

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem dan Analisa Sistem

II.1.1. Pengertian Sistem

Sistem adalah sekumpulan komponen yang saling berhubungan, bekerja sama untuk mencapai suatu tujuan dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur.

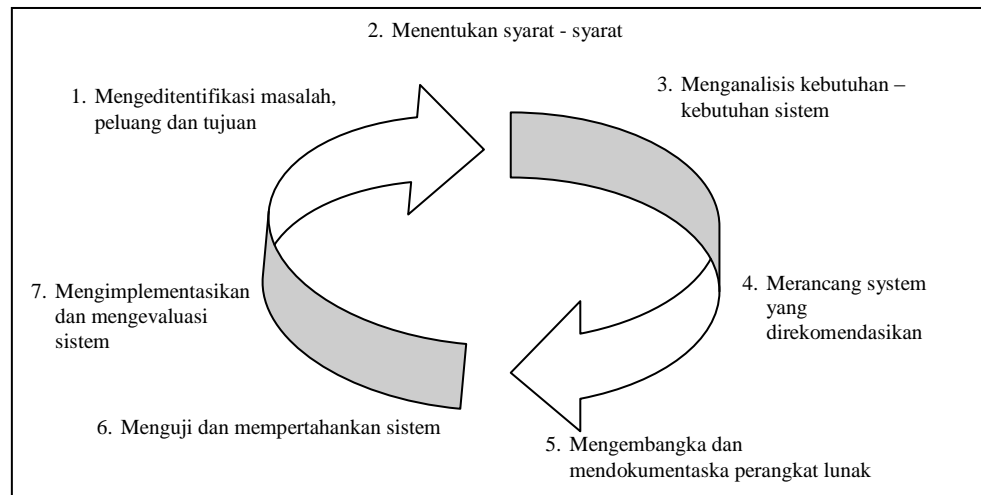
Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional (dengan tugas/fungsi khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu. Sebagai contoh, sistem kendaraan terdiri dari: komponen *starter*, komponen pengapian, komponen penggerak, komponen pengerem, komponen kelistrikan-spedometer, lampau dan lain-lain. Komponen-komponen tersebut diatas memiliki tujuan yang sama yaitu untuk membuat kendaraan tersebut bisa dikendarai dengan nyaman dan aman. Contoh lain yaitu sistem perguruan tinggi, yang terdiri dari dosen, mahasiswa, kurikulum, dan lain-lain. Sistem ini bertujuan untuk menghasilkan mahasiswa-mahasiswa yang memiliki kemampuan di bidang ilmunya. (Kusrini; 2011:1)

II.1.2. Siklus Hidup Pengembangan Sistem

Siklus hidup pengembangan sistem ini adalah pengembangan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem

tersebut telah di kembangkan dengan sangat baik melalui pengembangan siklus kegiatan penganalisisan dan pemakaian secara spesifik (Kusrini; 2008:2)

Adapun siklus hidup pengembangan sistem data pada gambar II.1. :



Gambar II.1. Siklus Hidup Pengembangan Sistem
(Sumber : Kusrini ; 2011 : 9)

II.1.3. Karakteristik Sistem

Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu, antara lain di defenisikan (Asbon Hendra 2012 : 158) yakni :

1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan.

2. Batasan Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang mempengaruhi operasi dari suatu sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara suatu subsistem dengan subsistem lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem, masukan dapat berupa masukan perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*Output*)

Merupakan hasil energi yang diolah oleh sistem meliputi *output* yang berguna dan *output* yang tidak berguna dikenal sebagai sisa pembuangan.

