2.1. Pengertian Pajak**

Banyak defenisi pajak yang telah dikemukakan, oleh para pakar yang sama, dan ada juga yang lain dan pada dasarnya memilih tujuan yang sama yaitu

merumuskan pengertian pajak sehingga mudah di mengerti dan di pahami oleh masing-masing pihak yang saat merumuskan pajak, dalam bukunya (Waluyo.B.R.Ilyas ; 2002:4), salah satu untuk mewujudkan kemandirian suatu bangsa atau negara dalam pembiayaan pembangunan yaitu menggali sumber dana yang bersal dari dalam negeri berupa pajak, pajak dipergunakan untuk kepentingan bangunan yang digunakan untuk bersama sedangkan, dalam Pengantar Hukum Ilmu Hukum Perpajakan (R.S.Brotodiharjo;1991:2) menyatakan, pajak adalah iuran kepada negara (yang dapat dipaksakan), yang terhutang dan wajib membayarnya menurut peraturan, dengan tidak mendapat prestasi kembali yang langsung dapat ditunjuk dan yang gunanya adalah untuk membiayai pengeluaran-pengeluaran umum, berhubung dengan tugas negara yang menyelenggarakan pemerintah.

Dari definisi diatas dapat diketahui, ciri-ciri yang melekat dalam pengertian diatas yakni:

1. Pajak dipungut berdasarkan undang-undang serta peraturan pelaksanaannya yang sifat nya dapat di paksakan.
2. Dalam pembayaran pajak tidak dapat ditunjukkan adanya kontraprestasi individual oleh pemerintah
3. Pajak dipungut oleh negara baik pemerintah daerah atau Pusat
4. Pajak diperuntukan bagi pengeluaran-pengeluaran pemerintah.
5. Pajak dapat pula mempunyai tujuan selain *budgeter*, yaitu mengatur

### **II.2.2. Pajak Pertambahan Nilai Barang Mewah (PPnBM)**

Untuk dapat memahami pengertian PPN, terlebih dahulu kita perlu mengetahui pengertian Pertambahan Nilai. (Judisseno ; 2004 : 128), mengemukakan definisi Pertambahan Nilai adalah jumlah antara biaya dikeluarkan dan tingkat laba yang diharapkan dalam suatu proses produksi. Artinya proses pertambahan Nilai timbul karena adanya biaya produksi yang dikeluarkan mulai dari bahan hingga menjadi barang jadi yang siap dan dijual dengan tingkat laba yang diharapkan.(Sukardji ; 2009 : 10), mengemukakan definisi Nilai tambah (*added value*) adalah penjumlahan unsur unsur biaya dan laba dalam rangka proses produksi atau distribusi barang atau jasa. Jadi nilai tambah tidak semata-mata dihasilkan dari perubahan bentuk atau sifat suatu barang dalam kegiatan produksi.

Pajak pertambahan nilai (PPN) mulai diberlakukan di Indonesia pada tanggal 1 April 1985 untuk menggantikan pajak penjualan (PPn) yang sudah berlaku sejak tahun 1951. Dalam perspektif hukum, PPN merupakan suatu jenis pajak yang menempatkan kedudukan pemikul beban pajak sebagai penanggung jawab pembayaran pajak ke kas Negara, apabila pembeli atau penerima jasa sudah membayar PPN kepada penjual atau pengusaha, berarti mereka sudah membayar PPN ke kas Negara. Apabila pembeli atau penerima jasa sudah membayar PPN kepada penjual atau pengusaha, berarti mereka sudah membayar PPN ke kas Negara (Untung ; 2010 : 12).

### II.3. Visual Basic 2010

Visual basic 2010 merupakan lingkungan pengembangan terintegrasi atau bisa disebut IDE yang dikembangkan bahasa pemrograman *BASIC*. Bahasa *BASIC* sendiri sebenarnya sudah lama dibuat dan dikembangkan oleh microsoft corporation dengan nama *Microsoft Quit Basic*. Kesederhanaan sintaks dan fleksibilitas bahasa Basic dengan menyebabkan bahasa pemrograman ini begitu fenomenal sehingga banyak disukai dan dipakai oleh programmer di seluruh dunia.

