

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pemasaran**

Pemasaran merupakan faktor penting dalam perkembangan perusahaan. Pemasaran yang baik dilakukan lewat produk (*product*), harga (*price*), tempat/distribusi (*place*), dan promosi (*promotion*). Kurangnya pengetahuan terhadap selera konsumen yang berubah-ubah atau dinamis, serta tidak efektifnya strategi pemasaran yang diterapkan akan berdampak kurang baik pada pencapaian target pasar. Pemasaran adalah Salah satu unsur dalam strategi pemasaran terpadu adalah strategi bauran pemasaran yang merupakan strategi yang dijalankan perusahaan, yang berkaitan dengan penentuan bagaimana perusahaan menyajikan penawaran produk pada segmen pasar tertentu, yang merupakan sasaran pasarnya. Bauran pemasaran rangkaian sarana pemasaran taktis terpadu yang dapat dikendalikan (produk, harga, tempat, dan promosi) untuk mengetahui respon pasar sasaran yang diinginkan oleh perusahaan Machfoedz pada tahun 2005. Pengertian dari *Marketing Mix* adalah himpunan variabel yang dikuasai dan dapat digunakan oleh manajer pemasaran untuk menarik konsumen guna mempengaruhi penjualan atau pendapatan perusahaan. (*Algrina Agnes Ulus ; 3 : 1136*)

Manajemen pemasaran ialah kegiatan menganalisa, merencana, mengimplementasikan dan mengawasi segala kegiatan guna mencapai tingkat pemasaran sesuai dengan tujuan yang telah ditetapkan oleh perusahaan. (*Valdy Ronald Wowor ; 2013 :1230*)

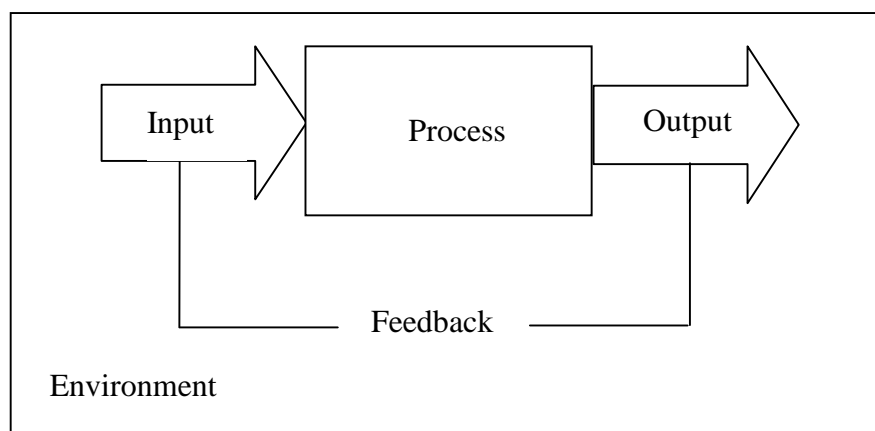
Dari pengertian diatas maka dapat disimpulkan pengertian pemasaran adalah suatu kegiatan yang strategis dalam merencanakan dan mengimplementasikan bentuk permintaan pasar, yang dimana suatu perusahaan harus mengikuti permintaan pasar sehingga dapat menjanjikan produk barang yang ditawarkan memiliki kualitas permintaan pasar untuk mencapai tujuan yang ditetapkan.

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

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan, Namun, secara sederhana sistem informasi dapat didefinisikan sebagai suatu sistem yang bertujuan untuk menyimpan, memproses dan mengkomunikasikan informasi. (*Purnomo Budi Santoso ; 2013 : 61*)

Sebuah sistem informasi merupakan kumpulan dari perangkat keras, lunak serta manusia yang akan mengolah dan menggunakan perangkat keras dan lunak tersebut. Informasi merupakan hal yang sangat penting dengan adanya informasi tersebut dapat diketahui kemajuan dan kegagalan proses pelaksanaan. Sistem yang kurang informasi menunjukkan bahwa sistem tersebut rapuh. Data merupakan informasi yang diolah supaya berguna bagi yang menerimanya. (*Indra Warman ; 2012 : 44*)

Berdasarkan pengertian yang tertera diatas maka, Kesimpulan Sistem Informasi adalah suatu perangkat atau elemen yang telah disusun sedemikian rupa sehingga dapat menyimpan, memproses serta mengkomunikasikan data informasi kepada para pengguna serta saling berintegrasi untuk mencapai suatu tujuan dari sistem organisasi.