7. Pengolahan Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

8. Tujuan Sistem (*Goal*)

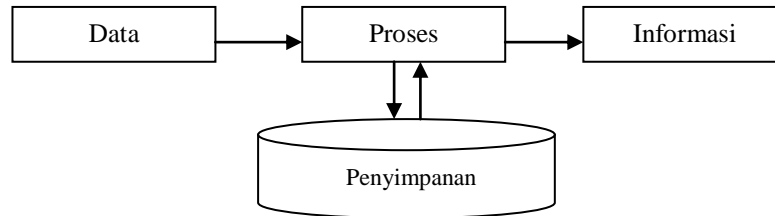
Setiap sistem pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan.

II.1.4. Pengertian Informasi

Informasi adalah data yang dapat dianalogikan dengan data – data, yang belum di kelola dan harus di olah untuk menjadi informasi yang akurat. (Kusrini ;2011:12). Agar informasi yang penulis sajikan lebih bermanfaat maka terlebih dahulu dibuat aliran informasi yang lebih jelas dan lengkap. Berkaitannya dengan penyedia informasi bagi manajemen dalam mengambil suatu keputusan, yang diperoleh harus berkualitas, maka kualitas dari informasi tergantung pada :

1. Akurat : akurat berarti bahwa informasi harus bebas dari kesalahan – kesalahan dan tidak biasa (menyesatkan) dan jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerimaan informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi atau merusak informasi tersebut.
2. Relevansi : relevansi berarti bahwa informasi benar – benar berguna bagi suatu tindakan dan keputusan oleh seseorang.
3. Tepat waktu : tepat waktu berarti bahwa informasi yang datang pada penerimaan tidak boleh terlambat. Informasi yang sudah dipakai tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan dalam pengambilan keputusan. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu di dapat.

Untuk lebih jelasnya informasi merupakan hasil atau *output* dari proses informasi data. Hal ini dapat dilihat seperti gambar II.2. dibawah ini :



Gambar II.2. Proses Data Menjadi Informasi
(Sumber : Kusriani ; 2011 :12)

II.2. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi. SIA juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi. Saat ini, digital dan informasi *online* semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai faktor yang terkait ketika mengatur sistem informasi akuntansi.

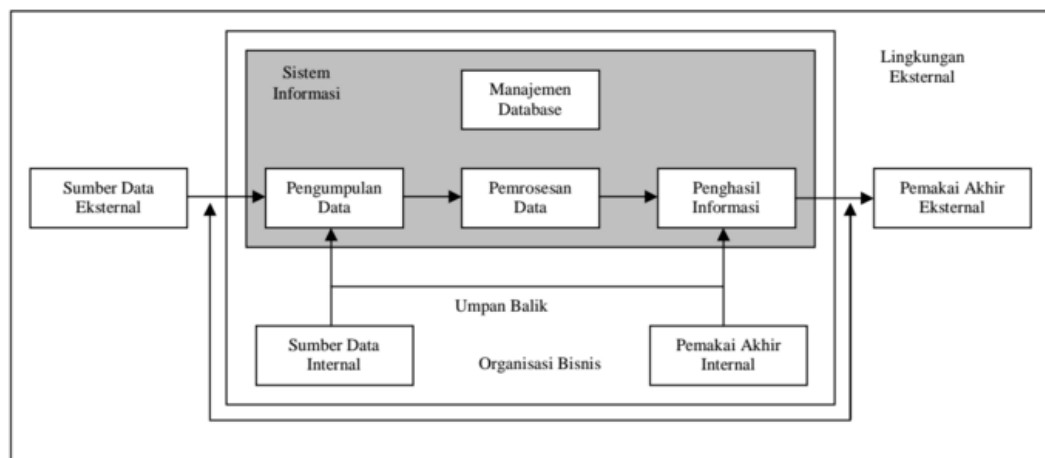
SIA pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran - pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.

