BAB II

LANDASAN TEORI

II.1. Sistem

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujua tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan. Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri. Sebagai contoh, sistem peredaran darah manusia merupakan sistem biologi yang memiliki tujuan untuk mengedarkan darah yang mengandung oksigen dan sari makanan ke seluruh tubuh. Sedangkan sistem buatan manusia seperti sistem yang terdapat di sekolah, organisasi bisnis, atau instansi pemerintah juga mempunyai tujuan yang berbeda-beda (Anastasia Diana dan Lilis Setiawati ; 2011;3).

Selain itu, sistem juga dapat didefinisikan sebagai sekumpulan objek-objek yang saling berelasi dan berinteraksi, serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai tujuan yang telah ditetapkan. Karakteristik suatu sistem:

1. Komponen atau elemen *(Component)*

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

2. Batas Sistem *(Boundary)*

   Batas sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Dengan kata lain, batas sistem merupakan ruang lingkup atau *scape* dari sistem atau subsistem itu sendiri.

3. Lingkungan Luar Sistem *(Environment)*

   Lingkungan luar merupakan segala sesuatu diluar batas sistem yang mempengaruhi operasi atau sistem. Lingkungan luar dapat besifat menguntungkan atau merugikan.

4. Penghubung Sistem *(Interface)*

   Penghubung sistem merupakan suatu media (penghubung) antara satu subsistem dengan subsistem lainnya yang membentuk satu kesatuan. Dengan kata lain, melalui penghubung, output dari subsistem akan menjadi input bagi subsistem lainnya.

5. Masukkan *(Input)*

   Input adalah energi atau sesuatu yang dimasukkan ke dalam suatu sistem yang dapat berupa masukkan yaitu energi yang dimasukkan supaya sistem dapat beroperasi atau masukkan sinyal yang merupakan energi yang diproses untuk menghasilkan suatu luaran.

6. Luaran *(Output)*

   Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi luaran yang berguna, juga merupakan luaran atau tujuan akhir dari sistem.

7. Pengolah *(Process)*

   Suatu sistem mempunyai bagian pengolah yang akan mengubah *input* menjadi *output.*

8. Sasaran *(Objekive)*

   Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

Sistem yang baik adalah sistem yang selalu menyesuaikan dengan perubahan lingkungan disekitarnya. Sistem tersebut harus dinamis menuju pada keadaan yang lebih baik secara berkelanjutan (Hamim Tohari ; 2014 ; 2-5).

II.2. Informasi

Sutabri, dalam bukunya *Sistem Informasi Manajemen* menjelaskan bahwa, informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (2002).

Informasi adalah data yang telah diproses sedemikian rupa, sehingga memiliki arti yang lebih bermanfaat bagi penggunanya. Informasi merupakan aset penting bagi suatu institusi atau organisasi. Oleh karena itu, informasi harus berkualitas, dijaga, dan dipelihara dengan baik.

Sedangkan kualitas dari suatu informasi tergantung dari tiga hal:

1. Akurat

   Informasi harus bebas dari kesalahan-kesalahan. Akurat harus mencerminkan maksud dan penyampaiannya harus akurat, dari sumber sampai penerima informasi.

2. Tepat waktu

   Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah tidak berlaku tidak akan mempunyai nilai lagi karena informasi merupakan landasan diadakannya pengambilan keputusan.

3. Relevan

   Informasi harus mempunyai manfaat bagi pemakainya (Hamim Tohari ; 2014 : 7-8).

II.3. Akuntansi

Akuntansi merupakan proses mengidentifikasi, mengukur, mencatat dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi (bisnis maupun nonbisnis) kepada pihak-pihak yang berkepentingan dengan informasi bisnis tersebut (pengguna informasi). Pada dasarnya fokus utama dari akuntansi adalah transaksi bisnis (Anastasia Diana dan Lilis Setiawati ; 2011; 14).

II.4. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan

dengan saksi keuangan. 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.

   Harta / kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetapa perusahaan.

2. Menghasilkan beragam informasi untuk pengambilan keputusan.

   Sebagai contoh, pengelola toko swalayan memerlukan informasi mengenai barang apa yang diminati oleh konsumen. Membeli barang dagangan yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.

3. Menghasilkan informasi untuk pihak eksternal.

   Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan atau tergantung pada laba rugi usaha. Selain untuk kepentingan perpajakan, adakalanya pengelola usaha juga terlibat dengan kegiatan utang piutang dengan bank atau koperasi simpan pinjam. Bank membutuhkan informasi omset dan laba rugi usaha untuk memutuskan besarnya yang akan diberikan.

