

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Dasar Sistem

II.1.1. Pengertian Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut.

“Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu” (Jogiyanto HM, 2005 : 1).

Prosedur didefinisikan oleh Ricard F. Neuschel sebagai suatu urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

Pendekatan yang lebih menekankan pada elemen atau komponennya didefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu. Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam

mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

Dapat disimpulkan sistem adalah kegiatan-kegiatan yang saling berhubungan antara satu sama yang lainnya yang terdiri dari objek-objek, unsur-unsur atau komponen-komponen sehingga membentuk suatu kesatuan pemrosesan untuk mencapai tujuan tertentu.

II.1.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*) dan keluaran (*ouput*). Karakteristik sistem adalah :

1. Komponen-komponen (*components*) suatu sistem yang saling berinteraksi, yang artinya saling bekerjasama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa satu subsistem atau bagian-bagian dari sistem.
2. Batas sistem (*Boundry*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

3. Lingkungan luar sistem (*Enviroments*) adalah apapun diluar sistem batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem tersebut. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem .
4. Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem yang lainnya.
5. Masukan (*Input*) adalah energi yang dimasukkan kedalam sistem masukan dapat berupa masukan perawatan dan masukan sinyal. *Maintenen* input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer.
6. Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan (Jogiyanto HM, 2005 : 3-4).

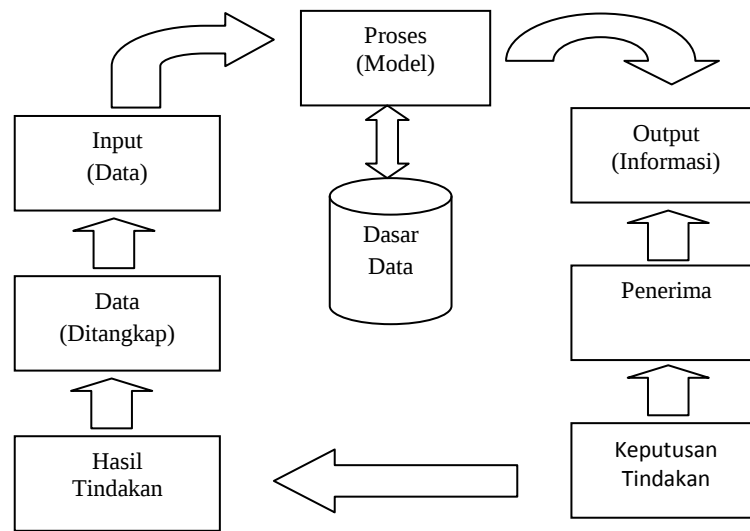
II.2. Pengertian Informasi

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya berakhir. Robert N. Anthony dan John Dearden menyebutkan keadaan dari sistem dalam hubungannya dengan keberakhirannya dengan istilah *entropy*. Informasi yang berguna bagi sistem akan menghindari proses *entropy* yang disebut dengan *negative entropy* atau *negentropy*. Informasi (*information*) dapat didefinisikan sebagai berikut :

“Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya” (Jogiyanto HM, 2005 : 8).

II.2.1. Siklus Informasi

Pertama-tama data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*). Siklus ini disebut juga dengan siklus pengolahan data (*data processing cycles*).



Gambar II.1. Siklus Informasi

Sumber : Jogiyanto HM (2005 : 9)

Kualitas dari suatu sistem informasi (*quality of information*) ditentukan oleh tiga faktor yaitu sebagai berikut.

1. Relevan

Relevan berarti informasi mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya adalah berbeda. Misalnya informasi mengenai harga produksi untuk ahli teknik perusahaan merupakan informasi yang kurang relevan jika ditujukan kepada akuntan.

2. Tepat waktu

Tepat waktu berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi sebab informasi merupakan landasan dalam mengambil keputusan.

3. Akurat

Akurat berarti informasi harus bebas dari kesalahan dan tidak menyesatkan harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi.