3. Siklus produksi. Berkaitan dengan perubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus Mujilan;2012:3).

II.2.1. Arsitektur Sistem Informasi Akuntansi

Arsitektur SIA merupakan kerangka model umum yang menggambarkan semua sistem yang digunakan dalam pembelajaran SIA dari mulai sumber data, proses pengumpulan, pemrosesan dan penyajian hasil informasi baik di lingkungan internal maupun eksternal perusahaan, yang dapat kami paparkan sebagai berikut :



Gambar : II.3. Arsitektur Komputer
(Sumber : *ebook ariskoerniawan 2012 : 5*)

II.2.2. Model-model Sistem Informasi Akuntansi

Model sistem informasi pada khususnya SIA telah disajikan dengan beberapa model atau pendekatan yang berbeda yang berkembang untuk mengurangi kelemahan atau kekurangan dari model sebelumnya. Model sistem

informasi (Evolusi Sistem Informasi Akuntansi) sesuai dengan pendekatan SIA antar lain :

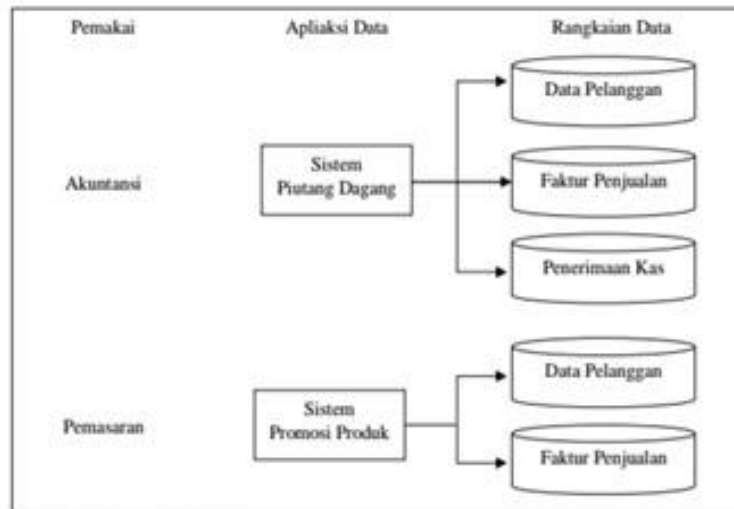
1. Model Proses Manual

Model ini merupakan suatu model sistem informasi yang diproses secara tradisional atau manual yang merupakan model sistem yang paling tua. Akan tetapi dengan proses manual banyak hal yang bisa diperoleh sebelum menggunakan sistem komputer antara lain :

- a. Dapat membantu seseorang untuk mudah memahami logika proses
- b. Dapat membantu membangun hubungan antar sistem dan memicu
- c. Sistem manual memfasilitasi dalam pemahaman kegiatan control

2. Model Flat File

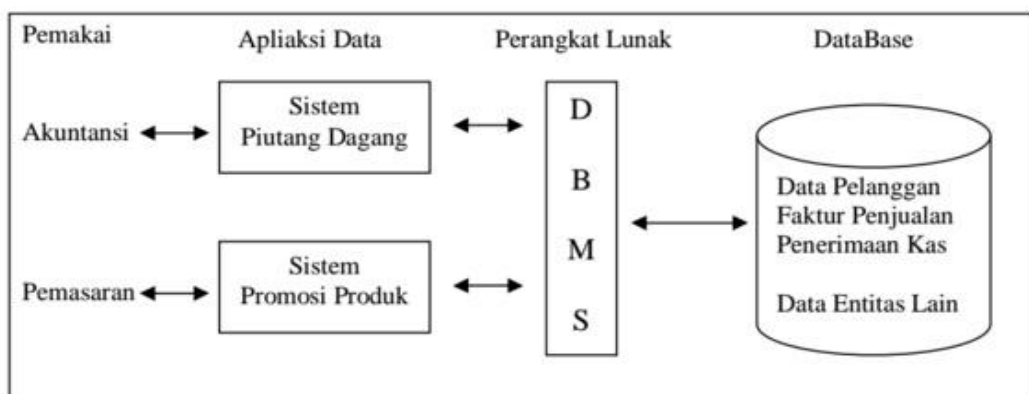
Pendekatan flat file sering disebut sebagai sistem warisan yang merupakan kerangka utama dari sistem yang banyak dilakukan dalam perusahaan. Sistem flat file menjelaskan suatu sistem di mana file-file data individual tidak berkaitan dengan file-file lainnya (aplikasi tunggal), dimana pemakai akhir memiliki data sendiri dan tidak menggunakannya secara bersama-sama dengan pemakai lainnya. Sehingga ketika ada pemakai lain memerlukan data yang sama untuk tujuan yang berbeda, pemakai harus membentuk rangkaian tersendiri (rangkaiannya data terpisah) yang disajikan secara spesifik sesuai dengan kebutuhan.