4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.

   Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi. Sebagai contoh, pengelola toko swalayan dapat

memanfaatkan data penjualan untuk menilai kinerja kasir. Apresiasi pada karyawan yang rajin berguna untuk memotivasi dan meminimalkan sikap malas-malasan ditempat kerja.

5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan).

   Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan). Satu hal yang penting, audit bukan eksklusif milik perusahaan publik. Semua perusahaan harus siap untuk menghadapi pemerriksaan.

6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan.

   Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas. Anggaran berperan dalam menerapkan skala prioritas pengeluaran sesuai dengan tujuan perusahaan. Dengan adanya sistem informasi akuntansi ini, dapat mempermudah pengawasan pengeluaran, apakah sudah melewati batas anggaran yang telah disetujui.

7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian.

   Sistem informasi akuntansi ini dapat digunakan untuk meramal pertumbuhan penjualan dan aliran kas atau untuk mengetahui tren jangka panjang beserta korelasinya (Anastasia Diana dan Lilis Setiawati ; 2011).

Secara umum Sistem informasi akuntansi membantu manajemen perusahaan untuk mengumpulkan data-data keuangan, mengolahnya menjadi informasi yang bermanfaat bagi pengguna, dan menghasilkan laporan keuangan.

Sistem Informasi Akuntansi yang baik dan efektif mempukan manajemen perusahaan dan para pihak yang berkepentingan mendapatkan informasi secara cepat dan akurat mengenai peusahaan, seperti dalam hal:

a. Besarnya kas yang dimiliki perusahaan;
b. Besar saldo utang yang harus dilunasi perusahaan;
c. Banyaknya aset yang dimiliki perusahaan ;
d. Besarnya laba yang dihasilkan perusahaan;
e. Besarnya dividen yang bisa dibagikan kepada perusahaan;
f. Kinerja operasional perusahaan.

Untuk perusahaan berskala kecil, SIA dapat berbentuk pencatatan manual atau semi manual dengan menggunakan program komputer seperti MS Excel. Untuk perusahaan besar, implementasi SIA memerlukan program dan manajemen basis data *(database)* khusus yang biasanya memerlukan dukungan dari perusahaan piranti lunak *(software)* seperti Oracle atau SAP (Dwi Martini & Sylvia Veronica NPS ; 2012 ; 60).

II.5. Petty Cash (Kas Kecil)

Pada umumnya pengendalian internal atas pengeluaran kas akan lebih efektif ketika pembayaran dilakukan dengan menggunakan cek atau transfer lewat rekening bank, dari pada dengan melibatkan uang kas secara langsung. Pengecualian dibuat untuk pengeluaran-pengeluaran tertentu yang jumlahnya relatif kecil, dimana pengeluaran-pengeluaran ini dapat dibiayai langsung dengan menggunakan dana kas kecil *(petty cash).* Alasan perlu dibuatnya (dibentuknya) sebuah sistem dana kas kecil adalah bahwa pembayaran-pembayaran yang jumlahnya relatif kecil ini, yang sering terjadi, mungkin pada akhirnya juga dapat menjadi suatu jumlah tertentu yang cukup signifikan jika ditotal. Oleh sebab itu agar pengeluaran-pengeluaran ini juga dapat dimonitor dengan baik maka pengendalian internal mutlak diperlukan, caranya adalah dengan membentuk sistem dana kas kecil (Hery, S.E., M.Si. ; 2014 ; 195).

Untuk keperluan pengeluaran dalam jumlah kecil, entitas tidak mungkin melakukannya dengan menggunakan cek karena tidak efisien. Untuk memenuhi kebutuhan pengeluaran kas dalam jumlah kecil entitas membentuk dana kas kecil. Jumlah dana kas kecil disesuaikan dengan kebutuhan entitas. Semakin besar ukuran entitas dan kebutuhan pengeluaran dana kas kecil besar, maka akan dibentuk kas kecil dalam jumlah besar (Dwi Martani & Sylvia Veronica NPS ; 2012 ; 182).