Berbekal kepopuleran tersebut, Microsoft mengembangkan bahasa Basic ini menjadi produk yang sangat terkenal di kalangan programmer, yaitu mulai Microsoft Basic 6.0 sampai sekarang, yaitu Microsoft Visual Basic 2010. Perkembangan teknologi dan penambahan banyak sekali fitur Pada Visual Basic 2010 tidak mengakibatkan adanya perubahan sintaks – sintaks dasar yang terdapat didalamnya. Sehingga dapat dikatakan, untuk dapat menjadi programmer Visual Basic 2010 yang sebenarnya anda diharuskan menguasai dan mengerti bagaimana menggunakan dan mengimplementasikan sintaks dasar dalam bahasa pemrograman Basic (Wahana Komputer ; 2010 : 36).

### II.4. Basis data

Basis Data, Menurut *Stephens* dan *Plew* (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. dengan basis data, pengguna dapat

menyimpan data secara terorganisir. Setelah data disimpan informasi harus mudah diambil.

Kemudian, Silberschatz, dkk., (2002) mendefinisikan basis data sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem manajemen basis data (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data.

*Ramakrishnan* dan *Gehkre* (2003) menyatakan basis data sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.

Defenisi Basis data, menurut *Mcleod*, dkk., (2001) adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi. Sistem manajemen basis data adalah aplikasi perangkat lunak yang menyimpan struktur basis data, hubungan antar data dalam basis data, serta berbagai formulir dan laporan yang berkaitan dengan basis data. Basis data yang dikendalikan oleh sistem manajemen basis data adalah suatu set catatan data yang berhubungan dan saling menjelaskan (Janner Simarmata ; 2010 : 2).

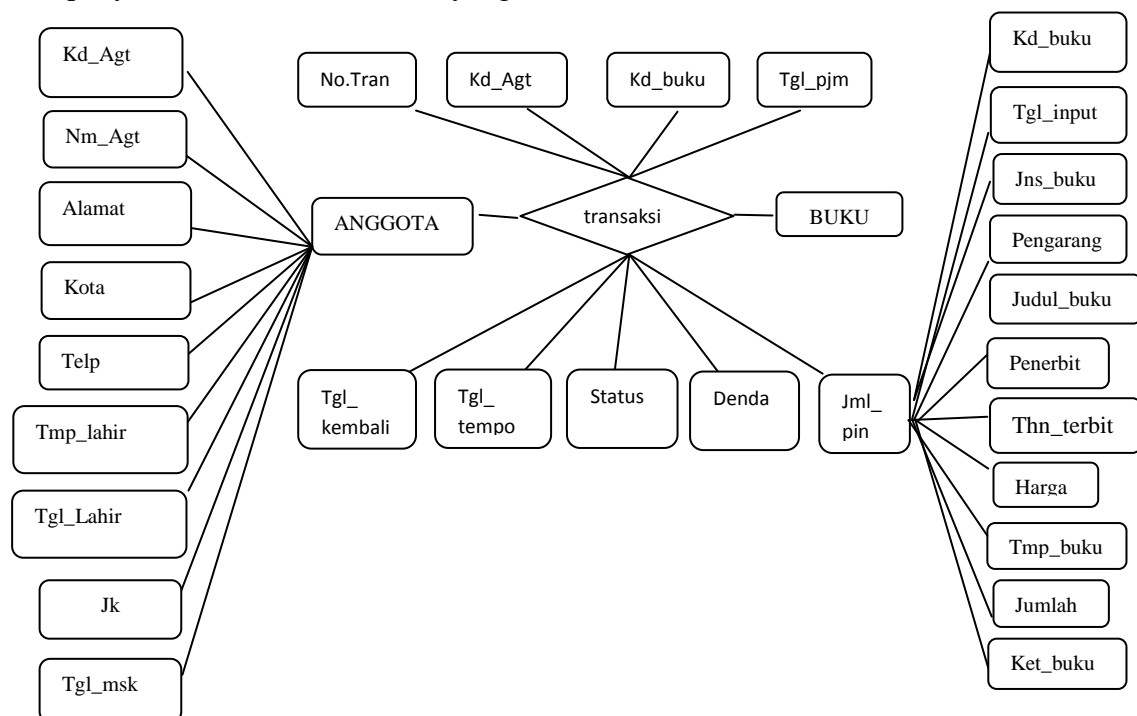
## **II.5. SQL Server 2008 Express**

SQL Server adalah Sebuah terobosan baru dari microsoft dalam bidang database. SQL Sever adalah sebuah DBMS (Database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolah data menyusul pendahulunya seperti IBM dan Oracle. SQL Server

2008, dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana computer ; 2010:2).

