

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Defenisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Berikut akan diberikan beberapa defenisi sistem secara umum :

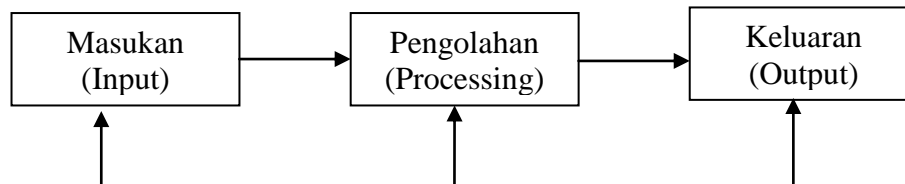
1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.

Contoh :

- a. Sistem Tatasurya
 - b. Sistem Pencernaan
 - c. Sistem Transportasi Umum
 - d. Sistem Otomatif
 - e. Sistem Komputer
 - f. Sistem Informasi
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan (Hanif Al Fatta; 2007 : 1).

Menurut Scoot (1996), sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Ciri pokok sitem menurut Gaspert ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan,

terdiri dari unsur-unsur, ditandai dengan saling berhubungan, dan mempunyai satu fungsi atau tujuan yang sama.



Gambar II.1 : Model Sistem

(Sumber : Hanif Al Fatta ; 2007 : 2)

II.1.1. Pengertian Informasi

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah, informasi dapat dikelompokkan menjadi 3 bagian, yaitu :

1. Informasi strategis, informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.
2. Informasi taktis, informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi teknis, informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan *stock*, retur penjualan, dan laporan kas harian.

Informasi adalah data yang telah diklasifikasikan atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tidak berguna menjadi berguna bagi yang menerimanya. Nilai informasi

berhubungan dengan keputusan. Bila tak ada pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan (Tata Sutabri ; 2012 : 21).

II.1.2. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem suatu 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 laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38).

II.2. Sistem Informasi Geografis

Pada dasarnya, istilah sistem informasi geografis merupakan gabungan dari tiga unsur pokok : sistem, informasi, dan geografis. Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dan memahami SIG. Dengan melihat unsur-unsur pokoknya, maka sudah jelas bahwa SIG juga merupakan salah satu tipe sistem informasi atau merupakan suatu sistem yang menekankan pada unsur informasi geografis. SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek-objek penting yang terdapat dipermukaan bumi.

Sistem Informasi Geografis (SIG) adalah sistem komputer yang digunakan untuk memanipulasi data geografis. Sistem ini diimplementasikan dengan

menggunakan perangkat keras dan perangkat lunak komputer yang berfungsi untuk : akuisi dan verifikasi data, kompilasi data, penyimpanan data, perubahan atau *updating* data, manajemen dan pertukaran data, manipulasi data, pemanggilan dan presentasi data, dan analisa data (Edy Prahasta : 2009: 109).

II.3. ArcView

ArcView adalah software dikeluarkan oleh *ESRI (Environmental Systems Research Institute)*. Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada *ArcView* untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *ArcView* banyak diterapkan dalam berbagai pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persil, pengendalian dampak lingkungan, bahkan untuk keperluan militer (Eko Budiyanto : 2010: 176).

II.4. Macromedia Dreamweaver 8

Macromedia Dreamweaver 8 adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman *web*. Dengan menggunakan *Dreamweaver 8*, ketika membuat halaman *web*, tidak perlu lagi mengetik kode-kode *HTML* atau kode-kode lainnya secara manual. Cukup melakukan klik beberapa kali, maka halaman web yang di inginkan sudah jadi. *Macromedia Dreamweaver 8* dapat membuat tabel hanya dengan melakukan dua kali klik saja. *Macromedia Dreamweaver 8* akan menciptakan kode-kode *HTML* yang sesuai

untuk membuat tabel yang diinginkan. Selain *HTML*, *Macromedia Dreamweaver* 8 juga mendukung *CSS*, *JavaScript*, *PHP*, *ASP* dan bahasa pemrograman lainnya untuk membuat *web* (Arief Ramadhan, S.Kom ; 2007 : 2).

II.5. PHP (*Hypertext Preprocessor*)

PHP (Hypertext Preprocessor) adalah kode/skrip yang akan dieksekusi pada *server side*. Skrip *PHP* akan membuat suatu aplikasi dapat diintegrasikan ke dalam *HTML*, sehingga suatu halaman *web* tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan skrip dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser* (Deni Sutaji ; 2012: 2).