II.6. Metode*Imprest*

Metode *imprest* kas kecil adalah mekanisme kas kecil dimana dana dipertahankan tetap. Pada awalnya dibentuk dana kas kecil dalam jumlah tertentu, setiap ada pengeluaran akan dibuat bukti pengeluaran tetapi tidak dibuat jurnal. Dalam rangka pengendalian, metode *imprest* lebih baik, karena jumlah dana kas kecil akan terkontrol dan tidak akan terjadi penumpukan dana kecil dalam unit pembayaran (kasir). Mekanisme pengendali lebih baik, karena jumlah dana kas kecil akan terkontrol dan tidak akan terjadi penumpukan dana kecil dalam unit pembayaran (kasir). Mekanisme pengendalian juga terjadi, karena setiap penggantian akan dilakukan penghitungan dana kas kecil terpakai dan tersisa sehingga dapat memonitor pemakaian dan memastikan tidak ada uang yang hilan juga terjadi, karena setiap penggantian akan dilakukan penghitungan dana kas kecil terpakai dan tersisa sehingga dapat memonitor pemakaian dan memastikan tidak ada uang yang hilang (Dwi Martani & Sylvia Veronica NPS ; 2012 ; 182-183).

Dalam sistem *Imprest Fund*, jumlah dana dalam rekening kas kecil selalu tetap, yaitu sebesar cek yang diserahkan kepada kasir kas kecil untukmembentuk dana kas kecil. Biasanya kas kecil inidiisi dengan sejumlah uang yang telah ditetapkan untuk keperluan pembayaran-pembayaran selama jangka waktu tertentu, misalnya satu minggu, dua minggu, ataupun sebulan. Jika jangka waktunya telah habis dan jumlah uang dalam kas kecil telah menipis, maka kas kecil diisi kembali dengan menarik dana dari kas besar sampai dengan jumlahdana yang telah ditetapkan besarnya. Untuk setiap pengisian kembali dana

kas kecil, pemegang kas kecil selalu melampirkan kas kecil serta bukti-bukti pendukungnya (Januar Wibisono Wibowo, Anak Agung Gde Agung, &Raswyshnoe Boing K ; 2012; 152-153).

Dengan menggunakan metode *imprest* ini tidak ada ayat jurnal tambahan yang diperlukan atas akun kas kecil, kecuali manajemen perusahaan memang bermaksud untuk mengubahnya (menambah atau mengurangi) jumlah kas kecil yang sudah dibentuk, begitu pula untuk mencatat pembayaran kas kecil tidak ada ayat jurnalnya. Dana kas kecil akan diisi kembali pada interval periode tertentu atau ketika jumlah uang yang ada dalam dana kas kecil telah mencapai tingkat minimum. Berikut adalah ayat jurnal yang perlu dibuat apabila perusahaan menggunakan metode imprest :

Jurnal pembentukan dana kas kecil :

Kas Kecil	xxx	
Kas		xxx

Jurnal pengisian kembali dana kas kecil :

Ongkos Angkut Masuk	xxx	
Beban lainnya	xxx	
Kas		xxx

(Hery, S.E., M.Si. ;2014 ;196-198).

II.7. Aplikasi

Menurut Panji M. Sudarmo (2006 : 21) aplikasi yaitu sejenis tugas atau pekerjaan yang dilakukan suatu program atau system computer misalnya

perancangan teknik, system pemesanan tiket pesawat terbang, administrasi keuangan dan sebagainya (Arif Rohman dan Nining ; 2010 : 4).

II.7.1. Adobe Dreamweaver

Dalam membuat sebuah website memerlukan program aplikasi yang andal sekaligus mudah dalam membangun web tersebut. Salah satu program aplikasi yang sekarang ini banyak digunakan adalah Adobe Dreamweaver .

Adobe Dreamweaver adalah aplikasi yang memberikan tampila yang lebih baik dan tentu saja semakin mudah dalam penggunaannya, aplikasi ini mengintegrasikan beragam fitur untuk memenuhi kebutuhan pengembangan website, termasuk pembuatan halaman web dan pengelolaannya. Adobe Dreamweaver menyertakan banyak tool yang berkaitan dengan pengkodean seperti HTML, CSS, XML dan pemrograman Client Side, yaitu JavaScript dengan penggunaan yang sangat mudah dan *user Friendly.* Aplikasi ini juga mendukung pemrograman Script Server Side seperti PHP, Active Server Page (ASP), ASP.NET, ASP JavaScript, ASP VBScript, ColdFusion, dan Java Server Page (JSP).

