

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem merupakan sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkuumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Verdi Yasin, 2012:260)

Pengertian sistem dilihat dari elemen-elemennya. Sistem adalah kumpulan elemen yang saling berkaitan dan bekerja sama dalam melakukan kegiatan untuk mencapai suatu tujuan. Contoh : sistem pernapasan manusia yang terdiri dari elemen-elemen hidung, tenggorokan, paru-paru, pembuluh darah, dan darah, elemen-elemen tersebut saling bekerja sama. Pengertian sistem dilihat dari masukan dan keluarannya. Sistem adalah suatu rangkaian yang berfungsi menerima *input* (masukan), mengolah *input*, dan menghasilkan *output* (keluaran). Sistem yang baik akan mampu bertahan dalam lingkungannya. Contoh : sistem produksi dalam perusahaan yang terdiri dari masukan berupa bahan baku dan kemudian diolah oleh mesin dan akan menghasilkan barang jadi. Jika ada salah satu yang terganggu maka sistem produksi akan terganggu (V. Wiratna Sujarweni, 2015:1)

Tujuan umum dari suatu sistem adalah menghubungkan berbagai bagian dari sistem tersebut. Meskipun tiap bagian berfungsi secara independen dari yang

lainnya, semua bagian tersebut melakukan tujuan yang sama. Jika komponen tertentu tidak memberikan kontribusinya pada tujuan bersama, maka komponen tersebut bukanlah bagian dari sistem tersebut. Contohnya, sepasang sepatu *ice-skating* dan jaring permainan voli adalah komponen. Akan tetapi, keduanya tidak memiliki tujuan bersama dan karenanya tidak membentuk sistem (James A. Hall, 2011:7)

## **II.2. Akuntansi**

Akuntansi adalah “suatu proses mencatat, mengklasifikasi, meringkas, mengolah dan menyajikan data, transaksi serta kejadian yang berhubungan dengan keuangan sehingga dapat digunakan oleh orang yang menggunakannya dengan mudah dimengerti untuk pengambilan suatu keputusan serta tujuan lainnya. Ada juga sebahagian orang mendefinisikan akuntansi itu adalah sebagai suatu seni mengumpulkan, mengelompokkan, mencatat, dan melaporkan transaksi-transaksi yang bernilai ekonomi secara sistematis”.

Akuntansi berasal dari kata asing *accounting* yang artinya bila diterjemahkan kedalam bahasa Indonesia adalah menghitung atau mempertanggungjawabkan. Akuntansi digunakan di hampir seluruh kegiatan bisnis di seluruh dunia untuk mengambil keputusan sehingga disebut sebagai bahasa bisnis (Elizar Sinambela, dkk, 2010:1)

Menurut V. Wiratna Sujarweni (2015:3), Akuntansi adalah proses transaksi yang dibuktikan dengan faktur, lalu dari transaksi dibuat jurnal, buku besar, neraca lajur, kemudian akan menghasilkan informasi dalam bentuk laporan

keuangan yang digunakan pihak-pihak tertentu. Pihak-pihak yang menggunakan laporan keuangan diantaranya adalah :

1. Pihak manajemen perusahaan di mana laporan keuangan dapat digunakan untuk mengambil keputusan.
2. Pemilik perusahaan, fungsi laporan keuangan digunakan untuk memberi tahu keadaan perusahaan dari sisi keuangan.
3. Investor & pemegang saham, disini investor biasanya melihat laporan keuangan untuk mengambil keputusan penanaman saham.
4. Kreditor atau pemberi utang biasanya melihat kesehatan perusahaan dari laporan keuangan untuk memutuskan perusahaan layak diberikan kredit atau tidak.
5. Pemerintah, berkepentingan untuk memungut pajak berdasarkan laporan keuangan yang ada.
6. Karyawan, memerlukan informasi akuntansi untuk mengetahui profitabilitas dan akuntabilitas perusahaan tempat mereka bekerja.

### **II.3. Sistem Akuntansi**

Sistem akuntansi adalah organisasi formulir, catatan, dan laporan yang dikoordinasi sedemikian rupa untuk menyediakan informasi keuangan yang dibutuhkan oleh manajemen guna memudahkan pengelolaan perusahaan.