II.5.1. Aturan PHP

Adapun aturan penulisan skrip *PHP* ada dua cara, yaitu :

1. *Embedded Script*

Dengan cara meletakkan *tag PHP* diantar *tag-tag HTML*.

Contohnya :

```
<html>
<body>
<?php echo "Belajar"; ?>
</body>
</html>
```

2. *Non Embedded Script*

Dengan cara ini, semua *skript HTML* diletakkan didalam *skript PHP*.

Contohnya :

```
<?php
echo "<html>";
echo "<body>";
echo "Belajar PHP";
echo "<body>";
echo "<html>";
?>
```

Sedangkan penulisan *PHP* ada empat macam, yaitu :

1. *Standart Style*

Format penulisannya : `<?php ... ?>`

2. *Short Style*

Format penulisannya : `<? ... ?>`

3. *Javascript Style*

Format penulisannya : `<skript language ' PHP ' > ... </skript>`

4. *ASP Style*

Format penulisannya : `<% ... %>`

II.6. Pengertian Database

Database merupakan komponen utama dalam membangun sebuah sistem yang menyangkut pendokumentasian data ke dalam sebuah *database*. Bentuk basis data adalah sebuah aturan yang mengatasi masalah tersebut. Saat ini basis data memiliki peranan yang sangat penting dalam mengolah data yang ada di

dalamnya. Validasi juga mencakup di dalamnya karena aturan-aturannya dalam sebuah data yang terdokumentasikan juga memiliki sebuah aturan yang dikenal dengan basis data. *Database* adalah sebuah cara mendokumentasikan berbagai macam data yang kemudian dimanajemen dengan sebuah sistem untuk disimpan dalam sebuah media penyimpanan. Dengan kemudian data tersebut dapat di akses dengan mudah dan cepat. Media penyimpanan tersebut dapat di ibaratkan sebagai sebuah *storage* penyimpanan, misalnya *Hardisk* (Bunafit Nugroho ; 2005: 71).

II.6.1. Pengertian Perancangan Basis Data

Sebelum merancang basis data, ada baiknya jika mengenal istilah –istilah yang sering digunakan didalamnya, yaitu:

1. **Database**, adalah sebuah bentuk media yang digunakan untuk menyimpan sebuah data. *Database* dapat diilustrasikan sebagai sebuah rumah atau gudang yang akan dijadikan tempat penyimpanan berbagai macam barang. Dalam *database* barang tersebut adalah data.
2. **Table**, digunakan untuk menyimpan sebuah kelompok data.
3. **Field**, yaitu bentuk terkecil dalam *database* untuk menyimpan data.
4. **Primary Key**, yaitu bentuk kolom atau *field* utama yang mewakili kolom lainnya dalam sebuah tabel untuk menghindari data yang kembar.
5. **Foreign Key**, yaitu kunci yang berasal dari tabel lain.
6. **Kunci Calon atau Kunci Alternatif**, adalah kunci yang dapat dinominasikan sebagai kunci primer (Bunafit Nugroho ; 2005 : 74).

II.6.2. Perancangan Basis Data

Sebelum melakukan perancangan *database* ada beberapa kriteria yang harus dipenuhi untuk memperoleh validasi data. Beberapa hal yang harus dilakukan untuk memenuhi kriteria tersebut adalah sebagai berikut :

1. Studi Kelayakan : langkah ini menjadi langkah pembuka yang harus dilakukan demi mendapatkan informasi yang valid sehingga dapat diterapkan ke dalam sebuah basis data.
2. Rencana Pendahuluan : merupakan langkah menentukan lingkup sistem yang dikerjakan. Hal ini berguna untuk melakukan penentuan waktu yang dibutuhkan dalam proses pembuatan sistem.
3. Menganalisa Sistem : setelah perencanaan selesai, langkah selanjutnya adalah melakukan penganalisaan data.
4. Merancang Sistem : melakukan perancangan sistem.
5. Implementasi : setelah perancangan sistem dilakukan dan muncul hasilnya, maka dapat melanjutkan proses analisis dengan mengimplementasikannya (Bunafit Nugroho ; 2005 : 80).

II.7 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structured Query Language)*. *MySQL* merupakan sebuah *database server* yang *free*, artinya bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL adalah sebuah program *database*, sedangkan *SQL* adalah bahasa perintah (*query*) dalam program *MySQL* (Bunafit Nugroho ; 2005: 1).