Fasilitas yang ada pada Adobe Dreamweaver memberikan kemudahan kepada *user* untuk melakukan pengeditan karena ditampilkan secara visual. Penambahan desain dan fungsi pada halaman web tidak harus dituliskan dalam baris kode. Anda hanya tinggal memilih dan menempatkan komponen web dengan melakukan *drag* ke dokumen web secara langsung dan cepat. Terdapat

beberapa macam tipe file untuk format halaman web yang dapat Anda pilih, seperti:

1. HTML *(Hypertext Markup Language)*

   HTML merupakan dasar untuk pembuatan desain web. File HTMl berisi instruksi tertentu yang dapat memberikan suatu format dokumen yang akan ditampilkan pada *World Wide Web.*

2. ColdFusion

   Merupakan bahasa *scripting* yang digunakan oleh Adobe ColdFusion, BlueDragon, dan sebagainya untuk *scripting server-side.*

3. PHP *(Hypertext Prepocessor)*

   Adalah sebuah bahasa pemrograman yang umum dipakai untuk *scripting server-side.* PHP biasanya terpasang pada HTML, dengan bahasa pemrogrman ini dapat dibuat suatu web yang dinamis.

4. ASP VBSceript *(ASP Visual Basic Script)*

   VBScript merupakan bahasa scripting turunan dari bahasa pemrograman Visual Basic yang dapat digunakan untuk membuat sebuah aplikasi HTML (yang memiliki ekstensi .HTA).

5. XSLT (*Extensible Stylesheet Language Transformations*)

   Adalah bahasa pemrograman berdasar XML yang digunakan untuk transformasi dokumen XML menjadi dokumen XML atau format lainnya.

6. CSS (*Cascading Style Sheet*)

   Adalah bahasa *Style sheet* yang digunakan untuk mengatur tampilan halaman web dan ditulis dengan HTML atau XHTML. CSS dapat juga digunakan untuk semua jenis dokumen XML termasuk SVG dan XUL.

7. JavaScript

   Adalah bahasa scipting yang mempunyai kesamaan dengan penggunaan sintaks bahasa pemrograman C. Script ini umum digunakan untuk pengembangan *web client-side.*

8. XML *(Extensible Markup Language)*

   Menggunakan markup tags seperti halnya HTML, namun penggunaannya tidak terbatas pada halaman web saja.

9. Dreamweaver Site

   Berfungsi untuk membuat website baru dengan program Dreamweaver. Dreamweaver adalah suatu bentuk program editor web yang dibuat oleh Macromedia. Dengan menggunakan program ini, seorang programmer web dapat dengan mudah membuat dan mendesain webnya, karena bersifat WYSIWYG (What You See Is What You Get). (Uswatun Hasanah, Sukadi 2013 : 8).

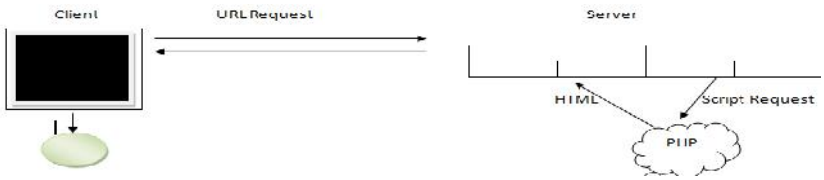
II.7.2. PHP

PHP adalah salah satu bahasa pemrograman web yang banyak digunakan. Kemudahan dan fleksibilitas dalam mempelajari dan implementasi pada web, menjadikan bahasa PHP banyak diminati oleh para programmer web. PHP

merupakan salah satu bahasa pemrograman web *server-side* yang populer dan banyak digunakan sampai saat ini.

PHP dirancang oleh Rasmus Ledrof pada tahun 1954. Awalnya PHP digunakan untuk mendeteksi *user* yang berkunjung pada situs. Selain sebagai bahasa pemrograman web yang dapat dieksekusi sendiri, PHP juga dikenal sebagai *embedded language* Artinya, Anda dapat memasukkan kode PHP ke dalam bahasa HTML yang sebelumnya dikenal sebagai bahasa pembentuk halaman website.