Gambar II.4. Model Flat File
 (Sumber : ebook ariskoerniawan 2012:8)

3. Model Database

Model Database merupakan pendekatan dengan memusatkan organisasi data ke sebuah database (tempat penyimpanan) yang dapat dipakai secara bersama oleh para pemakai. Dimana organisasi data di tempatkan pada lokasi yang sentral dan pengontrolan database biasanya menggunakan sistem manajemen database (*Database Management System / DBSM*).



Gambar II.5. Model Database
 (Sumber : ebook ariskoerniawan 2012 : 8)

4. Model REA

Model sistem informasi yang bertumpu pada kerangka akuntansi berupa *Resources, Events, Agents*.

- a. Resources (Sumber Daya) adalah aktiva/ harta yang dimiliki oleh suatu perusahaan
- b. Events (Peristiwa) adalah kejadian/ peristiwa ekonomi yang mempengaruhi sumber daya perusahaan.

II.3. Penjualan

Sistem Informasi Akuntansi Penjualan adalah suatu sistem informasi yang mengorganisasikan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisa, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan mengenai penjualan. Tujuan sistem penjualan adalah : Mencatat order penjualan dengan cepat dan akurat, memverifikasi konsumen yang layak menerima kredit, mengirim produk dan memberikan jasa tepat waktu, sesuai yang dijanjikan kepada konsumen, membuat tagihan atas produk dan jasa secara tepat waktu dan akurat, mencatat dan mengelompokkan penerimaan kas secara cepat dan akurat, memposting penjualan dan penerimaan kas ke rekening piutang, untuk menjaga keamanan produk dan untuk menjaga kas perusahaan. (Rochmawati Daud, Valeria Mimosa Windana ; 2014 : 3).

II.4. Metode *Free On Board Shipping Point*

Free On Board Shipping Point adalah transaksi penjualan barang dagang di mana penyerahan hak kepemilikan atas barang dagang tersebut dilakukan di gudang penjual. Konsekuensinya, seluruh beban pengiriman barang dagang sejak dari gudang penjual hingga gudang pembeli menjadi tanggungan pembeli. Itu berarti jika transaksi penjualan dilakukan dengan menggunakan *Free On Board Shipping Point* maka biaya pengiriman tidak di catat dan di jurnal oleh pihak penjual. Sebaliknya, pihak pembeli harus mencatat dan menjurnal biaya pengiriman tersebut dalam buku jurnalnya. Akibatnya harga beli barang dagang tersebut akan bertambah sebesar biaya pengirimannya. (Rudianto ; 2012 : 117)

Free On Board Shipping Point adalah dimana dalam hal ini perusahaan harus membayar biaya transportasi dari tempat pengiriman. Biaya-biaya semacam ini merupakan total biaya dalam pembelian barang (persediaan) dan harus ditambahkan ke harga pokok persediaan dengan mendebit persediaan barang dagang. Dalam biaya angkut pembelian barang dagangan akan muncul masalah biaya pengangkutan barang gudang penjual ke gudang pembeli. Siapa yang menanggung biaya angkut ini, tergantung pada perjanjian yang dibuat antara penjual dan pembeli. Biaya angkut pembelian bisa ditanggung oleh penjual, dan jika seperti itu perusahaan pembeli terbebani oleh biaya angkutan. Namun seringkali biaya angkut ini ditanggung oleh pembeli, biaya ini akan ditambahkan pada harga pokok pembelian barang dagangan, dengan mendebit biaya angkut dalam membeli barang dagangan sebesar yang ditentukan oleh perusahaan dan dibayar tunai. (David H. M. Hasibuan ; 2010 : 7)