**Gambar II.1. Bagan Sistem**

*Sumber : Purnomo Budi Santoso.*

- a. Masukkan (*Input*) merupakan bagian awal sistem yang meliputi semua hal yang dijadikan masukan atau masuk ke dalam sistem tersebut.
- b. Proses (*Process*) merupakan suatu pekerjaan yang menstransformasikan masukan menjadi keluaran.
- c. Keluaran (*Output*) adalah produk jadi atau hasil dari pengolahan masukan oleh proses.
- d. Lingkungan (*Environment*) adalah tempat dimana sistem tersebut berada atau diletakkan. Yang termasuk lingkungan adalah semua elemen yang berada di luar sistem yang mempengaruhi dan dipengaruhi oleh sistem tersebut.
- e. Umpan Balik (*Feedback*) adalah suatu sensor yang berguna untuk mencegah

timbulnya keluaran yang tidak sesuai dengan yang diinginkan atau memberikan informasi bila hal ini terjadi. (Purnomo Budi Santoso ; 2013 : 61)

### **II.2.1. Sistem Informasi Manajemen**

Sistem Informasi Manajemen Pengertian Sistem Informasi Manajemen adalah : “Suatu susunan komponen-komponen yang terinterelasi bekerja secara bersama-sama untuk mengumpulkan, memproses, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, *control*, analisis. (Purnomo Budi Santoso ; 2013 : 61)

Sistem informasi manajemen merupakan perangkat lunak yang didisain untuk membantu dalam hal pemeliharaan dan utility data dalam jumlah yang besar. (Indra Warman ; 2012 : 45)

Dalam uraian tentang sistem informasi manajemen diatas maka dapat disimpulkan bahwa sistem informasi manajemen adalah suatu perangkat lunak atau komponen-komponen yang telah tersusun rapi dalam jumlah besar yang bekerja bersama-sama dalam mengolah, menyimpan serta memproses maupun membantu pemeliharaan data informasi yang telah tersusun.

Dan *visualisasi* dalam sebuah organisasi.” Satu komponen kunci yang memungkinkan SIM memenuhi kebutuhan- kebutuhan pada bagian-bagian yang beragam tersebut adalah penggunaan *entity-wide database*, yaitu penyimpanan semua data yang ada dalam organisasi. Untuk kasus UKM Dinoyo ini, jenis dan tingkatan laporan yang dihasilkan oleh sistem informasi adalah bervariasi, antara lain meliputi :

- a. Laporan bersifat rangkuman (*summary*)
- b. Laporan bersifat perkecualian/keanehan (*exceptions*)
- c. Laporan bersifat rutin, misal mingguan, semesteran, tahunan, dst.
- d. Laporan bersifat dadakan, misalnya mencari alamat *supplier*. (*Purnomo Budi Santoso ; 2013 : 61*)

### II.3. *MySql*

*MySQL Server 2000* adalah suatu Perangkat lunak *Relational Database Management system* (RDBMS) yang handal. Didesain untuk mendukung proses transaksi yang besar (seperti *order entri yang online, inventori, akuntansi* atau manufaktur). *MySQL Server* akan secara otomatis menginstal enam *database* utama, yaitu *master, model, tempdb, pubs, Northwind, dan Msdb*. (*Anis nurhanafi ; 2013 : 5*)

MySQL adalah sebuah system managemen database relasi (relational database management system) yang bersifat “terbuka” (*open source*). MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. Perintah SQL sering jgua disebut *Query*. Karena menggunakan bahasa standar yang sama, maka tidak akan menjadi kendala besar bila suatu saat nanti anda berhubungan dengan database selain MySQL. Tentu saja, tetap ada sedikit perbedaan sintaks bahasa antara paket program SQL tersebut. (*Dahlan Abdullah ; 2013 : 155*)

Untuk kesimpulan dari *MySql* adalah suatu perangkat lunak atau software yang menggunakan bahasa standart SQL yang dapat mengolah atau memproses data dalam jumlah besar seperti akuntansi atau manufaktur maupun bentuk lainnya.

