BAB II

LANDASAN TEORI

II.1. Sistem

Menurut (Asbon Hendra :2012 : 157) Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertetu.

II.1.1. Karakteristik Sistem

Menurut (Asbon Hendra :2012 : 158-160) ada beberapa karakteristik sistem adalah sebagi berikut :

1. Komponen (*Component)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary)*

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luar.

3. Lingkungan Luar Sistem (*Environment)*

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

4. Penghubungan Sistem (*Interface)*

   Merupakan media penghubung antara satu subsistem dengan subsistem yang laninya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu kesubsistem yang lain.

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.

6. Keluaran Sistem (*Output)*

   Merupakan hasil dari energi yang diolah oleh sistem meliputi *output* yang berguna.

7. Pengolah Sistem (*Process)*

   Merupakan bagian yang memproses masukan untuk menjadi keluaran yang di inginkan.

8. Tujuan Sistem (*Goal)*

   Suatu sistem akan dikatakan berhasil kalau pengoprasian sistem itu mengenai sasaran atau tujuannya.

II.1.2. Informasi

Menurut (Edhy Sutanta :2011: 13) Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat

mendatang.untuk meperoleh informasi, diperlukan data yang akan diolah dan unit pengolah.

II.1.3. Sistem Informasi

Menurut (Asbon Hendra :2012 : 168-169) Sistem informasi adalah sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/ mengambil, mengolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan.

Menurut (Edhy Sutanta :2011: 16) Sistem informasi dapat dipahami sebagai sekumpulan subsistem yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berinteraksi dan bekerja sama antara bagian satu dengan bagian lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menrima masukan(*Input*) berupa data-data, kemudian mengolahnya (*Processing*), dan menghasilkan keluaran (*Output*) berupa informasi sebagai dasar bagi pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun di masa mendatang, mendukung kegiatan operasional, manajerial, dan strategis organisasi, dengan memanfaatkan berbagai sumber daya yang ada dan tersedia bagi fungsi tersebut guna mencapai tujuan.

II.1.4. Sistem Informasi Akuntansi

Menurut (Anastasia Diana & Lilis Setiawati ; 2011 : 4), Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi-transaksi keuangan. Misalnya, salah satu input dari Sistem Informasi Akuntansi pada

sebuah toko baju, adalah transaksi penjualan, memproses transaksi dengan mencatat penjualan tersebut kedalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian secara periodik Sistem Informasi Akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi.

II.2. Laporan Laba Rugi

Menurut **(**Zakiyudin ; 2012 : 124), laporan Rugi/Laba adalah laporan yang disusun secara sistematis tentang pendapatan-pendapatan yang diperoleh perusahaan, serta beban-beban yang harus ditanggung oleh perusahaan dalam menjalankan kegiatan usahanya dalam satu periode tertentu.

Ada dua macam bentuk laporan Rugi/Laba, yaitu :

a. Bentuk Single Step : yaitu laporan yang hanya diberikan secara garis besarnya saja, atau disebut juga bentuk langsung, karena dalam bentuk ini semua pendapatan dikurangi dengan semua biaya sebagai sastu jumlah.
b. Bentuk Multiple Step : yaitu laporan yang diberikan secara terperinci, seluruh hasil dan biaya dimana terdapat sub bagian dan sub total, atau disebut juga bentuk bertahap. Bentuk laporan ini sering digunakan oleh perusahaan dagang dan perusahaan industri. Perusahaan dagang adalah perusahaan yang kegiatannya membeli barang untuk dijual kembali tanpa merubah bentuk atau mengolahnya lagi, dengan tujuan untuk memperoleh keuntungan atau laba. Dari pengertian tersebut, maka pendapatan utama perusahaan dagang berupa

hasil penjualan *(sales)* barang dagangan tersebut. Penjualan terjadi pada saat penyerahaan barang dagangan yang dijual sehingga mengurangi persediaan barang dagangan *(merchandise invetory)*. Untuk usaha tersebut, perusahaan mengeluarkan beban-beban operasional *(operational expense).*

Sehingga laba/rugi perusahaan dagang diperoleh dari selisih antara hasil penjualan bersih dengan harga pokok barang yang dijual, setelah dikurangi beban-beban operasional.