Dengan syarat ini barang yang diperjual belikan menjadi milik pembeli pada saat barang masih dalam perjalanan. Kemudian segala resiko dalam perjalanan sampai ke gudang pembeli menjadi tanggung jawab pembeli. Jika dengan syarat *FOB Shipping Point*, maka :

1. Untuk penjual nilai persediaan akan berkurang pada saat barang dikirim.
2. Untuk pembeli nilai persediaan akan bertambah ketika barang telah dikirim.

II.5. Alat Bantu Pengembangan Sistem

Alat bantu pengembangan sistem yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

II.5.1. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Haviluddin;2011:1).

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin;2011:1).


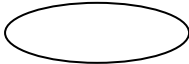


II.5.2. Tipe Diagram UML

Tipe-tipe di dalam diagram UML adalah :

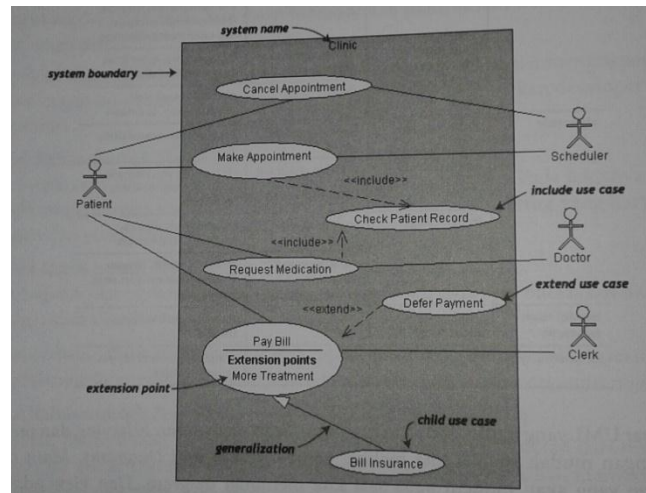
1. Use Case Diagram

Use Case Diagram menggambarkan fungsionalisme yang diharapkan dari sebuah sistem. Sebuah *Use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Tabel II.1. Simbol Use Case Diagram

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

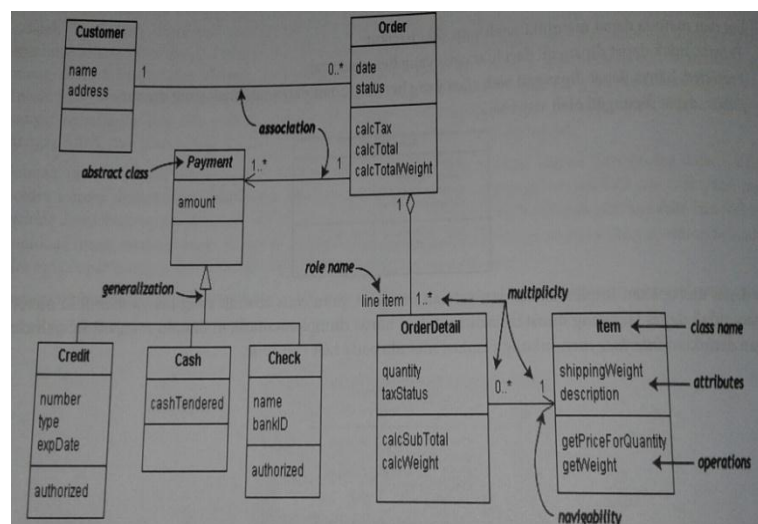
(Sumber :Verdi Yasin 2012 : 270)



Gambar II.6. Use Case Model
(Sumber : Verdi Yasin 2012 : 198)

2. Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.