II.3. Konsep Dasar Sistem Informasi

II.3.1. Pengertian Sistem Informasi

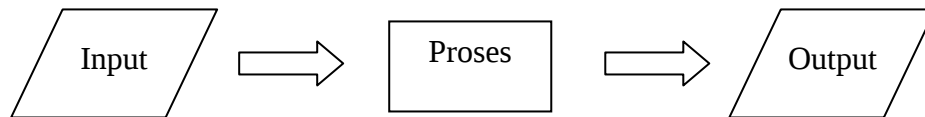
Informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah dimana informasi tersebut didapat. Informasi dapat diperoleh dari sistem informasi. Robert A Leitch dan K.Roscoe Davis mendefinisikan sistem informasi sebagai berikut:

“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” (Jogiyanto HM, 2005 : 11).

II.3.2. Komponen-Komponen SI

Komponen suatu sistem informasi terdiri dari input, proses dan output. Input dalam sistem informasi adalah data-data yang relevan untuk menghasilkan

informasi yang diinginkan. Proses adalah langkah-langkah yang perlu dilakukan untuk mengolah data menjadi informasi. Sedangkan output adalah berupa informasi yang merupakan hasil dari pemrosesan data.



Gambar II.2. Komponen Sistem Informasi

Sumber : Diana Anastasia (2011 : 4)

II.4. Konsep Dasar Sistem Informasi Akuntansi (SIA)

II.4.1. Pengertian SIA

Sistem informasi, yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen baik manual ataupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut.

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan (Diana Anastasia 2011 : 4).

II.4.2. Tujuan SIA

Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta/kekayaan perusahaan.
2. Menghasilkan beragam informasi untuk mengambil keputusan.
3. Menghasilkan informasi untuk pihak eksternal.
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian.

II.5. Piutang

II.5.1. Pengertian Piutang

Piutang adalah klaim perusahaan atas uang, barang atau jasa kepada pihak lain akibat transaksi di masa lalu.

Walaupun terdapat begitu banyak macam piutang yang mungkin dimiliki suatu perusahaan, tetapi berdasarkan jenis dan asal piutang, maka piutang di dalam perusahaan dapat diklasifikasikan ke dalam dua kelompok, yaitu :

1. Piutang usaha adalah piutang yang timbul dari penjualan barang dan jasa yang dihasilkan perusahaan. Dalam kegiatan normal perusahaan, piutang usaha biasanya akan dilunasi dalam tempo kurang dari satu tahun, oleh karena itu piutang usaha dikelompokkan ke dalam kelompok aktiva lancar.
2. Piutang bukan usaha adalah piutang yang timbul bukan sebagai akibat penjualan barang atau jasa yang dihasilkan perusahaan. Termasuk dalam kelompok ini adalah :
 - a. Persekot dalam kontrak pembelian.
 - b. Klaim terhadap perusahaan angkutan untuk barang rusak atau hilang.
 - c. Klaim terhadap perusahaan asuransi atau kerugian yang di pertanggungjawabkan.
 - d. Klaim terhadap karyawan perusahaan.
 - e. Klaim terhadap restitusi pajak.
 - f. Piutang Dividen.

II.5.2. Pencatatan Piutang

Sesuai dengan standar akuntansi keuangan yang berlaku di Indonesia, piutang dicatat dan diakui sebesar jumlah bruto (nilai jatuh tempo) dikurangi dengan taksiran jumlah yang tidak akan diterima. Itu berarti piutang harus dicatat sebesar jumlah yang diharapkan akan dapat ditagih. Karena itu berkaitan dengan pengelolaan piutang, perusahaan harus membuat suatu cadangan piutang tak tertagih yang merupakan taksiran jumlah piutang yang tidak akan dapat ditagih dalam periode tersebut (Rudianto, 2008 : 224-226).