Berikut ini adalah rumus perhitungan laba/rugi bentuk multiple step

Penjualan bersih - Harga Pokok Penjualan = Laba Kotor Penjualan
Laba Kotor Penjualan - Beban Operasional = Laba Bersih

(Sumber : Ais Zakiyudin ; 2012 : 41)

II.3. PHP

Menurut Anhar (2010 : 3), PHP singkatan dari *hypertext preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

II.4. Database MySQL

Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Database MySQL (*My Structure Query Language*) adalah salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle, MS SQL, Postgre SQL*, dan lainya. Mysql berfungsi untuk mengolah database menggunakan bahasa SQL. Mysql bersifat *open source* sehingga kita bisa menggunakan secara gratis. Pemrograman PHP juga sangat mendukung/support dengan database MySQL (Anhar ; 2010 : 45).

Struktur file yang menyusun sebuah database adalah *Data Record* dan *Field* (Anhar ; 2010 : 46).

a. Data adalah satu satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu *file database*.
b. *RECORD* adalah data yang isinya merupakan satu kesatuan seperti NamaUser dan *Password.* Setiap keterangan yang mencakup NamaUser dan *Password* dinamakan satu *record*. Setiap *record* diberi nomor urut yang disebut nomor record *(Record Number).*
c. *FIELD* adalah sub bagian dari *record*. Dari contoh isi *record* di atas, maka terdiri dari 2 *field*, yaitu: *field* NamaUser dan *Password*

Selain berisi data, *database* juga berisi *metadata*. Metadata adalah data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh, Anda dapat memperoleh informasi tentang nama-nama kolom dan tipe yang ditampilkan tersebut disebut *metadata.*

II.4.1. Pemodelan Data

Menurut **(**Yudi Priyadi ; 2014 : 10) Terdapat beberapa penjelasan mengenai pemodelan basis data. Suatu basis data dapatdigunakan secara bebas untuk menggambarkan dan memberikan deskripsi mengenai kumpulan informasi uang tersimpan dalam *data storage* komputer. Secara sederhana, defenisi untuk model basis data adalah sekumpulan notasi atau simbol untuk menggambarkan data dan relasinya, berdasarkan suatu konsep dan aturan tertentu suatu pemodelan.

II.4.2. Notasi Diagram E-R

Menurut **(**Yudi Priyadi ; 2014 : 20) Pemodelan basis data dengan menggunakan diagram relasi antar entitas, dapat dilakukan dengan menggunakan suatu pemodelan basis datayang bernama Diagram *Entity-Relational* (selanjutnya disingkat Diagram E-R). Pada Gambar II.2, terdapat suatu simbol/notasi dasar yang digunakan pada Diagram E-R,yaitu entitas,relasi, atribut, dan garis penghubung.

1. Entitas

   Merupakan notasi untuk mewakili suatu objek dengan karakteristik sama, yang dilengkapi oleh atribut, sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan objek lainya. Pada umumnya, objek dapat berupa benda, pekerjaan, tempat dan orang.

2. Atribut

   Merupakan notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai key yang bersifat unik, yaitu *Primary Key*

atau *Foreign Key*. Selain itu, atribut juga dapat sebagai atributdeskriptif saja, yaitu sebagai pelengkap deskripsi suatu entitas dan relasi.