#### II.4. *Database*

*Database* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan diperangkat lunak untuk memanipulasinya. (Indra Warman ; 2012 : 45)

*Database*, merupakan kumpulan *file-file* yang berhubungan secara logis dan digunakan secara rutin pada operasi-operasi sistem informasi manajemen. (Purnomo Budi Santoso ; 2013 : 60)

Berdasarkan pengertian diatas maka dapat disimpulkan bahwa *Database* adalah sekumpulan elemen-elemen data yang telah tersusun didalam computer yang disusun ke dalam *record-record* dan *file-file* dengan berbagai cara untuk memenuhi kebutuhan informasi bagi pengguna (*user*)

Sistem *database* manajemen dibentuk untuk mengurangi masalah-masalah dalam organisasi. Misalnya data/informasi tidak tersedia atau saling tumpang tindih. Prinsip manajemen *database* adalah :

- a. Ketersediaan Data mudah diakses oleh suatu program dan pemakai ( *user* ) dimanapun dan kapanpun diperlukan.
- b. Pemakaian bersama Struktur data disusun sedemikian hingga dapat digunakan oleh beberapa pemakai bersama-sama untuk mengurangi redudansi data.
- c. Pengembangan *Databases* dapat dikembangkan sesuai dengan perkembangan kebutuhan pemakai. *Databases* dapat dimodifikasi untuk pengembangan selanjutnya dan dapat beradaptasi dengan lingkungan.

- d. Kesatuan *Databases* dibentuk dalam satu kesatuan untuk memudahkan pengotrolannya ( pemeliharaan dan pengawasan ) mudah dilakukan. (*Anis nurhanafi ; 2013 : 5*)

#### **II.4.2. Jenis-Jenis Database**

##### **a. Database hirarki**

Yaitu suatu data yang tersusun dengan bentuk hirarki pohon. Susunan yang seperti ini terdiri dari beberapa unsur komponen yang saling mempengaruhi dan tidak dapat dipisahkan, jenis database ini merupakan hubungan satu komponen dengan banyak komponen.

##### **b. Database relasi**

Adalah suatu data yang disusun dalam bentuk tabel yang terdiri dari dua definisi dan tersusun secara terstruktur. Bentuk susunan dua dimensi ini terdiri dari beberapa kolom dan *record* yang tersusun berbentuk baris dari kiri kekanan. Data- data yang susunannya berbentuk baris adalah susunan yang menurun kebawah. Dimana pada setiap baris berisikan data- data yang saling berkaitan satu sama lainnya. Artinya setiap pemasukan data yang tersimpan pada field merupakan kesatuan dalam bentuk satu baris. (*Indra Warman ; 2012 : 46*)

#### **II.4.3. Pengertian Database Management System (DBMS)**

*Database Management System* (DBMS) adalah satu koleksi data yang saling berelasi dan satu *set* program untuk mengakses data tersebut. Jadi DBMS terdiri dari *database* dan *set* program pengelola untuk menambah, menghapus data, mengambil data dan membaca data.

*Database* adalah suatu koleksi data komputer yang terintegrasi, diorganisasikan dan disimpan dalam suatu cara yang memudahkan pengambilan kembali (*Raymond McLeod, Jr.* Jilid 1 Edisi Bahasa Indonesia, 1995). Sedangkan *set* program adalah paket program yang diolah dan dibuat untuk memudahkan dalam pemasukkan atau pembuatan data. Menurut *Date*, basis data dapat dianggap sebagai tempat untuk sekumpulan berkas data terkomputerisasi (Kadir, 1999:9). (*Anis Nurhanafi ; 2013 : 3*)

## **II.5. Website**

*Website* (Situs *Web*) merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan *file-file* lain yang terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi *website*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah kehalaman lain yang terdapat dalam *website* tersebut. (*Yoni Widhiarso ; 2013 : 3*)

*Website* merupakan kumpulan halaman-halaman yang berisi informasi yang disimpan diinternet yang bisa diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa mengakses internet itu sendiri seperti komputer. (*Harun Al-Rosyid ; 2013 : 2*)

Berdasarkan sumber diatas tentang pengertian *website* maka dapat disimpulkan bahwa *Web* sebenarnya penyederhanaan dari sebuah istilah dalam dunia komputer yaitu *WORLD WIDE WEB* yang merupakan bagian dari teknologi *Internet*, jadi *website* adalah sekumpulan halaman-halaman situs *internet* yang dapat diakses oleh setiap pengguna.