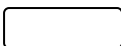


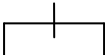
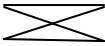
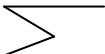



Gambar II.7. Class Diagram
(Sumber : Verdi Yasin 2012 : 200)

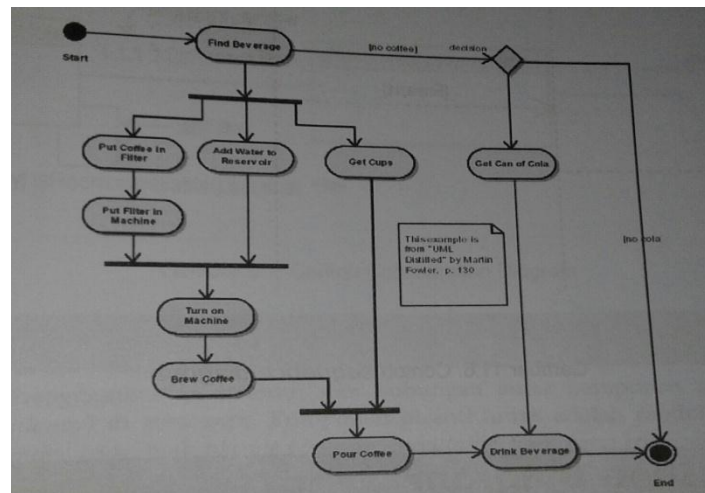
3. Activity Diagram

Activity diagram merupakan *state* diagram khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Tabel II.2. Simbol Activity Diagram

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork : Untuk menunjukkan kegiatan yang dilakukan secara parallel.
	Rake : Menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Penerimaan
	Aliran Akhir (Flow Final)

(Sumber :Verdi Yasin 2012 : 271)

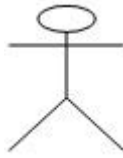
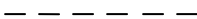


Gambar II.8. Activity Diagram
(Sumber : Verdi Yasin 2012 : 201)

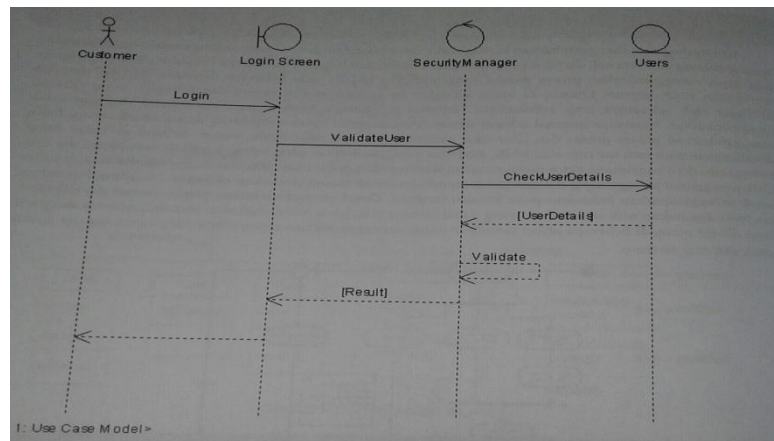
4. Sequence Diagram

Sequence Diagram adalah menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal objek-objek yang terkait).

Tabel II.3. Simbol Sequence Diagram

Simbol	Keterangan
	<i>Actor</i> : Untuk menggambarkan pengguna sistem
	<i>Lifeline</i> : Untuk menggambarkan kelas dan objek.

(Sumber :Verdi Yasin 2012 : 272)



Gambar II.9. Sequence Diagram
(Sumber : Verdi Yasin 2012 : 202)

II.6. Sistem Basis Data

Basis data adalah penyimpanan kumpulan informasi secara sistematis dalam sebuah komputer sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sedangkan sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses pekerjaan. Sehingga bisa dikatakan bahwa sistem basis data adalah sistem yang terdiri atas kumpulan file-file yang saling berhubungan dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi data tersebut. (Politeknik Telkom ; Sistem Manajemen Basis Data : 2009).