### II.5.1. Entity Relationship Diagram (ERD)

Pada dasarnya ERD (*Entity Relationship Diagram*) sebuah diagram yang secara konseptual, memetakan hubungan antara penyimpanan, ERD ini digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan ERD, ini dilakukan untuk mengurangi tingkat kesulitan dalam penyusunan sebuah data base yang baik.



**Gambar II.1 Entity Relationship Diagram**

**Sumber : Wahana Komputer ; 2010 : 31**

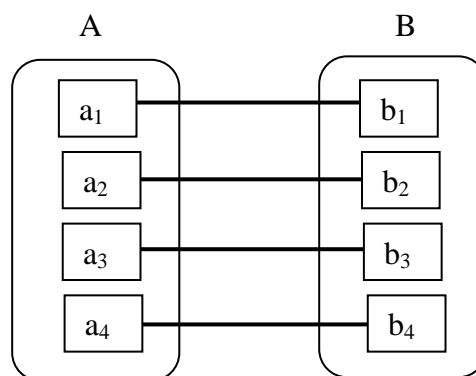
## II.5.2. Pemetaan Kardinalitas

Pemetaan kardinalitas menyatakan jumlah entitas dimana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi.

Pemetaan kardinalitas sangat berguna dalam menentukan himpunan relasi biner meskipun pemetaan dapat berperan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

Untuk suatu himpunan relasi biner  $R$  antara himpunan entitas  $A$  dan  $B$ , pemetaan kardinalitas harus salah satu dari berikut:

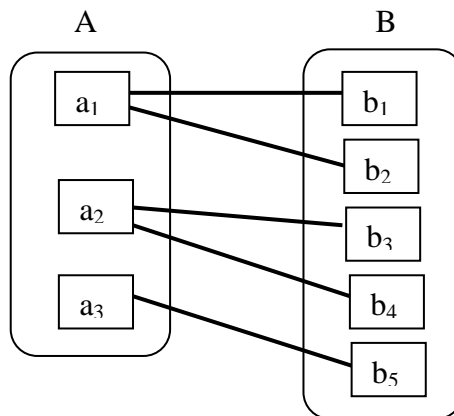
- a. *One-to-One*, sebuah entitas  $A$  berhubungan dengan paling banyak satu entitas pada  $B$  dan sebuah entitas pada  $B$  berhubungan dengan paling banyak satu entitas pada  $A$ .



**Gambar II.2 Hubungan one-to-one**

**Sumber : Janner Simarmata ; 2010: 64**

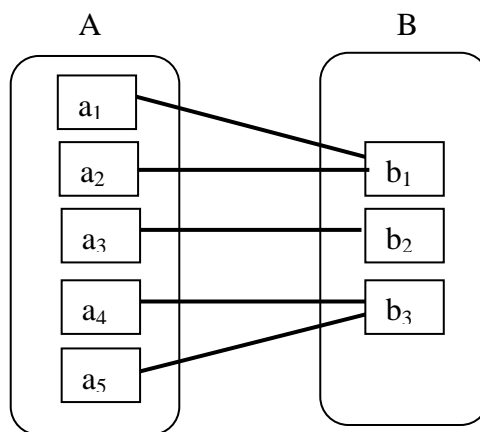
- b. *One-to Many*, sebuah entitas pada  $A$  berhubungan dengan nol atau lebih entitas pada  $B$ . Sebuah entitas pada  $B$  dapat dihubungkan dengan paling banyak satu entitas pada  $A$ .