Secara umum *website* mempunyai fungsi sebagai berikut :

1. Fungsi Komunikasi

Beberapa fasilitas yang memberikan fungsi komunikasi, seperti : *chatting*, *web base email* dan lain-lain.

2. Fungsi Informasi

Fungsi informasi *website* seperti : *News*, *Profile*, *Library*, referensi dan lain-lain.

3. Fungsi *Intertainment*

Website mempunyai fungsi hiburan. Misalnya *web-web* yang menyediakan *game on-line*, *music on-line* dan lain-lain. (Yoni Widhiarso ; 2013 : 3)

## II.6. UML (*Unified Modelling Language*)

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan

bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 *Booch, Rumbaugh dan Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease *draft* pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. *Booch, Rumbaugh dan Jacobson* menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma

menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. (Yuni Sugiarti ; 2013 : 33)

UML adalah sebuah bahasa yang sudah menjadi standar dalam industri untuk merancang, menspesifikasi dan mendokumentasi sistem perangkat lunak. UML memberikan standar penulisan tersendiri pada sebuah sistem *blue print*, yang mencakup konsep proses bisnis, penulisan kelas-kelas pada bahasa program yang spesifik, skema database dan komponen-komponen yang dibutuhkan dalam sistem piranti lunak. (Arzan Muharom ; 2013 : 2)




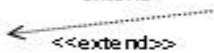


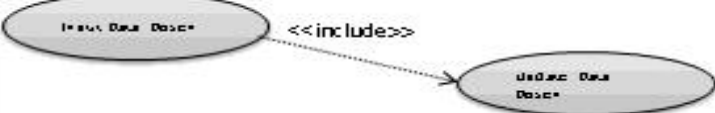
Dalam pengertian yang tertera diatas maka kesimpulan dari UML adalah sebuah bahasa permodelan yang memiliki standart dalam industri yang berguna untuk merancang sebuah rancangan yang diimplementasikan sebelumnya sehingga memiliki tujuan yang ingin dicapai.

Dalam pembuatan skripsi ini penulis menggunakan diagram *Use Case* yang terdapat di dalam UML. Adapun maksud dari *Use Case* Diagram diterangkan dibawah ini.

### **1. *Use Case Diagram***

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat

membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-extend *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. (Yuni Sugiarti ; 2013 : 41)

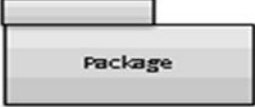







Simbol	Deskripsi
<p>Use Case</p> 	<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case</p>
<p>Aktor</p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem 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 aktor</p>
<p>Asosiasi / association</p> 	<p>komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor</p>
<p>Extend</p> 	<p>relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjukkan pada use case yang dituju contoh :</p> 
<p>Include</p> 	<p>relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, contoh :</p> 

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

Sumber : (Yuni Sugiarti ; 2013 ; 42)