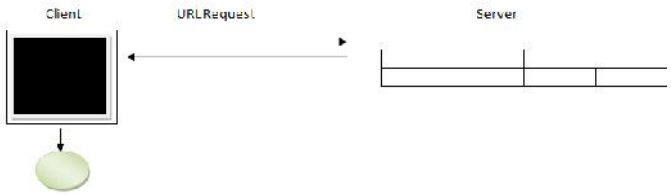
Sebagai *embedded language,* PHP juga hampir sama dengan bahasa pemrograman lain yang dapat dimasukkan ke dalam kode HTML seperti JavaScript VBScript. Perbedaannya terletak pada tempat eksekusi. Jika Java Script dieksekusi di sisi klien atau browser, PHP dieksekusi di sisi server. Karena eksekusi dilakukan pada sisi server, maka PHP dapat mengambil data, mengolah, dan mengirimkan data dari database server ke halaman browser. Berikut ini gambar II.1. dan gambar II.2. perbedaan proses yang terjadi pada website yang dinamis dan website statis.



Gambar II.1. Proses website dinamis
(Sumber : Eko Priyo Utomo; 2014; 3)

Pada gambar II.1. ketika klien meminta *request* ke *server*, maka akan dilakukan cek keberadaan skrip PHP. Jika ditemukan, maka akan dijalankan skrip PHP di server. Setelah proses selesai, skrip dikirimkan kembali ke klien dalam format HTML dan tampil pada layar monitor.

Berikut ini gambar II.2. proses website statis



Gambar II.2. Proses website statis
(Sumber : Eko Priyo Utomo; 214;3)

Pada website *statis*, request yang diminta langsung dikembalikan ke *client* dalam format HTML tanpa melakukan cek keberadaan skrip.

Salah satu keunggulan PHP adalah kemampuan mendukung berbagai macam database, antara lain: MySQL, Oracle, PostgreSQL, dBase, dan sebagainya. (Eko Priyo Utomo;2014;1-4).

II.8. Perancangan Database

Database atau basisdata di dalam buku Simarmata & Paryudi (2006:1), sebagai berikut:

1. Menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untukmenyimpan informasi atau data.
2. Menurut silberschatz, dkk (2002) mendefenisikan basisdata sebagai kumpulan databerisi informasi yang sesuai untuk sebuah perusahaan.

3. Menurut Ramakrishnan dan Gehrke (2003) menyatakan basisdata sebagai kumpulandata, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yangberhubungan.
4. Menurut McLeod, dkk (2001), adalah kumpulan seluruh sumber daya berbasiskomputer milik organisasi. (D. Tri Octafian ; 2011 ; 150).

Menurut James Martin (1975), *Database adalah A database may be defined as a collection of interrelated data stored together without harmful or unnecessary redundancy to serve one or more applications in an optimal fashion; the data are stored so that they are independent of programs with use the data; a common and controlled approach is used in adding new data and in modifying and retrieving existing data within the database.*Dengan memahami definisi diatas maka istilah basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data)* yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data, data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali; data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal; data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya; data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol. (Edhy Sutanta;2011:29).

Basis data merupakan bagian terpenting dalam sebuah sistem informasi. Basis data dalam sistem informasi dapat mempunyai peranan sebagai berikut:

1. Basis data sebagai komponen penyusun sistem informasi

   Basis data merupakan salah satu komponen terpenting dalam sebuah sistem informasi. Oleh karena itu, keberadaan basis data dalam sebuah sistem informasi adalah mutlak. Suatu sistem informasi tidak akan berfungsi bahkan tidak terwujud, tanpa melibatkan basis data.

2. Basis data sebagai infrastruktur sistem informasi

   Basis data dan sistem manajemen basis data *(Database Management)* menyediakan suatu sarana infrastruktur kepada organisasi-organisasi sistem informasi yang dibangun.

3. Basis data sebagai sumber informasi bagi sistem informasi

   Basis data mempunyai peran penting dalam sistem informasi, yaitu sebagai sumber penyedia data utama untuk memenuhi kebutuhan-kebutuhan informasi seluruh pemakai atau informasi bagi para pengambil keputusan.

4. Basis Data sebagai sarana mencapai efisiensi sistem informasi

   Basis data dirancang dan dibangun dengan orientasi kebutuhan informasi bagi seluruh pemakai dalam sistem.

5. Basis data sebagai sarana mencapai efektifitas sistem informasi

   Basis data memberikan dukungan bagi tercapainya efektivitas sistem informasi karena data-data yang disimpan sebagai file-file basis data yang hanya membuat data yang benar. (Edhy Sutanta;2011: 17-20).

II.8.1. Normalisasi *(Normalization)*

Normalisasi adalah sebagai suatu teknik yang menstrukturkan/ mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan *(anomallies)* yang terjadi akibat adanya kerangkapan data dalam relasi dan efisiensi pengolahan.