3. Relasi

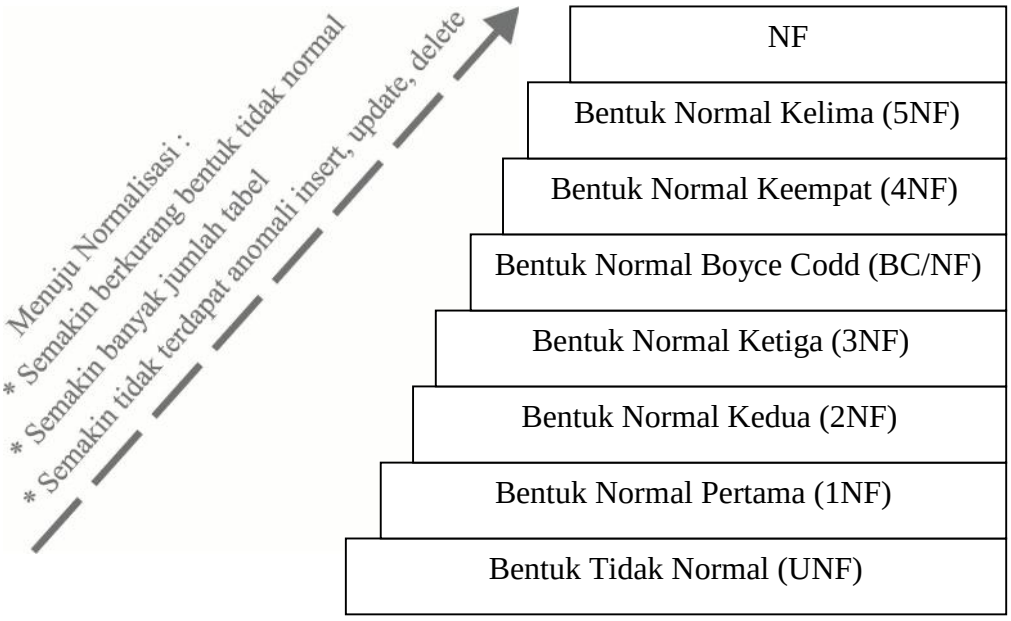
   Merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.

4. Garis penghubung

   Merupakan notasi untuk merangkaikan keterkaitan antar notasi yang digunakan dalam Diagram E-R, yaitu entitas, relasi dan atribut.

II.4.3. Normalisasi

Menurut **(**Yudi Priyadi ; 2014 : 67) Normalisasi merupakan proses sistematis yang dilakukan pada struktur tabel basis data menjadi struktur tabel yang memiliki integritas data, sehingga tidak memiliki data anomali pada saat melakukan *insert, delete,* dan *update.* Pada Gambar II.1, tahapan proses sistematis yang dilakukan mulai dari bentuk tidak normal menjadi bentuk normal memiliki suatu syarat yang harus dipenuhi pada saat menuju suatu bentuk yang lebih baik (*well structured relation*).



Gambar II.1 : Tahapan Proses Bentuk Normalisasi

(Sumber : Yudi Priyadi ; 2014 : 67)

Setiap syarat dalam tahapan suatu bentuk normal memiliki keterkaitan, hal ini disebabkan karena pada setiap bentuk normal mengalami penyempurnaan untuk bentuk normal selanjutnya. Bentuk tidak normal akan semakin berkurang, setelah melalui tahapan berubahan bentuk normalisasi, sehingga berdampak pada jumlah tabel yang semakin banyak, tetapi menuju perbaikan ke dalam bentuk *well structured relation.* Hal ini terjadi akibat dari pengelompokan data suatu tabel agar memiliki ketergantungan secara fungsional.

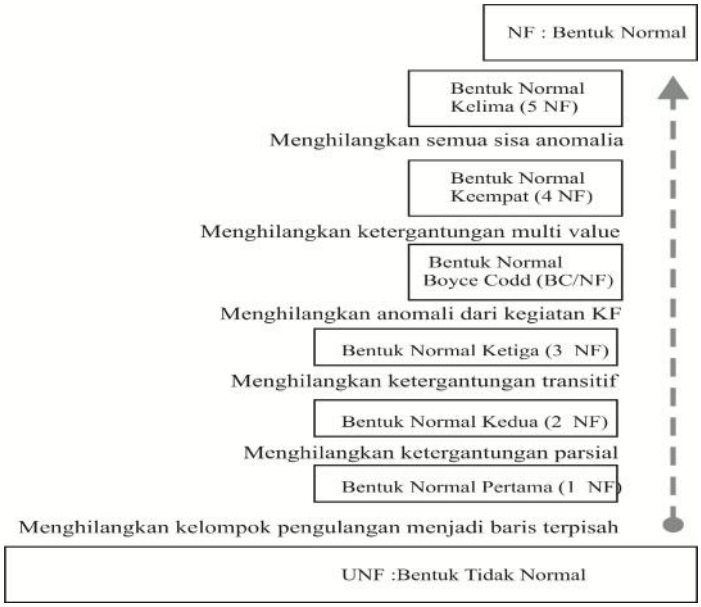
II.4.4. Aturan Proses Normalisasi

Menurut **(**Yudi Priyadi ; 2014 : 68) Secara sederhana, kegiatan normalisasi adalah melakukan dekomposisi atau penguraian tabel beserta datanya, menjadi tabel yang normal menurut konsep RDBMS. Merujuk pada gambar II.2, dekomposisi diawali dengan melakukan analisis pada suatu tabel atau beberapa