Dari definisi sistem akuntansi tersebut, unsur-unsur sistem akuntansi pokok adalah formulir, catatan yang terdiri dari jurnal, buku besar dan buku pembantu, serta laporan (Mulyadi, 2013:3).

#### **II.4. Estimasi**

Menurut Kamus Besar Bahasa Indonesia (KBBI) Estimasi adalah perkiraan, penilaian atau pendapat. Estimasi adalah suatu metode dimana kita dapat memperkirakan nilai dari suatu populasi dengan menggunakan nilai dari sampel. Estimator adalah nilai pendugaan/suatu data statistik, sebagai sampel yang digunakan untuk mengisi suatu parameter. (kajianpustaka 2014/02/)

#### **II.5. Pembiayaan**

Pembiayaan adalah Penyerahan nilai ekonomi sekarang atas kepercayaan dengan harapan mendapatkan kembali suatu ekonomi yang sama di kemudian hari. (kajianpustaka 2014/02/)

#### **II.6. Bahan Baku**

Bahan baku adalah bahan yang digunakan dalam membuat produk di mana bahan tersebut secara menyeluruh tampak pada produk jadinya (atau merupakan bagian terbesar dari bentuk barang). Sedangkan biaya bahan baku adalah seluruh biaya untuk memperoleh sampai dengan bahan siap untuk digunakan yang meliputi harga bahan, ongkos angkut, penyimpanan dan lain-lain(Bahan baku 2014/02).

#### **II.7. Visual Basic 2010**

Visual Basic 2010 merupakan salah satu bagian dari produk pemrogramanterbaru yang dikeluarkan oleh Microsoft, yaitu Microsoft Visual Basic 2010. Sebagai produk lingkungan pengembangan terintegrasi atau IDE

andalan yang dikeluarkan oleh Microsoft, Visual Basic 2010 menambahkan perbaikan-perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi Visual Studio pendahulunya, yaitu Microsoft Visual Studio 2008.

Visual studio merupakan produk pemrograman andalan dari Microsoft Corporation, yang di dalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C#, dan Visual F#. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework terbaru, yaitu .Net Framework 3.5. Adapun database standar yang disertakan adalah Microsoft SQL Server 2008 express (Wahana Komputer, 2010:2).

## **II.8. Basis Data (*Database*)**

Basis data (*database*) adalah kumpulan tabel dari suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Sebagai contoh, basis data akademis mengandung tabel-tabel yang berhubungan dengan data siswa, data kelas, data mata pelajaran, dan nilai yang diperoleh siswa (Verdi Yasin, 2012:279).

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Pada buku ini menggunakan basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS).

Kebutuhan basis data dalam sistem informasi meliputi :

- Memasukkan, menyimpan, dan mengambil data
- Membuat laporan berdasarkan data yang telah disimpan

Tujuan dari dibuatnya tabel-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom (Rosa A.S. dan M. Shalahuddin, 2015:43).

## **II.9. Normalisasi**

Menurut Verdi Yasin (2012 : 278) Normalisasi adalah suatu teknik untuk menghasilkan kumpulan relasi dengan properti yang diinginkan untuk mengetahui kebutuhan data dalam organisasi. Tujuan dari normalisasi adalah untuk menghilangkan kerangkapan data, untuk mengurangi kompleksitas dan untuk mempermudah modifikasi data. *Unnormalized Form* (UNF) adalah tabel yang berisi satu atau lebih *repeating group*. Dari bentuk inilah dilakukan proses normalisasi. Bentuk proses normalisasi dapat dijelaskan sebagai berikut :

a. Bentuk tidak normal (UNF)

Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi dan data dikumpulkan apa adanya.

b. Bentuk normal pertama (1NF)

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom (tidak perlu ada indeks dalam memberi nama kolom).

c. Bentuk normal kedua (2NF)

Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer dan tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama.

d. Bentuk normal ketiga (3NF)

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh.