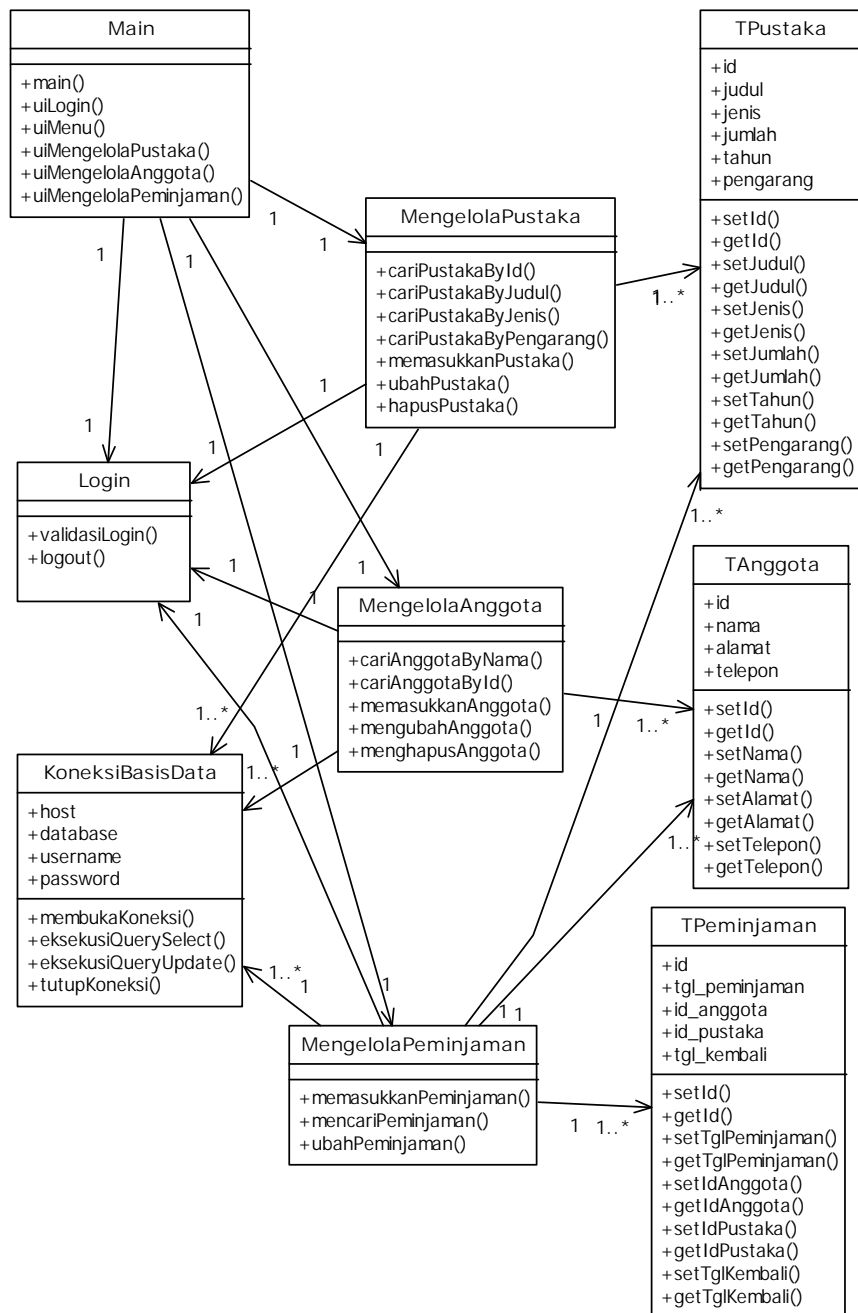
## 2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol pada diagram kelas :

Simbol	Deskripsi
<p>Package</p> 	Package merupakan sebuah bungkusian dari satu atau lebih kelas
<p>Operasi</p> 	Kelas pada struktur sistem
<p>Antarmuka / interface</p> 	sama dengan konsep interface dalam pemrograman berorientasi objek
<p>Asosiasi</p> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
<p>Asosiasi berarah/directed asosiasi</p> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
<p>Generalisasi</p> 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan / defedency</p> 	relasi antar kelas dengan makna kebergantungan antar kelas
<p>Agregasi</p> 	relasi antar kelas dengan makna semua-bagian (whole-part)