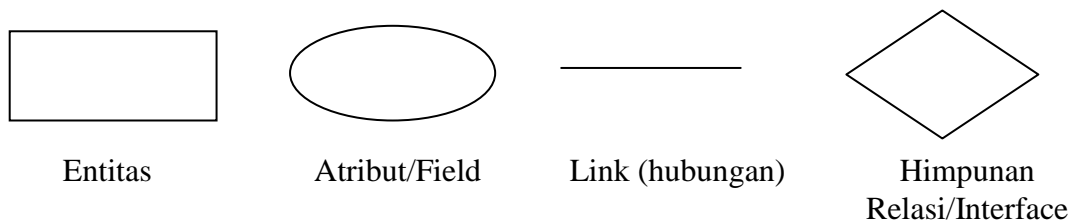
II.7.1. Kelebihan MySQL

MySQL adalah sebuah database server, dapat juga berperan sebagai *client /server*. *Database* ini memiliki kelebihan dibanding database lain, di antaranya adalah :

1. *MySQL* sebuah *Database Management System (DBMS)*.
2. *MySQL* sebagai *Relation Management System (RDBMS)*.
3. *MySQL* adalah sebuah *software database* yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja.
4. *MySQL* mampu menerima *query* yang bertumpuk dalam satu permintaan.
5. *MySQL* merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *gigabyte*.
6. *MySQL* dapat menciptakan lebih dari 16 kunci per tabel, dan dalam satu kunci memungkinkan berisi belasan kolom (Bunafit Nugroho ; 2005: 3).

II.8. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD merupakan salah satu alat (*tool*) berbentuk grafis yang populer untuk *desain database*. *Tool* ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, *tool* ini mencapai 2NF (Ir. Yuniar Supardi ; 2010 : 448).



Gambar II.2 : Bentuk Simbol ERD
 (Sumber : Ir. Yuniar Supardi ; 2010 : 448)

II.9. Teknik Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan atau kriteria standar untuk menghasilkan struktur tabel yang normal.

Dalam perspektif normalisasi sebuah database dikatakan baik jika setiap tabel yang membentuk basis data sudah berada dalam keadaan normal. Suatu tabel dikatakan normal, jika :

- Jika ada kelompok / penguraian tabel maka dekomposisinya dijamin aman.
- Terpeliharanya ketergantungan *functional* pada saat perubahan data.
- Tidak melanggar *Boyce Code Normal Form (BCNF)*.

II.9.1. Bentuk-bentuk Normalisasi

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 apa adanya sesuai keadaannya.

2. Bentuk normal tahap pertama (1st Normal Form)

Sebuah tabel disebut 1NF jika : tidak ada baris yang diduplikat dalam tabel tersebut, masing-masing *cell* bernilai tunggal.

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk Normal Kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*).

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka: X haruslah *superkey* pada tabel tersebut, atau A merupakan bagian dari *primary key* pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal Keempat berkaitan dengan sifat ketergantungan banyak nilai pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (*PJNF*).

II.10. *Unified Modeling Language (UML)*

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier. Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para

anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati ; 2011 ; 6).

II.10.1. Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

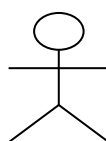
1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*Package Diagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*) bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya.

II.10.2. Notasi Dasar UML

1. Aktor

Sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.



Gambar II.3 : Aktor

(Sumber : Probowo Pudjo Widodo ; 2011:17)

2. Use Case

Use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. *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 *ellips/oval*.



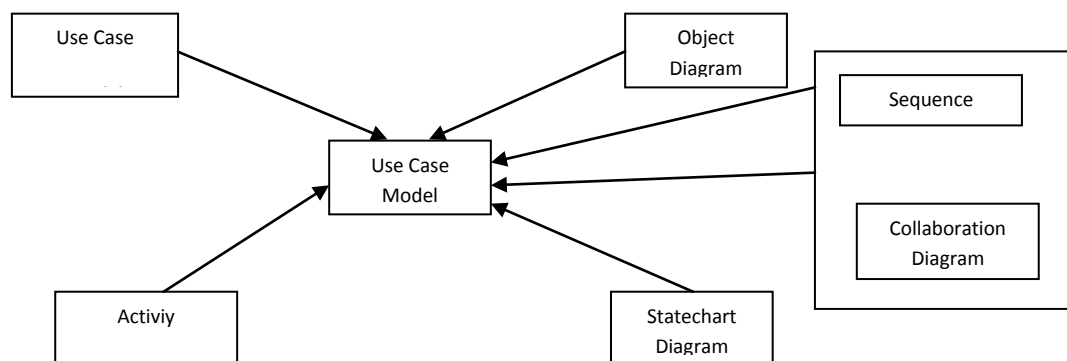
Gambar II.4 : Simbol Use Case

(Sumber : Probowo Pudjo Widodo ; 2011:22)

3. Diagram Kelas (*Class Diagram*)

Diagram kelas 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 merubah kode program menjadi model (Probowo Pudji Widodo; 2011 : 37).