## II.10. SQL Server

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelolah data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

SQL mulai berkembang pada tahun 1970an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini :

No.	Tahun	Nama
1	1986	SQL-86
2	1989	SQL-89
3	1992	SQL-92
4	1999	SQL:1999
5	2003	SQL:2003
6	2006	SQL:2006
7	2008	SQL:2008
8	2011	SQL:2011

(Sumber : Rosa A. S dan M. Shalahuddin, 2015:46)

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama (Rosa A. S dan M. Shalahuddin, 2015:46).

## II.11. UML (*Unified Modeling Language*)

Menurut Verdi Yasin (2012 : 267) *Unified Modeling Language* (UML) adalah notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer. Di dalam pemodelan objek guna menyajikan sistem yang berorientasi pada objek pada orang lain, akan sangat sulit dilakukan jika pemodelan tersebut dilakukan dalam bentuk kode bahasa pemrograman. Kesulitan yang muncul adalah timbulnya ketidakjelasan dan salah interpretasi di dalam pembacaan kode pemrograman untuk pemodelan objek tersebut.

UML disebut sebagai bahasa pemodelan bukan metode. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka Anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain. UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

### - Tujuan UML

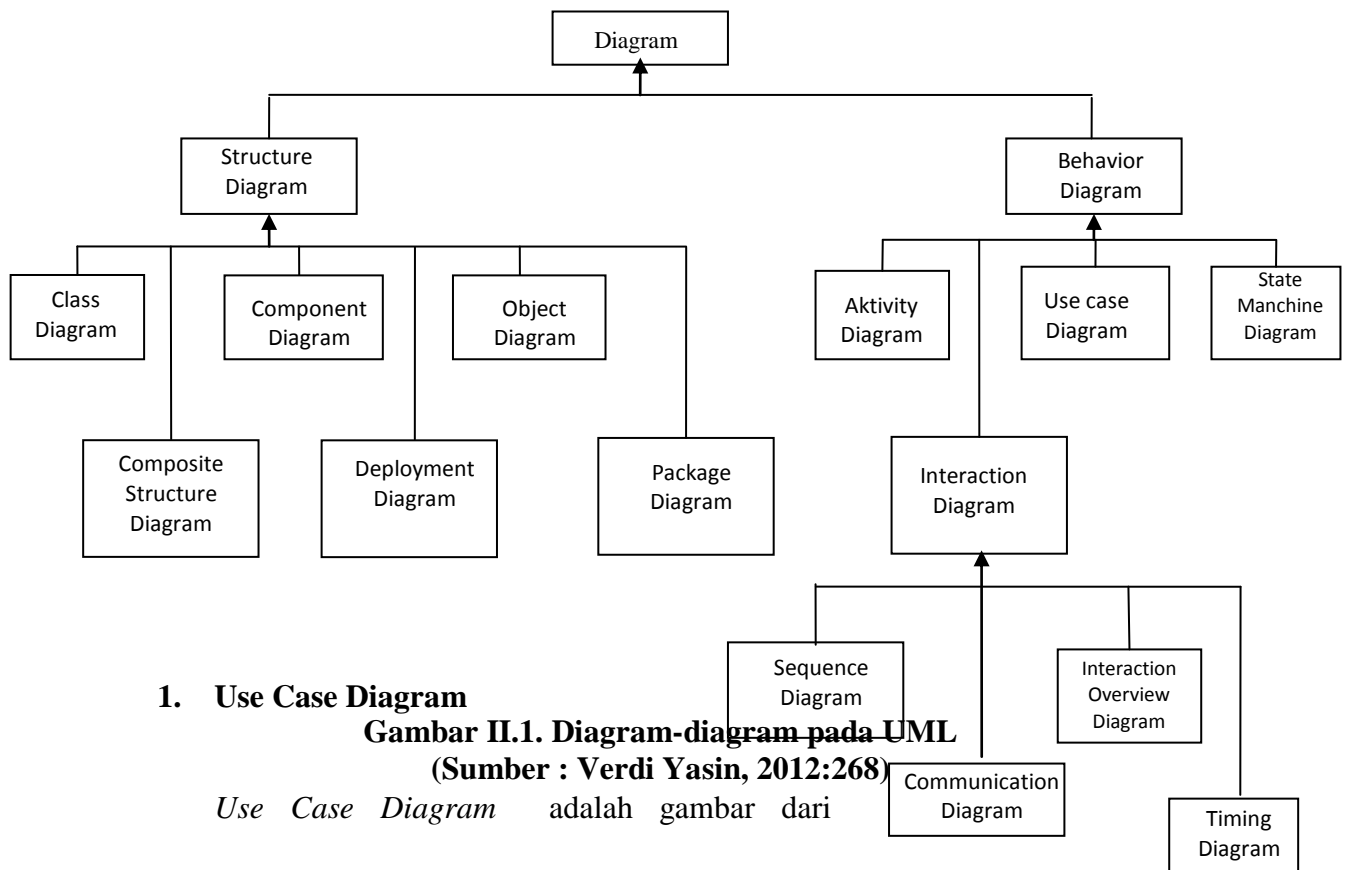
Tujuan UML Diantaranya adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan yang dapat saling menukar model dengan mudah dan dimengerti secara umum.