Dalam membuat cadangan kerugian piutang atau piutang tidak tertagih, terdapat dua dasar utama yang dapat digunakan, yaitu :

1. Jumlah penjualan (persentase tertentu dari penjualan), berarti cadangan kerugian piutang didasarkan pada persentase tertentu dari saldo akun penjualan pada saat cadangan kerugian piutang tersebut disusun, atau didasarkan pada persentase tertentu dari taksiran jumlah penjualan atau jumlah penjualan kredit pada suatu periode tertentu.
2. Saldo piutang :
 - a. Persentase tertentu dari saldo piutang, berarti cadangan kerugian piutang didasarkan pada saldo akun piutang pada saat piutang tersebut disusun atau didasarkan pada taksiran penjualan kredit pada periode yang bersangkutan.
 - b. Analisis umur piutang, adalah suatu metode pembuatan cadangan kerugian piutang di mana cadangan piutang yang tidak dapat ditagih dari suatu

perusahaan didasarkan pada besarnya risiko atau kemungkinan tidak tertagihnya suatu piutang. Dasar dari metode ini adalah pemikiran bahwa semakin lama umur suatu piutang, maka semakin besar pula kemungkinan terjadinya kemacetan proses penagihan piutang tersebut.

II.5.3. Penghapusan Piutang

Biasanya cadangan kerugian piutang dibuat pada awal periode akuntansi. Cadangan kerugian piutang merupakan suatu taksiran besarnya piutang yang tidak dapat ditagih pada suatu periode akuntansi. Setelah periode tersebut berjalan, sering kali terdapat sejumlah piutang yang benar-benar tidak dapat ditagih karena berbagai alasan. Piutang yang jelas-jelas tidak dapat ditagih karena debiturnya lari, meninggal, bangkrut atau sebab lain harus dihapuskan dari saldo piutang. Penghapusan piutang ini merupakan kerugian karena pencatatannya tidak dibebankan ke akun kerugian piutang tetapi ke akun cadangan kerugian piutang.

Untuk menghapus suatu piutang terdapat dua metode yang bisa digunakan, yaitu:

1. Metode Cadangan Kerugian Piutang (seperti yang telah dibahas sebelumnya).
2. Metode penghapusan langsung, adalah metode penghapusan piutang dengan cara menunggu sampai diperoleh kepastian bahwa piutang tersebut benar-benar tidak dapat ditagih, tanpa perlu dibuat estimasinya lebih dahulu.

Jurnal yang perlu dibuat berkaitan dengan penghapusan piutang tersebut adalah sebagai berikut :

Tabel II.1 Jurnal cadangan dan penghapusan langsung

Metode cadangan		Metode penghapusan langsung	
Beban kerugian piutang	Rp. xxxxx	Tidak ada jurnal	
Cadangan kerugian piutang	Rp. xxxxx		
Cadangan kerugian piutang	Rp. xxxxx	Kerugian piutang	Rp. xxxxx
Piutang	Rp. xxxxx	Piutang	Rp. xxxxx

Sumber : Rudianto (2009 : 229)

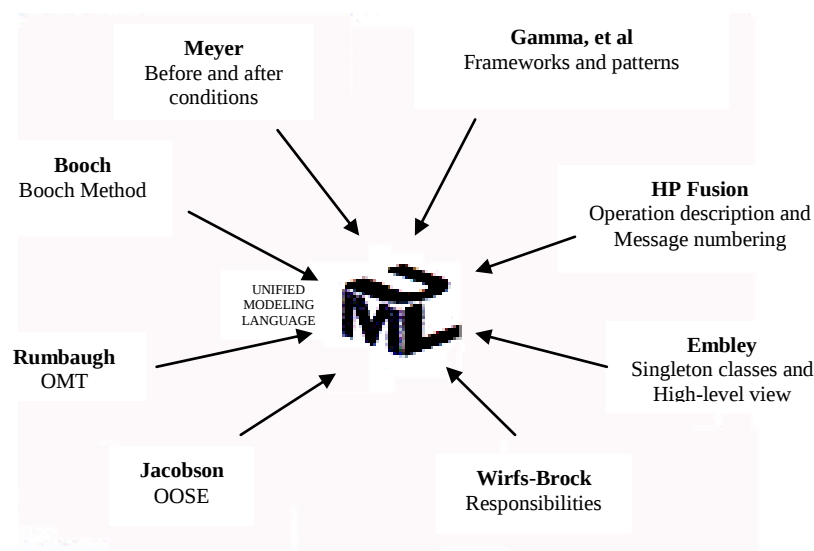
II.6. *Unified Modelling Language (UML)*

II.6.1. Pengenalan UML

Unified Modelling Language (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan baik (Munawar, 2005 : 17).