**Gambar II.3 Hubungan one-to-many**

**Sumber : Janner Simarmata ; 2010: 64**

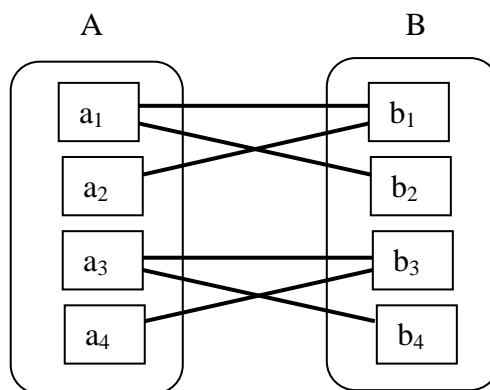
- c. *Many-to-One*, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



**Gambar II.4 Hubungan many-to-one**

**Sumber : Janner Simarmata ; 2010: 65**

- d. *Many-to-Many*, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



**Gambar II.5 Hubungan many-to-many**

**Sumber : Janner Simarmata ; 2010: 66**

### II.5.3. *Data Defenition Language (DDL)*

Skema basis data ditentukan sekumpulan defenisi yang dinyatakan dengan bahasa tertentu yang disebut *Data Defenition Language (DDL)*. Struktur penyimpanan dan metode akses yang digunakan oleh sistem basis data dengan menggunakan sekumpulan pernyataan dalam tipe DDL tertentu disebut bahasa penyimpanan dan defenisi data. pernyataan ini menentukan penerapan detail skema basis data, yang sering disembunyikan dari pengguna. Contohnya :*Create table rekening, (no\_rekening char(10),Saldo integer)* (Janner Simarmata ; 2010 : 38).

#### **II.5.4. Tabel**

Tabel adalah unit penyimpanan fisik utama untuk data dalam basis data. Ketika mengakses basis data, pengguna mengacu tabel untuk data yang diinginkan. Basis data terdiri atas banyak *table* sehingga harus dapat relasi antar tabel. Karena menyimpan data, tabel membutuhkan penyimpanan fisik pada host komputer untuk basis data (Janner Simarmata ; 2010 : 38).

### **II.6. Normalisasi**

#### **II.6.1 Bentuk Tidak Normal**

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti *format* tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan sesuai keadaannya.

#### **II.6.2 Bentuk Normal Tahap Pertama**

Bentuk normal tahap pertama (1NF) terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*multivalued attribute*) atau lebih dari satu *atribut* dengan nilai *domain* yang sama. Masing-masing sel bernilai tunggal. Permintaan yang menyatakan tidak ada yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci (Janner Simarmata ; 2010 : 79).

#### **II.6.3. Bentuk Normal Tahap Kedua**

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah

tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama (Janner Simarmata ; 2010 :81).

#### **II.6.4. Bentuk Normal Tahap Ketiga**

Bentuk Normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga atau 3NF jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama (Janner Simarmata ; 2010 : 82).

#### **II.6.5. Bentuk Normal Tahap Keempat**

Sebuah tabel relasional berada pada bentuk normal keempat atau 4NF jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat atau 4NF didasarkan pada konsep ketergantungan multivalued atau MVD. Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak garis bernilai sama, tetapi kolom lain bernilai berbeda (Janner Simarmata ; 2010 : 85).

#### **II.6.6. Bentuk Normal Tahap Kelima**

Sebuah tabel berada pada bentuk normal kelima 5NF jika dia tidak dapat mempunyai dekomposisi *lossles* menjadi sebuah tabel lebih kecil. Empat bentuk

normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan konsep ketergantungan gabungan atau *join dependence*. Ketergantungan gabungan berarti bahwa sebuah tabel, setelah didekomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain, 5NF menunjukkan ketika sebuah tabel tidak dapat didekomposisi lagi (Janner Simarmata ; 2010 : 86).

## **II.7. UML (*Unified Modelling Language* )**

*UML* singkatan dari *Unifed Modelling Language* yang berarti bahasa pemodelan standar. (Chonoles,2003: bab 1)

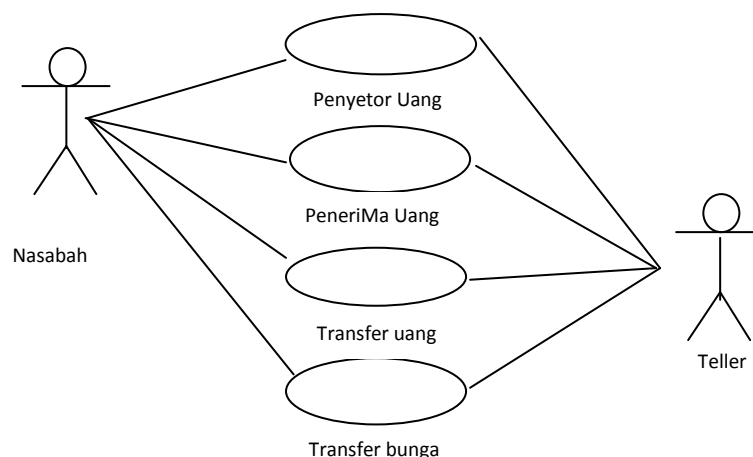
Mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantik. Ketika kita berbuat model menggunakan konsep *UML* ada aturan- aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya ? Bagaimana sistem mengatasi eror yang terjadi ? Bagaimana keamanan terhadap sistem yang kita buat ? Dan sebagainya dapat dijawab dengan *UML*.