2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa
3. Menyatuhkan praktek-praktek terbaik yang terdapat dalam pemodelan.

#### - Tipe-tipe diagram UML

UML merupakan sintak umum untuk membuat model logika dari suatu sistem dan digunakan untuk menggambarkan sistem agar dapat dipahami selama fase analisis dan desain. UML biasanya disajikan dalam bentuk diagram/gambar yang meliputi *class* beserta atribut dan operasinya, serta hubungan antar *class* yang meliputi *inheritance*, *association* dan komposisi. UML terdiri dari banyak diagram, secara garis besar diagram yang terdapat pada UML dapat diperlihatkan pada gambar dibawah ini :



#### 1. Use Case Diagram

##### Gambar II.1. Diagram-diagram pada UML

(Sumber : Verdi Yasin, 2012:268)

*Use Case Diagram* adalah gambar dari

beberapa atau seluruh aktor dan *use case* dengan tujuan mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempersentasikan sebuah interaksi antar *actor* dengan sistem. *Use case* menggambarkan kata kerja seperti login ke sistem, *maintenance user* dan sebagainya. Oleh karena itu, *use case diagram* dapat membantu menganalisa kebutuhan suatu sistem. Dalam *use case diagram* terdapat istilah seperti aktor, *use case* dan *use case relationship*.

a. Aktor

Aktor mewakili siapa pun atau apa saja yang harus berinteraksi dengan sistem. Aktor bisa didefinisikan sebagai berikut :

1. Aktor hanya memberikan informasi kepada sistem.
2. Aktor hanya menerima informasi dari sistem.
3. Aktor memberikan dan menerima informasi ke dan dari sistem.

b. *Use case*

*Use case* model adalah dialog antara aktor dengan sistem yang akan menggambarkan fungsi yang diberikan oleh sistem.

c. *Use case relationship*

*Use case relationship* adalah suatu hubungan, baik itu antara aktor dan *use case* atau antara *use case* dan *use case*. Hubungan antara aktor dan *use case* disebut dengan *communicate association*.

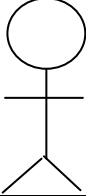
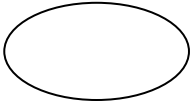

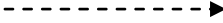
d. *Association/Directed association*

Asosiasi yaitu hubungan statis antar elemen. Umumnya menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain. Tanda panah menunjukkan arah *query* antar elemen.

e. *Generalization/Pewarisan*

Pewarisan merupakan hubungan hierarkis antar elemen. Elemen dapat diturunkan dari elemen lain dan mewarisi semua atribut dan metode elemen asalnya dan menambahkan fungsionalitas baru, sehingga disebut anak dari elemen yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

**Tabel II.1. Simbol Use Case Diagram**

Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan system yang sedang dikembangkan.
	<i>Use Case</i> : peringkat tertinggi dari fungsionalitas yang dimiliki system.
	<i>Association</i> : adalah relasi antara <i>actor</i> dan <i>use case</i>
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.