UML merupakan kesatuan bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique (OMT)* dan *Object Oriented Engineering (OOSE)*. Metode Booch dari *Grady Booch* sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu: identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian interface dan implementasi. Keuntungan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen.

Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambahkan dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya. Unsur-unsur yang membentuk UML ditunjukkan dalam Gambar II.3



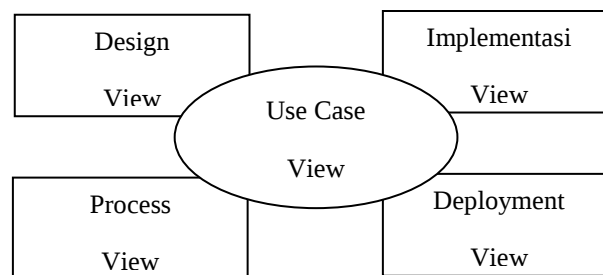
Gambar II.3 Unsur-unsur yang membentuk UML

Sumber : Munawar (2005 : 18)

UML adalah hasil kerja dari konsorsium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam OOAD (*Object Oriented Analysis dan Design*). UML tidak hanya dominan dalam penotasian di lingkungan OO tetapi juga populer di luar lingkungan OO. Ada tiga karakter penting yang melekat di UML yaitu sketsa, cetak biru dan bahasa pemrograman. Sebagai sebuah sketsa UML bisa berfungsi sebagai sebuah cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi detail tentang coding

program (*Forward engineering*) atau bahkan membaca program yang tidak terdokumentasi asli hilang atau bahkan belum dibuat sama sekali. Sebagai bahasa pemrograman, UML dapat menterjemahkan diagram yang ada di UML menjadi kode program siap untuk dijalankan.

UML dibangun atas model 4+1 view. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam 5 view dimana salah satu diantaranya *use case view*. *Use case view* ini memegang peran khusus untuk mengintegrasikan *content* ke view yang lain. Model 4+1 view ditunjukkan pada gambar II.4



Gambar II.4. Model 4+1 View

Sumber : Munawar (2005 : 20)

Kelima *view* tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap *view* berhubungan dengan perspektif tertentu dimana sistem akan diuji. *View* yang berbeda akan menekankan pada aspek yang berbeda dari sistem yang mewakili tentang sistem bisa dibentuk dengan menggabungkan informasi-informasi yang ada pada kelima *view* tersebut.

Use case view mendefinisikan perilaku eksternal sistem. Hal ini menjadi daya tarik bagi *end user*, analis dan *teste*. Pandangan ini mendefinisikan kebutuhan sistem karena mengandung semua *view* yang lain yang mendeskripsikan aspek-aspek tertentu dari peran dan sering dikatakan yang mendrive proses pengembangan perangkat lunak.

Design view mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan di *use case*. *Design view* ini berisi definisi komponen program, *class-class* utama bersama-sama dengan spesifikasi data, perilaku dan interaksinya. Informasi yang terkandung di *view* ini menjadi perhatian para programmer karena menjelaskan secara detil bagaimana fungsionalitas sistem akan diimplementasikan.

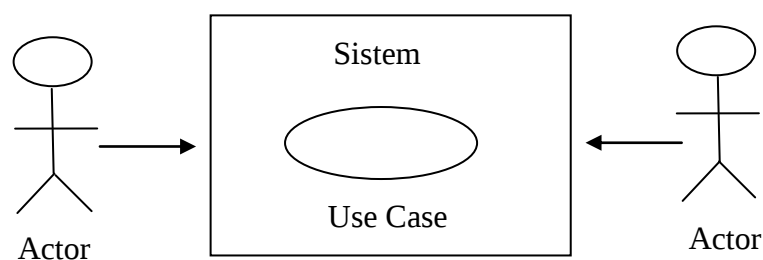
Implementasi view menjelaskan komponen-komponen fisi dari sitem yang akan dibangun. Hal ini berbeda dengan komponen logic yang dideskripsikan pada *design view*. Termasuk disini diantaranya *file exe*, *library* dan *database*. Informasi yang ada di *view* dan integrasi sistem.