Proses normalisasi menghasilkan relasi yang optimal, yaitu:

1. Memiliki struktur *record* yang konsisten secara logik;
2. Memiliki struktur *record* yang mudah untuk dimengerti;
3. Memiliki struktur *record* yang sederhana dalam pemeliharaan;
4. Memlliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna;
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

II.8.1.1 Level Normalisasi

Level normalisasi adalah bentuk normal suatu relasi dijelaskan berdasarkan kriteria tertentu pada bentuk normal. Bentuk normal yang dikenal hingga saat ini meliputi bentuk sebagai berikut:

a. Relasi bentuk tidak normal *(un normalized form/UNF)*

   Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam definisi basis data dan karakteristik RDBM menghasilkan relasi UNF. Bentuk ini harus dihindari dalam perancangan relasi dalam basis data.

Tabel II.1. Proses Unnormalisasi (UNF)

Kode_Supplier	Status	Kota	Kode_Barang	Jumlah_Barang
S01	10	Jakarta	B01	100
			B02	150
			B03	200
S02	20	Surabaya	B02	250
			B04	200
S03	30	Yogyakarta	B05	150
			B06	100

(Sumber : Edhy Sutanta; 2011; 179)

b. Relasi bentuk normal pertama *(first norm form/1NF)*

Relasi disebut sebagai 1NF jika memenuhi kriteria sebagai berikut:

i. Jika seluruh atribut dalam relasi bernilai atomik *(atomic value)*

ii. Jika seluruh atribut dalam relasi bernilai tunggal *(single value)*

iii. Jika relasi tidak memuat set atribut berulang

iv. Jika semua *record* mempunyai sejumlah atribut yang sama.

Tabel II.2. Proses Normalisasi Dalam Bentuk 1NF

Kode_Supplier	Status	Kota	Kode_Barang	Jumlah_Barang
S01	10	Jakarta	B01	100
S01	10	Jakarta	B02	150
S01	10	Jakarta	B03	200
S02	20	Surabaya	B02	250
S02	20	Surabaya	B04	200
S03	30	Yogyakarta	B05	150
S03	30	Yogyakarta	B06	100

(Sumber : Edhy Sutanta; 2011; 180)

c. Bentuk normal kedua *(second norm form/2NF)*

Relasi disebut sebagai 2NF jika memenuhii kriteria sebagai berikut:

i. Jika memenuhi kriteria 1NF;

ii. Jika semua atribut nonkunci FD pada PK.

Tabel II.3. Proses Normalisasi Dalam Bentuk 2NF

Relasi Supplier

Kode_Supplier	Status	Kota
S1	10	Jakarta
S2	20	Surabaya
S3	30	Yogyakarta

Relasi Barang

Kode_Supplier	Kode_Barang	Jumlah_Barang
S1	P1	100
S1	P1	95
S2	P2	75
S2	P2	70
S3	P3	60
S3	P3	50

(Sumber : Edhy Sutanta; 2011; 181)

d. Bentuk normal ketiga *(third norm form/3NF)*

Suatu relasi disebut sebagai 3NF jika memenuhi kriteria sebagai berikut:

i. Jika memenuhi kriteria 2NF;

ii. Jika setiap atribut nonkunci tidak TDF *(non transitive dependency)* terhadap PK.

Tabel II.4. Proses Normalisasi Dalam Bentuk 3NF

Relasi Supplier

Kode_Supplier	Status
S1	10
S2	20
S3	30

Relasi Kota

Status	Kota
10	Jakarta
20	Surabaya
30	Yogyakarta

(Sumber : Edhy Sutanta; 2011; 182)

II.9. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, *et. al*. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, *et. al.* 2004).

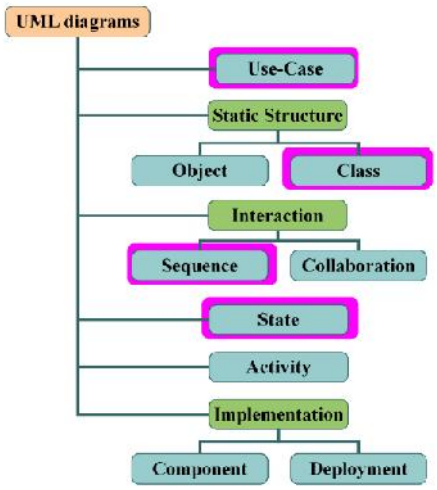
Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh Object Management Group (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rasional Software

Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rasional.

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. 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;1-2).