Komponen-komponen pada sebuah sistem basis data antara lain:

1. Perangkat keras
2. Sistem operasi
3. Basis data
4. DBMS (Database Management System)
5. Pemakai
6. Aplikasi lain

Pada sebuah institusi, data merupakan salah satu hal yang sangat penting. Setiap bagian/divisi dari institusi memiliki data sendiri-sendiri. Tapi setiap bagian pun membutuhkan sebagian data dari bagian yang lain. Hal ini yang biasa dikenal sebagai “*shared data*”. Setiap divisi memiliki aplikasi sendiri-sendiri dalam melakukan manipulasi dan pengambilan data tersebut. Setiap aplikasi memiliki file-file dalam sistem operasi yang digunakan untuk menyimpan data-data. Seiring dengan berkembangnya institusi, bertambahnya bagian/divisi, bertambah pula data dan aplikasi yang digunakan. Bertambahnya aplikasi, bertambah pula file-file yang dibuat.

Gaya sistem pemrosesan-file tersebut menyebabkan setiap data disimpan dalam bentuk record dalam berbagai macam file, dan diperlukan aplikasi yang berbeda dalam melakukan pengambilan record dari, dan penambahan record ke dalam file. Hal ini berlaku pada masa sebelum adanya Sistem Basis Data (DBMS). (Politeknik Telkom ; Sistem Manajemen Basis Data : 2009).

Menyimpan data dalam bentuk file yang berbeda-beda, memiliki kekurangan-kekurangan :

1. Data *redundancy* dan *inconsistency*

Dikarenakan *programmer* yang berbeda membuat file dan aplikasi masing-masing, menyebabkan beragam format dan aplikasi yang dibuat. Bahkan, aplikasi pun dibuat menggunakan bahasa pemrograman yang berbeda-beda. Lebih jauh lagi, data atau informasi yang sama bisa terdapat dalam beberapa file yang berbeda. Ini yang disebut dengan *redundancy*. *Redundancy* data ini lama kelamaan akan menyebabkan *inconsistency* dari data.

2. Kesulitan dalam pengaksesan data

Dikarenakan setiap aplikasi memiliki file tersendiri untuk penyimpanan dan pengambilan data, maka jika suatu bagian dari institusi membutuhkan data dari bagian lain, akan menemui kesulitan. Hal ini dikarenakan aplikasi yang dimiliki bagian tersebut, tidak dapat membaca file yang terdapat di bagian lain.

3. Isolasi data

Dikarenakan data tersebar dalam berbagai macam file, dan file tersebut dalam beragam format, pembuatan aplikasi baru akan terasa sulit ketika harus membaca format dari masing-masing file tersebut.

4. Masalah integritas

Data yang disimpan harus memenuhi hal yang dinamakan dengan *consistency constraint*. Jika sebuah *constraint* berubah, maka seluruh aplikasi yang digunakan harus mengakomodasinya. Masalah akan

muncul, jika *constraint* melibatkan beberapa data dari file yang berbeda-beda.

5. Masalah keamanan

Tidak semua pengguna dari basis data dapat mengakses semua data. Hal ini akan sulit dilakukan jika menggunakan gaya penyimpanan data dalam file. (Politeknik Telkom ; Sistem Manajemen Basis Data : 2009).

Tujuan utama dari sistem basis data adalah untuk menyediakan fasilitas untuk *view* data secara abstrak bagi penggunanya. Namun bagaimana sistem menyimpan dan mengelola data tersebut, hanya diketahui oleh sistem itu sendiri. Abstraksi data merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data. Berikut ini tiga level abstraksi data :

1. Level fisik

Merupakan level terendah pada abstraksi data yang menunjukkan bagaimana sesungguhnya data disimpan. Pada level ini pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri.