**Gambar II.3. Class Diagram**

*Sumber : (Yuni Sugiarti ; 2013 : 59)*



**Gambar II.4. Contoh Class Diagram**

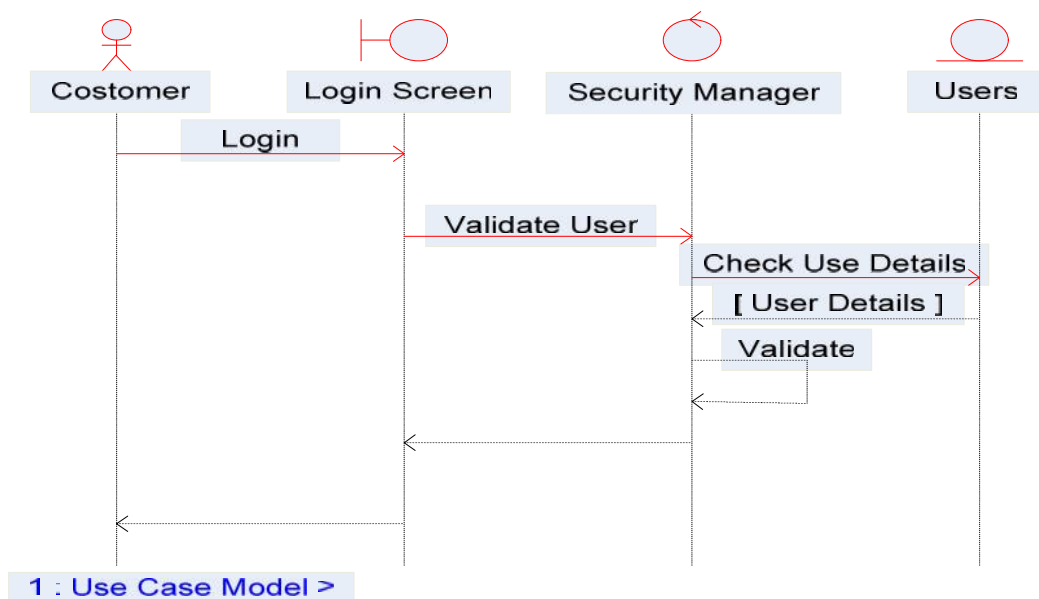
Sumber : (Yuni Sugiarti ; 2013 : 63)

### 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 menggambarkan 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 interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.



**Gambar II.5. Contoh Sequence Diagram**

Sumber : (Yuni Sugiarti ; 2013 : 63)

#### 4. Activity Diagram

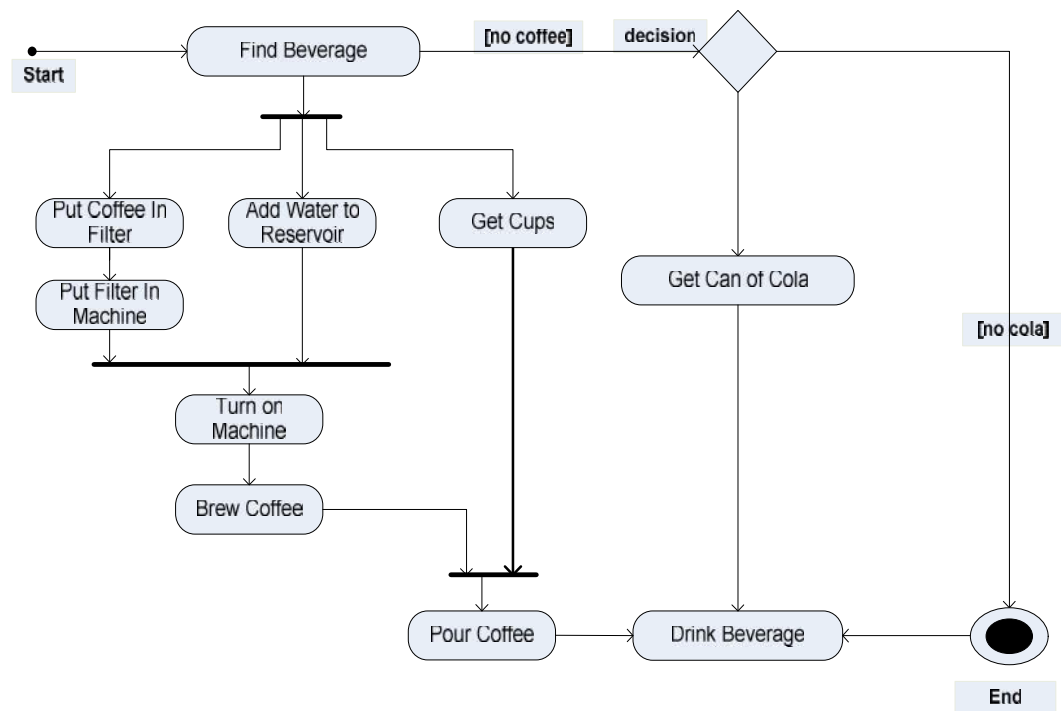
*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

*Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



**Gambar II.6. Activity Diagram**  
*Sumber : (Yuni Sugiarti ; 2013 : 76)*