*Unified Modelling Language* (*UML*) adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan desain berorientasi objek. *UML* menyediakan standar dan diagram yang bisa digunakan untuk memodelkan suatu sistem. *UML* memungkinkan developer

melakukan pemodelan secara visual, yaitu penekanan pada penggambaran. (Prabowo P.Widodo, dkk, menggunakan UML 2011)Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari objek, mempermudah penggambaran interaksi antar elemen dalam sistem dan mempertahankan konsistensi antara desain dan implementasi dalam pemrograman. Tiga artifak penting UML untuk proses analisis dan desain adalah Use Case Diagram, Sequence Diagram dan Class (Prabowo Pudjo Widodo ; 2011 : 7).

### **II.7.1. Use Case Diagram**

Menurut (Pilone,2005: bab7.1) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten, 2004: 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal . *Use case* digambarkan dalam bentuk elips/ oval. Use case sangat menentukan karakteristik sistem yang kita buat, oleh karena itu (Chonoles, 2003: bab 8) menawarkan cara untuk menghasilkan *use case* yang baik.



**Gambar II.6 Diagram Use Case**

**Sumber : Prabowo Pudjo Widodo ; 2011 : 17**

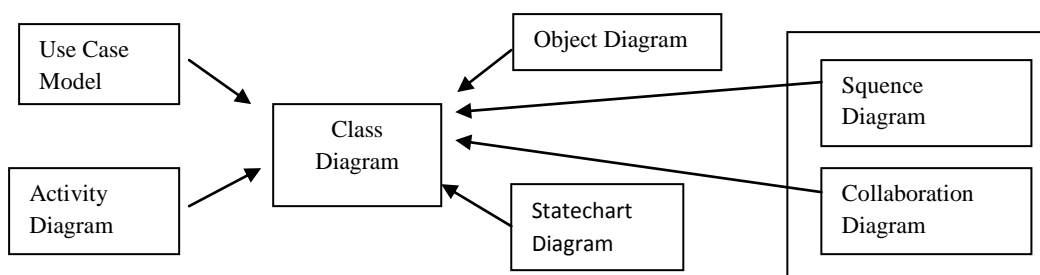
### **II.7.2. Sequence Diagram**

*Sequence* diagram menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan yang dianalisa dalam *use case*, hal ini meliputi penjabaran interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Pembuatan *sequence* diagram merupakan aktivitas yang paling kritical dari proses disain karena artifak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan juga berisi aliran kontrol program.

*Sequence* diagram biasanya terdiri dari elemen objek, interaction dan *message*. *Interaction* menghubungkan 2 objek dengan pesannya. Diagram ini menjelaskan aspek dinamis sistem yang sedang dibangun.

### II.7.3. Class Diagram

*Class* diagram adalah inti dari proses pemodelan objek. Baik forward engineering maupun reverse engineering memanfaatkan diagram ini. forward engineering adalah proses perubahan model menjadi kode program sedangkan reverse engineering sebaliknya menggunakan kode program menjadi model.



**Gambar II.7 Class Diagram**

**Sumber : Prabowo Pudjo Widodo ; 2011 :38**

### II.7.4. Activity Diagram

Diagram aktifitas lebih mefokuskan diri pada eksekusi dan alur sistem, daripada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software, melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk, kumpulan aksi-aksi. Ketika digunakan dalam pemodelan software, diagram aktivitas mempresentasikan, pemanggilan sesuatu fungsi tertentu, misalnya *call*, Sedangkan dipergunakan dalam pemodelan bisnis, diagram ini digambarkan atau dipicu, dengan kejadian-kejadian yang di luar seperti pemesanan, atau kejadian internal, misalnya proses pendataan (prabowo pujo widodo ; 2011 : 143).