2. Level logik

Merupakan level berikutnya pada abstraksi data, menggambarkan data apa yang disimpan pada basis data dan hubungan apa saja yang ada di antara data tersebut.

3. Level view

Merupakan level tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari basis data. Banyak *user* dalam sistem basis data tidak akan terlibat dengan semua data atau informasi yang ada atau yang disimpan.

Para *user* umumnya hanya membutuhkan sebagian data atau informasi dalam basis data yang kemunculannya di mata *user* diatur oleh aplikasi *end user*. (Politeknik Telkom ; Sistem Manajemen Basis Data : 2009).

II.7. *MySQL*

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama *MySQL AB* yang pada saat itu bernama *TcX DataKonsult AB* sekitar tahun 1994-1995. Awalnya *TcX* membuat *MySQL* dengan tujuan mengembangkan aplikasi web untuk klien. *TcX* merupakan perusahaan pengembang software dan konsultan *database*.

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengolahan datanya. Kepopuleran *MySQL* antara lain karena *MySQL* menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan – perusahaan skala menengah-kecil. *MySQL* juga bersifat *open source* dan *free* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali pada *windows* yang bersifat *shareware*). *MySQL* di distribusikan dengan lisensi *open source GPL (General Public License)* mulai versi 3.23 pada bulan Juni 2000.

MySQL merupakan *database* yang pertama kali di dukung oleh bahasa pemrograman *script* untuk internet. *MySQL* dan *PHP* dianggap sebagai pasangan *software* pengembangan aplikasi web yang ideal. *MySQL* lebih sering digunakan

untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script PHP*. (M. Rudyanto Arief ; 2011 : 151)

Fitur-fitur *MySQL* antara lain :

1. *Relational Database System*

Seperti halnya software database lain yang ada di pasaran, *MySQL* termasuk RDBMS.

2. *Arsitektur Client-Server*

MySQL memiliki arsitektur *client-server* dimana *server* database *MySQL* terinstal di *server*. *Client MySQL* dapat berada di komputer yang sama dengan *server*, dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet. Mengenal perintah *SQL* standar. *SQL (Structured Query Language)* merupakan suatu bahasa standar yang berlaku di hampir semua *software* database. *MySQL* mendukung *SQL* versi *SQL:2003*.

3. *Mendukung Sub Select*

Mulai versi 4.1 *MySQL* telah mendukung *select* dalam *select (sub select)*.

4. *Mendukung Views*

MySQL mendukung *views* sejak versi 5.0

5. *Mendukung Stored Prosedured (SP)*

MySQL mendukung *SP* sejak versi 5.0

6. Mendukung *Triggers*

MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas.

Pengembang *MySQL* berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.

7. Mendukung *replication*

8. Mendukung transaksi

9. Mendukung *foreign key*

10. Tersedia fungsi GIS

11. *Free* (bebas didownload)

12. Stabil dan tangguh

13. Fleksibel dengan berbagai pemrograman

14. Security yang baik

II.8. PHP (*Hypertext Preprocessor*)

PHP adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis. Karena *PHP* merupakan *server-side scripting* maka sintaks dan perintah – perintah *PHP* akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan halaman web lebih terjamin. *PHP* dirancang untuk membentuk halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web.

PHP Merupakan singkatan recursive dari PHP : Hypertext Preprocessor Pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994, penulisan harus ditulis di antara tag :

<? dan ?>

<?php dan ?>

<script language="php"> dan </script>

<% dan %>

Setiap satu statement (perintah) biasanya diakhiri dengan titik-koma (;) CASE SENSITIVE untuk nama identifier yang dibuat oleh user (berupa variable, konstanta, fungsi dll), namun TIDAK CASE SENSITIVE untuk identifier built-in dari PHP. (M. Rudyanto Arief ; 2011 : 43).