Proses view berhubungan dengan hal-hal yang berkaitan dengan *concurrency do* dalam sistem. Sedangkan *deployment view* menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan dijalankan. Kedua *view* ini menunjukkan kebutuhan non fungsional dari sistem seperti toleransi kesalahan dan hal-hal yang berhubungan dengan kinerja (Munawar, 2005 : 17-21).

II.6.2. Diagram – diagram Pada Metode UML

1. Use Case Diagram

Use case adalah alat bantu terbaik guna menstimulasikan pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu. Ide dasarnya adalah bagaimana melibatkan penggunaan sistem di *fase-fase* awal analisis dan perancangan sistem. Diagram *use case* menunjukkan 3 aspek dari sistem yaitu *actor*, *use case* dan sistem / sub sistem *boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*. Gambar II.5 mengilustrasikan *actor*, *use case* dan *boundary*.



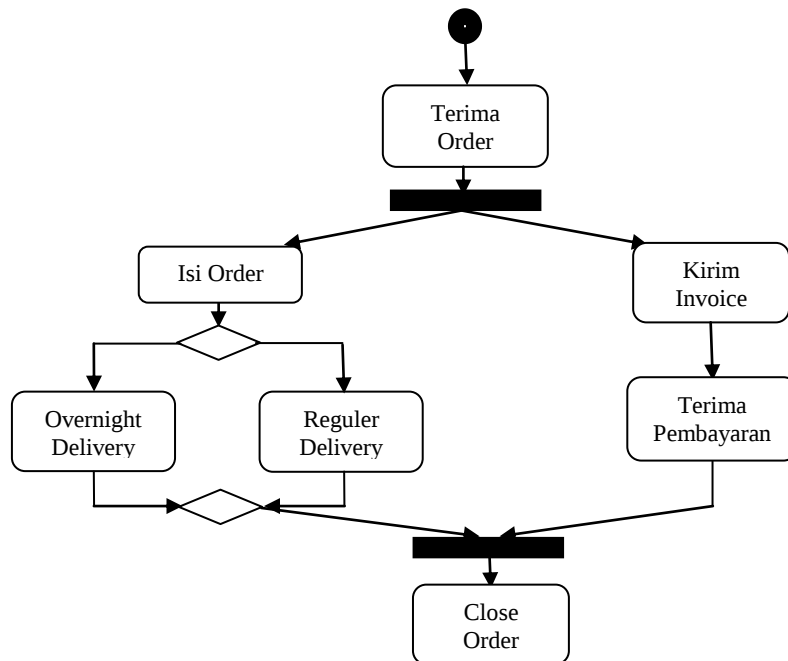
Gambar II.5. Use Case Model

Sumber : Munawar (2005 : 64)

2. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram*

mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut gambar dari sederhana dari *Activity diagram*.



Gambar II.6 Contoh Activity Diagram Sederhana

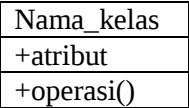
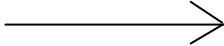
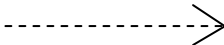
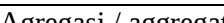
Sumber : Munawar (2005 : 111)

3. Class Diagram

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.

- a. Atribut merupakan varabel-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel II.2 Simbol-Simbol pada Class Diagram

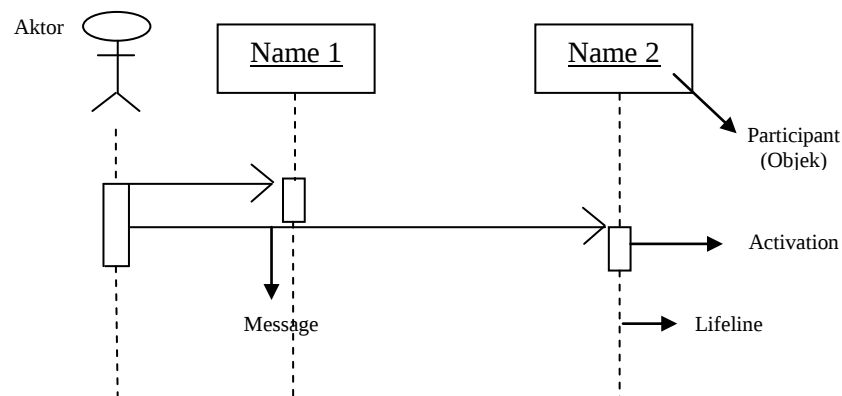
Simbol	Deskripsi
Kelas 	kelas pada struktur sistem
Antarmuka / interface  Nama_interface	sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi / association 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah / directed association 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / dependency 	relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / aggregation 	relasi antar kelas dengan makna