Berikut ini adalah gambar II.3 dari diagram UML :



Gambar II.3. Diagram UML

(Sumber : Haviluddin, 2011)

II.9.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapan oleh Schmuller J. (2004), *"The purpose of the diagrams is to present multiple views of a system; this*

set of multiple views is called a model". Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

II.9.2. Jenis-jenis Diagram UML

Berikut adalah beberapa diagram yang ada pada UML :

1. Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.

Hal yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* menyatakan sebuah aktivitas atas pekerjaan tertentu. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain. Oleh karena itu, duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common.* Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behavior*-nya sendiri. Hubungan generalisasi antar *use case* menunjukan bahwa *use case* merupakan spesialisasi dari yang lain (Hamim Tohari, 2014;47-48).

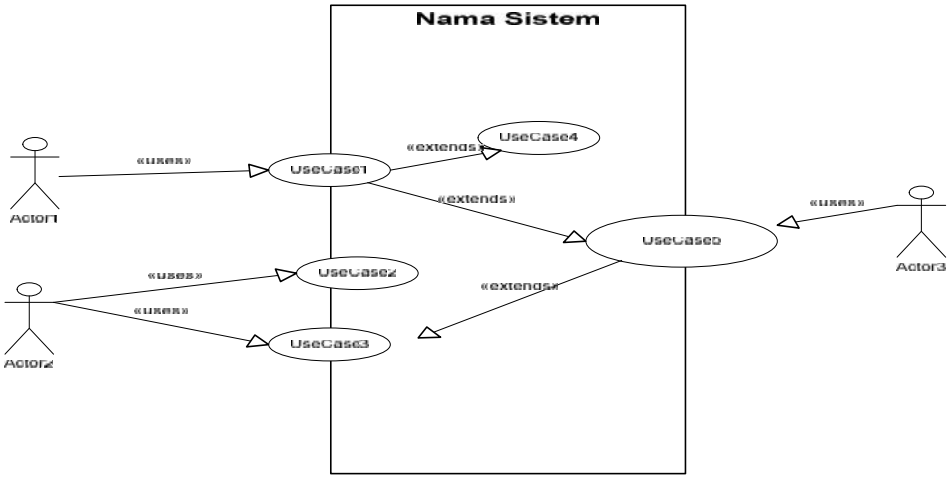
a. Menyusun *Use Case* Diagram

Langkah-langkah yang dibutuhkan untuk menyusun diagram *use case* :

1. Mengidentifikasi pelaku bisnis.
2. Mengidentifikasi *use case* persyaratan bisnis.
3. Membuat diagram model *use case.*
4. Mendokumentasikan naratif *use case* persyaratan bisnis (Hamim Tohari, 2014;49).

b. Elemen-Elemen *Use Case* Diagram

Beberapa elemen yang digunakan pada diagram *use case* dapat dilihat pada gambar II.4 berikut.



Gambar II.4. Elemen-Elemen Diagram *Use Case*
(Sumber : Hamim Tohari ; 2014)

1. Sistem, menyatakan batasan sistem dalam relasi dengan *actor-actor* yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Sistem digambarkan dengan segi empat yang membatasi semua *use case* dalam sistem terhadap pihak mana sistem akan berinteraksi.
2. *Actor* dapat berupa manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem. Digambarkan dengan *icon* yang mungkin bervariasi namun konsepnya sama.
3. *Use case*, mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan *user/actor*. Setiap *use case* mengekspresikan *goal* dari sistem yang harus dicapai. Diberi nama sesuai dengan *goal*-nya dan digambarkan dengan elips (dengan nama didalamnya).

4. *Association,* mengidentifikasikan interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu. Digambarkan dengan garis antara *actor* terhadap *use case* yang bersangkutan. Asosiasi bias berarah (garis dengan anak panah) jika komunikasi satu arah, namun umumnya terjadi kedua arah (tanpa anak panah) karena selalu diperlukan demikian.
5. *Stereotape,* memungkinkan perlluasan UML tanpa memodifikasinya. Berperan sebagai kualifier pada suatu elemen modal. Menyediakan informasi lebih banyak mengenai peranan dari elemen tanpa menyebutkan implementasinya.
6. *Dependency*, dependensi <<include>> mengidentifikasi hubungan antar dua *use case* diamana yang satu memanggil yang lain, digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis, dan arah mata panah sesuai dengan arah pemanggilan. Dependensi <<extend>> jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>> dan digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan (Hamim Tohari, 2014;51-54).