contoh formulir yang sudah memiliki data lengkap dalam basis data, tetapi masih dalam bentuk yang tidak normal (UNF). Oleh karena itu agar dapat memenuhi syarat bentuk normal pertama (1NF), pada setiap barisnya diisikan suatu *value* dengan kelompok data yang sama, berdasarkan suatu atribut key. Dengan demikian, kelompok pengulangan dalam suatu baris dapat dihilangkan, karena sudah tidak terdapat *value* yang kosong untuk setiap *field* dan *record*nya

Setelah memenuhi syarat bentuk normal pertama (1NF), proses berikutnyaadalah menghilangkan ketergantungan secara parsial, yaitu dengan cara melakukan dekomposisi tabel menjadi beberapa kelompok tabel berdasarkan field yang memiliki status sebagai key. Hal ini dapat dilakukan oleh salah satu field saja, dengan tetap tidak mengubah arti relasi dan ketergantungannya. Oleh sebab itu, disebut ketergantungan fungsional sebagian (*partiallly functional*), sehingga syarat bentuk normal kedua (2NF) sudah tercapai.

Bentuk normal kedua (2NF) merupakan syarat yang harus dimiliki untuk menuju bentuk normal ketiga (3NF). Pada proses ini, dilakukan dengan menghilangkan ketergantungan secara transitif, yaitu suatu konsep untuk tabel dari hasil relasi yang didalamnya terdapat ketergantungan secara tidak langsung pada beberapa atributnya. Pada umumnya proses normalisasi sudah dapat tercapai pada bentuk normal ketiga (3NF), yaitu dengan menghasilkan tabel yang tidak mengalami anomali basis data pada saat proses *insert, delete,* dan *update.*



Gambar II.2 : Tahapan Aturan Proses Normalisasi

(Sumber : Yudi Priyadi ; 2014 : 69)

II.5. Unified Modeling Language (UML)

Menurut (Rosa A.S & M. Shalahuddin ; 2011 : 118) Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakanteknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metode berorientasi objek.

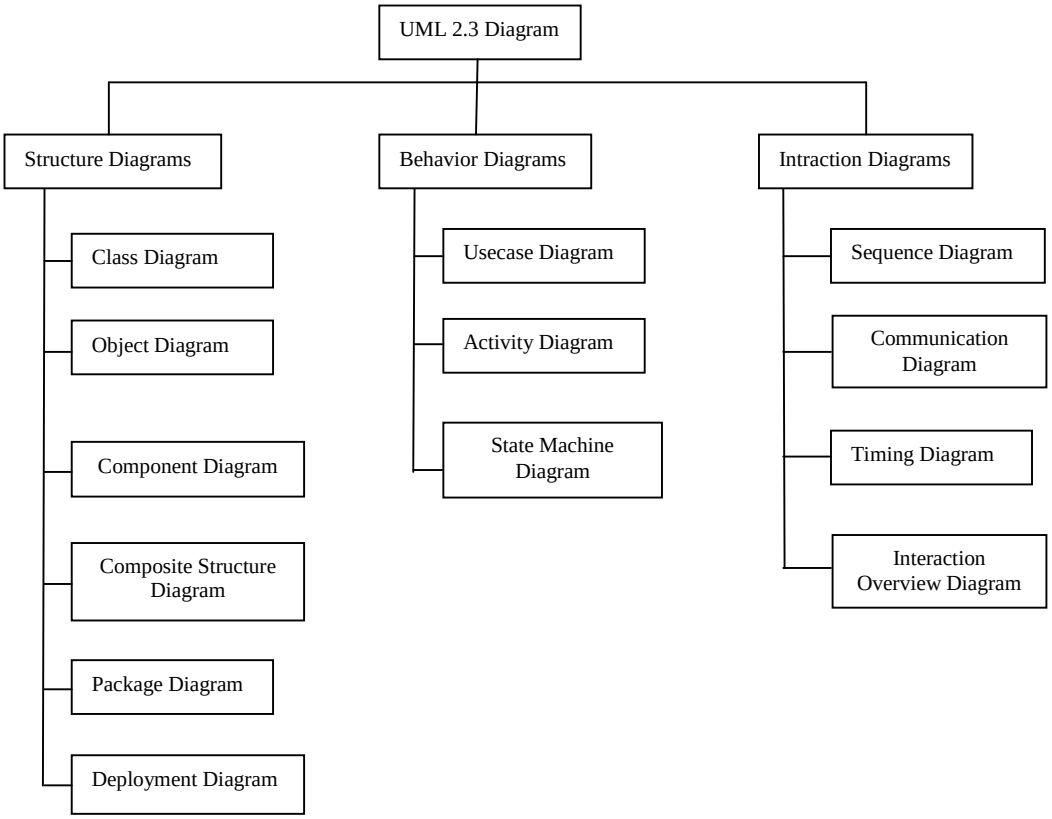
Menurut (Prabowo Pudjo Widodo & Herlawati ; 2011 : 6) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