## 2. Activity Diagram

*Activity* diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

a. *Activity*

Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.

b. *Transition*

Notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari *activity* ke *activity*.

c. *Decision*

Notasi yang menandakan kontrol cabang aliran berdasarkan decision point.



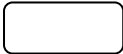



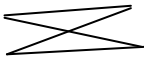
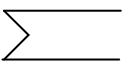

d. *Synchronization bars*

Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan (pararel).

*Activity* diagram menggambarkan berbagai alir aktivitas dalam sebuah sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi.

*Activity* diagram tidak menggambarkan sifat internal dari sebuah sistem dan interaksi antara beberapa sub sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

**Tabel II.2. Simbol Activity Diagram**

Simbol	Keterangan
	Titik awal
	Titik akhir
	Activity
	Pilihan untuk mengambil keputusan
	Fork : untuk menunjukkan kegiatan yang dilakukan secara paralel
	Rake : menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda penerimaan
	Aliran akhir (Flow Final)

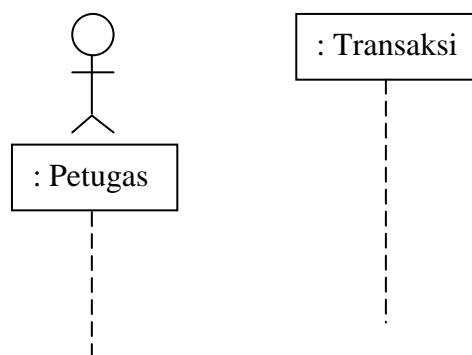
(Sumber : Verdi Yasin, 2012 : 271)

### 3. *Sequence Diagram*

*Sequence* diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence* diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan

untuk menghasilkan sesuatu sesuai dengan *use case* diagram. *Sequence* diagram berhubungan erat dengan *use case* diagram dimana 1 *use case* akan menjadi 1 *sequence* diagram. Dalam *sequence* diagram terdapat 2 simbol yaitu :

- a. *Actor*, untuk menggambarkan penggunaan sistem
- b. *LifeLine*, untuk menggambarkan kelas dan objek



**Gambar II.2. Actor dan LifeLine**

(Sumber : Verdi Yasin, 2012 : 273)

Dalam *sequence* diagram terdapat 3 relasi yaitu ;

1. *Create*

Relasi ini digunakan untuk melakukan inisialisasi suatu objek

2. *Synchronous*

Relasi ini digunakan untuk memanggil operasi atau method yang dimiliki oleh suatu objek. *Synchronous* mengharuskan kita menyelesaikan 1 proses baru kemudian memanggil proses berikutnya

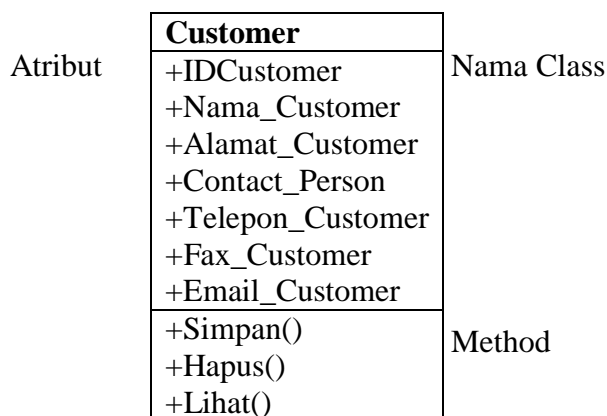
### 3. *Asynchronous*

Relasi ini digunakan untuk memanggil operasi atau method yang dimiliki oleh suatu objek. *Asynchronous* memberikan kita fasilitas untuk menjalankan proses lain ketika proses sebelumnya belum selesai.

### 4. *Class Diagram*

*Class diagram* menggambarkan struktur dan dekripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Objek adalah nilai tertentu dari setiap *attribute* kelas *entity*.

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan berorientasi objek. *Class* menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasikan keadaan tersebut (metode/fungsi).



**Gambar II.3. Class Dalam UML**  
(Sumber : Verdi Yasin, 2012 : 274)