2. *Class* Diagram

Kelas (*Class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan perancangan berorientasi objek. Kelas menggambarkan keadaan (atribut/property) suatu *system,* sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut

(metode/fungsi). Dalam pemodelan statis dari sebuah sistem, diagram kelas biasanya digunakan untuk memodelkan salah satu dari tiga hal berikut :

1. Perbendaharaan dari sistem.
2. Kolaborasi.
3. Skema basis data *logical.*

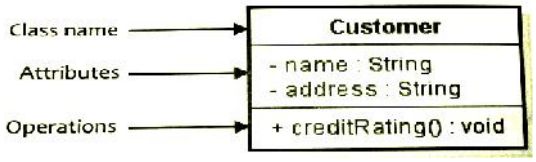
Kelas memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metode atau operasi

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected,* hanya dapat dipa nggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public,* dapat dipanggil oleh siapa saja.

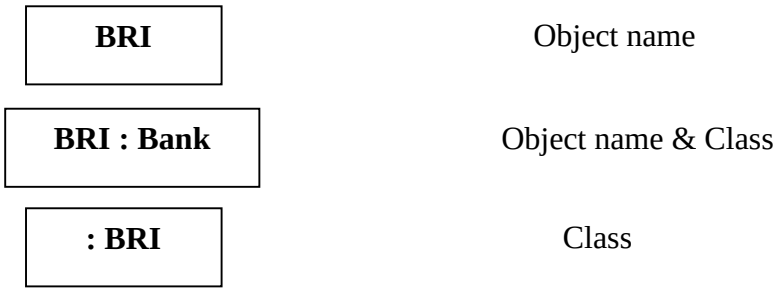
Kelas dapat berupa implementasi dari sebuah *interface,* yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstasiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metode pada saat *run-time* (Hamim Tohari, 2014;83-84). Adapun gambar *Class* Diagram dapat dilihat pada gambar II.5 dibawah ini :



Gambar II.5. Contoh Kelas
(Sumber : Hamim Tohari ; 2014)

3. *Sequence* Diagram

Sequence Diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi system. Dalam UML, objek pada diagram *sequence* digambarkan dengan segi empat, yang berisi nama dari objekyang digarisbawahi. Terdapat 3 cara untuk menamai objek yaitu, nama objek, nama objek dan *class* serta nama *class.* Misalnya pada gambar II.6 sebagai berikut :



Gambar II.6. Penamaan Objek pada *Sequence Diagram*

(Sumber : Hamim Tohari ; 2014)

Pada *diagram sequence*, setiap objek hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar objek digambarkan dengan anak panah dari objek yang mengirimkan pesan ke objek yang menerima pesan (Hamim Tohari, 2014;101).

4. *Activity* Diagram

Activity diagram memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup

menguntungkan untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behaviour* atau menggambarkan interaksi antara beberapa *use case*.

a. Elemen-Elemen *Activity* Diagram

1. Status *Start* (mulai) dan *end* (akhir).
2. Aktivitas yang mempresentasikan sebuah langkah dalam *workflow*.
3. *Transition* menunjukan terjadinya perubahan status aktivitas (*transition show what State follows another*).
4. Keputusan yang menunjukkan *alternative* dalam *workflow*.
5. *Synchronization bars* yang menunjukkan *subflow parallel*. *Synchronization bars* dapat digunakan untuk menunjukkan *concurrent threads* pada *workflow* proses bisnis.
6. *Swimlanes* yang mempresentasikan *role* bisnis yang bertanggung jawab pada aktivitas yang berjalan (Hamim Tohari, 2014;114-115).

II.10. MySQL

MySQL merupakan salah satu database yang banyak digunakan oleh para pengguna komputer. MySQL menggunakan bahasa SQL (*structured query language*) untuk berinteraksi dengan database. Setiap database menggunakan bahasa SQL dalam pengoperasiannya. Pada bahasa SQL, perintah yang digunakan bersifat *case-insensitive.* Sebaiknya anda membiasakan menggunakan huruf kapital untuk perintah SQL. Kebiasaan ini untuk membedakan nama database, nama tabel, atau nama kolom (Eko Priyo Utomo, 2014;58-59).

MySQL merupakan sistem basis data relasional dimana item data diorganisasikan dalam bentuk tabel. Untuk menciptakan sebuah tabel, sebuah *database* harus dibuat terlebih dahulu (Riyanto, 2014;63).