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

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timming diagram*) dan lainya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini.

II.5.1 Diagram-Diagram UML

Menurut (Rosa A.S & M. Shalahuddin ; 2011 : 120) Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar II.3 di bawah ini



Gambar II.3 : Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untukmenggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

A. *Class Diagram*

Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1) Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2) Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut table II.1 menerangkan simbol-simbol pada diagram kelas :

Table II.1 : Diagram Kelas

Simbol	Deskripsi
Kelas Nama_kelas +atribut +operasi()	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	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / *aggregation*	Semua bagian (*whole part*)

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 124)

B. *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefenisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefenisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefenisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai.

Berikut adalah Table II.2 menerangkan simbol-simbol diagram objek

Table II.2 : Diagram Paket

Simbol	Deskripsi
Objek Nama_objek : nama_kelas Atribut = nilai	Objek dari kelas yang berjalansaat sistem dijalankan
Link ———————	Relasi antar objek

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 124)

C. *Component Diagram*

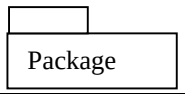
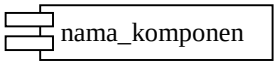
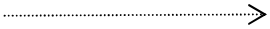
Diagram komponen atau component diagram dibuatuntuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

1) Komponen *user interface* yang menangani tampilan
2) Komponen *bussiness procesiing* yang menangani fungsi-fungsi proses bisnis
3) Komponen data yang menangani manipulasi data

4) Komponen *security* yang menangani keamanan sistem

Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan, berikut Table II.3 yang menerangkan simbol-simbol yang ada pada diagram komponen

Table II.3 : Diagram Komponen

Simbol	Deskripsi
Package 	*Package* merupakan sebuah bungkusan dari satu atau lebih komponen
Komponen 	Komponen Sistem
Kebergantungan / *dependency* 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antar muka / *interface*  nama_interface	Sama dengan konsep *interface*pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
Link	Relasi antar komponen

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 126)

D. *Use Case Diagram*

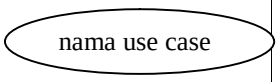
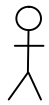
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour)*sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar,*use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama

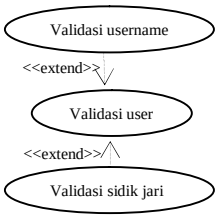
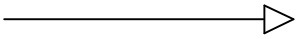
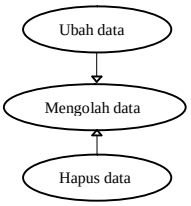
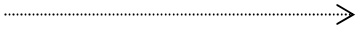
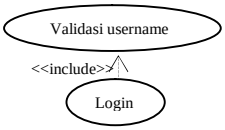
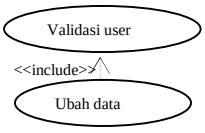
didefenisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefenisian apa yang disebut aktor dan *use case.*