Gambar II.5 : Hubungan Diagram Kelas Dengan Diagram UML lainnya

(Sumber : Probowo Pudjo Widodo ; 2011 : 38)

4. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

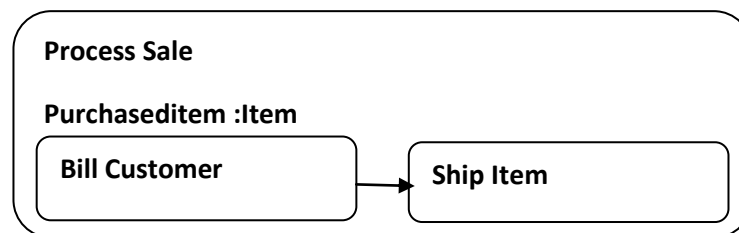
Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku.

Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.6 : Aktivitas serderhana tanpa rincian
(Sumber : Probowo Pudjo Widodo ; 2011:145)

Detail aktivitas dapat dimasukkan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.

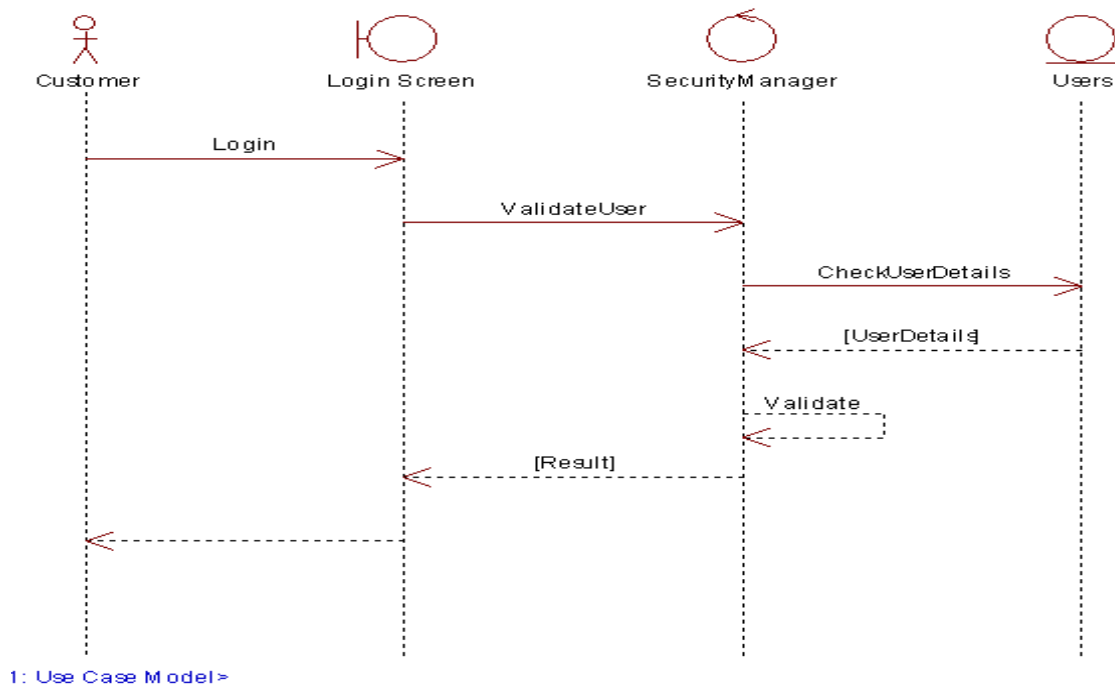


Gambar II.7: Aktivitas dengan detail rincian
(Sumber : Probowo Pudjo Widodo ; 2011:145)

5. Sequence Diagram

Ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.8. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya.



Gambar II.8 : Diagram Urutan
(Sumber : Probowo Pudjo Widodo ; 2011:175)