Sumber : Rosa A.S.M.Shalahuddin (2011 : 123)

4. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan pesan yang diletakan diantara obyek-obyek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical* (Munawar, 2005 : 87).



Gambar II.7 Simbol-Simbol pada Sequence Diagram

Sumber : Munawar (2005 : 89)

II.7. PHP

II.7.1. Sekilas Sejarah PHP

PHP (*PHP Hypertext Preprocessor*) adalah kode/skrip yang akan dieksekusi pada *server side*. Skrip PHP akan membuat suatu aplikasi dapat di-integrasikan 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).

Penggunaan PHP memungkinkan Web dapat dibuat dinamis sehingga *maintenance* situs Web tersebut menjadi lebih mudah dan efisien. PHP merupakan *software Open-Source* yang disebarakan dan dilisensikan secara gratis serta dapat di-*download* secara bebas dari situs resminya.

PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada *homepage*-nya. Rasmus Lerdorf adalah salah satu pendukung open source. Oleh karena itu, ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis, kemudian menambah kemampuan PHP 1.0 dan meluncurkan PHP 2.0.

Pada tahun 1996, PHP telah banyak digunakan dalam *website* di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken, Shane Caraveo, dan Jim Winstead bekerja sama untuk menyempurnakan PHP 2.0. Akhirnya, pada tahun 1998, PHP 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0 (Kasiman Peranginangin : 2006 : 2).

II.7.2. Kelebihan - Kelebihan PHP

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa script sejenis PHP difokuskan pada pembuatan script *server-side* yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih daripada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain *Linux*, *Unix* (termasuk variannya *HP-UX*, *Solaris*, dan *OpenBSD*), *Microsoft Windows*, *Mac*

OS X. PHP juga mendukung banyak *Web Server*, seperti *Apache*, *Microsoft Internet Information Server (MIIS)*, *Personal Web Server (PWS)*, *Netscape and iPlanet servers*, *Oreilly Website Pro Server*, *audium*, *Xitami*, *OmniHTTPd*, dan masih banyak lainnya.

PHP tidak terbatas pada hasil keluaran HTML (*Hypertext Markup Language*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, File PDF, dan *movies Flash*. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya.

Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungannya terhadap banyak *database*, seperti *Adabas D*, *dBase*, *Direct MS-SQL*, *Empress*, *FilePro*, *FrontBase*, *Hyperware*, *IBM DB2*, *Informix*, *MSQL*, *MySQL* dan masih banyak lagi (Kasiman Peranginangin : 2006 : 3-4).

II.7.3. Sintaks PHP

Sintaks program / script PHP ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok script PHP (Kasiman Peranginangin : 2006 : 4):

1. `<?PHP ... ?>`
2. `<script language = "PHP"> ... </script>`
3. `<? ...?>`
`<% ... %>`

II.8. MySQL

MySQL (*My Strukture Query Language*) adalah DBMS yang didistribusikan secara gratis dibawah lisensi dari *General Public License* (GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan bersifat *close source* (komersial).

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, pengubahan dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis (Deni Sutaji, 2012 : 40).

II.9. Database (basisdata)

Basisdata adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasikan dan dihapus (Janner Simarmata, 2006 : 1).

II.10. Konsep Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang teduplikasi dari tabel relasional (Janner Simarmata, 2006 : 77).