1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2) *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut table II.4 menerangkan simbol-simbol pada diagram *use case*

Table II.4 : Diagram Use case

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor / *actor*  nama aktor	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
Asosiasi / *association*	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case*, atau *usecase* memiliki interasi dengan aktor
Ekstensi / *extend* 	Relasi usecase tambahan ke subuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanppa *use case* tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan misal

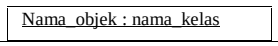
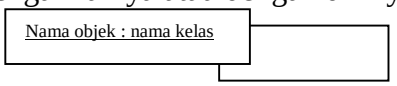
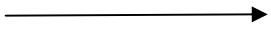
	 arah panah mengarah pada *use case* yang ditambahkan
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainya misalnya :  Arah panah mengarah pada use case yang menjadi generalisasinya (umum)
Menggunakan / *include* / *uses* <<include>>  <<uses>> 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini Ada 2 sudut pandang yang cukup besar mengenai include di usecase 1. include berarti use case yang ditambahkan akan selalu dipanggil saat use case dijalankan misal pada kasus berikut :  2. include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah di jalankan sebelum use case tambahan di jalankan, misal pada kasus berikut :  Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 131)

E. Communication Diagram

Diagram komunikasi mengelompokkan message pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi / metode yang di jalankan antara objek yang satu dengan objek lainnya secara keseluruhan, oleh karna itu dapat di ambil dari jalanya interaksi pada semua diagram sekuen. Berikut adalah Table II.5 yang menerangkan simbol-simbol yang ada pada diagram komunikasi :

Table II.5 : Diagram Komunikasi

Simbol	Deskripsi
Objek 	Objek yang melakukan interaksi pesan
Link	Relasi antar objek yang menghubungkan objek satu dengan lainya atau dengan dirinya sendiri 
Arah pesan / stimulus 	Arah pesan yang terjadi, jika pada suatu link ada dua arah pesan yang berbeda, maka arah juga deigambarkan dua arah pada dua sisi link 

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 140)

F. *Activity Diagram*

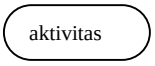
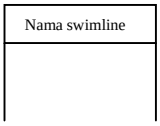
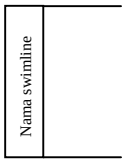
Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefenisikan hal-hal berikut :

1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sitemyang didefenisikan

2) Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitasdianggap memiliki sebuah rancangan antarmuka tampilan
3) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefenisikan kasus ujinya.

Berikut adalah Table II.6 yang menggambarkan simbol-simbol yang ada pada diagram aktivitas :

Table II.6 : Diagram Aktivitas

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / decesion 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

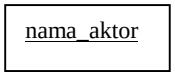
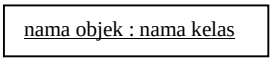
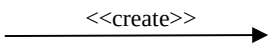
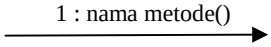
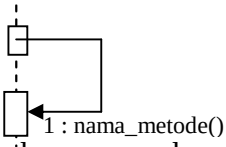
(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 134)

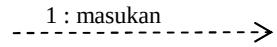
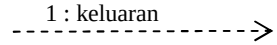
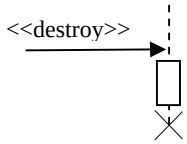
G. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan messasge yang dikirimkan dan diterima antar objek. Banyaknya diagram objek yang digambarkan adalah

sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefenisikan interaksi jalanya pesan sudah dicakup dapa diagram sekuen sehingga semakin banyak use case yang didefenisikan maka diagram sekuen yang harus dibuat juga semakin banyak.Berikut adalah Table II.7 yang menerangkan simbol-sombol yang ada pada diagram sekuen :

Table II.7 : Diagram Sequence

Simbol	Deskripsi
Aktor  atau  tampa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya di nyatakan menggunakan kata benda di awali frase nama aktor
Garis hidup / lifeline	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berintaraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create 	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tope call 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri  Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi

Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainya, arah panah mengarah pada objek yang dikirimi
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasiatau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 138)