Pada waktu menormalisasi basisdata, ada empat tujuan yang harus dicapai, yaitu :

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimalkan jumlah data berulang dalam basisdata.
3. Membuat basisdata yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

Langkah – langkah normalisasi :

1. Bentuk Normal Pertama (1NF)

Sebuah tabel relasional secara definisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomik. Ini berarti kolom-kolom tidak mempunyai nilai berulang.

2. Bentuk Normal Kedua (2NF)

Normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk Normal Ketiga (3NF)

Bentuk normal ketiga menghapuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

4. Bentuk Normal *Boyce-Code* (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Bentuk Normal *Boyce-Code* (BCNF) adalah versi 3NF yang lebih teliti dan berhubungan dengan tabel relasional yang mempunyai :

- a. Banyak kunci kandidat.
- b. Kunci kandidat gabungan.

c. Kunci kandidat yang saling tumpang tindih.

5. Bentuk Normal Keempat (4NF)

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

II.11. Entity Relationship Diagram (ERD)

II.11.1. Pengertian ERD

Entity Relationship Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata, 2006 : 67).

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. Entitas (Entity)

Entitas adalah sesuatu yang nyata atau abstrak di mana kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai, pembayaran, kampus dan buku. Contoh suatu entitas disebut instansi, misalnya pegawai Adi, Pembayaran Joko dan lain sebagainya.

2. Relasi (Relationship)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya, mahasiswa bisa mengambil banyak mata kuliah.

3. Atribut (Attribute)

Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah properti, elemen data dan field. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasikan satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya, nomor pegawai adalah kunci utama untuk pegawai.

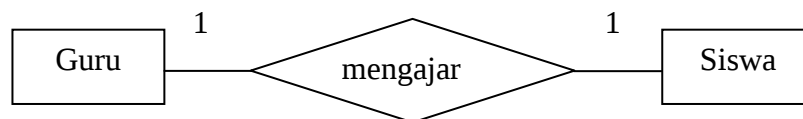
II.11.2. Pemetaan Kardinalitas

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

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

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

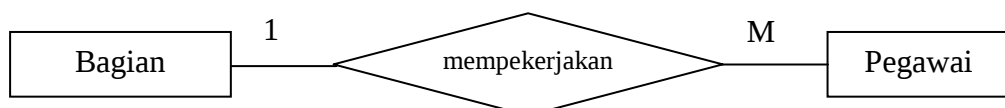
1. One-to-One, misalnya suatu pengajaran privat mempunyai satu guru satu siswa. Seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru.



Gambar II.8 Relational One to One

Sumber : Janner Simarmata (2006 : 64)

2. One-to-Many, misalnya dalam suatu perusahaan, satu bagian mempekerjakan banyak pegawai. Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian.



Gambar II.9 Relational One to Many

Sumber : Janner Simarmata (2006 : 64-65)

3. Many-to-Many, misalnya dalam universitas seorang mahasiswa dapat mengambil banyak mata kuliah. Satu mahasiswa mengambil banyak mata kuliah dan satu mata kuliah diambil banyak mahasiswa.



Gambar II.10 Relational Many to Many

Sumber : Janner Simarmata (2006 : 65-66)

II.12. Kamus Data

II.12.1. Pengertian KD

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke distem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang input, merancang laporan-laporan dan database (Jogiyanto HM, 2005 : 725).

II.12.2. Isi KD

KD harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka KD harus memuat hal-hal berikut ini :

1. Nama Arus Data.

Karena KD dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di KD, sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DAD dapat langsung mencarinya dengan mudah di KD.

2. Alias.

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada.

3. Bentuk data.

Bentuk dari data ini perlu dicatat di KD, karena dapat digunakan untuk mengelompokkan KD ke dalam kegunaannya sewaktu perancangan sistem.

4. Arus Data.

Arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju.

5. Penjelasan.

Untuk lebih menjelaskan lagi tentang makna dari arus data yang dicatat di KD.

6. Periode.

Periode ini menunjukkan kapan terjadinya arus data ini.

7. Volume.

Volume yang perlu dicatat di KD adalah tentang volume rata-rata dan volume puncak dari arus data.

8. Struktur data.

Struktur data menunjukkan arus data yang dicatat di KD terdiri dari item-item data apa saja.