

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Dasar Sistem Informasi.

II.1.1. Pengertian Sistem

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

1. Komponen sistem

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

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya

3. Lingkaran Luar Sistem

Lingkungan Luar sistem dari suatu elemen adalah apapun dari luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan Luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sistem subsistem ke subsistem lainnya.

5. Masukan Sistem

Adalah energi yang dimasukkan kedalam sistem. masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang dioalah dan diklafisikasikan menjadi keluaran berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mmepunyai sasaran, maka operasi sistem tidak ada gunanya.
(Sulindawati, 2010 : 375)

II.1.2. Pengertian Sistem Informasi

informasi adalah data yang telah diklafifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya.

Sistem informasi merupakan penerapan sistem didalam organisasi untuk mendukung informasi yang dibutuhkan oleh semua tingkat manajemen. telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Darimana informasi tersebut bisa didapatkan? Informasi diperoleh dari sistem informasi, sistem informasi didefinisikan sebagai berikut:

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri, 2004 : 36).

II.1.3. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen – komponen yang disebut dengan istilah blok bangunan (*building block*) yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. sebagai suatu sistem, keenam blok tersebut saling berinteraksi satu dengan lain yang membentuk satu kesatuan untuk mencapai sasaran.

a. Blok masukan (*input block*)

Input mewakili data yang masuk kedalam sistem informasi. yang dimaksud input disini termasuk metode dan media menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.

b. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan menapulasi data iput dan data yang tersimpan dibasis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok keluaran(*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.

d. Blok teknologi (*technology block*)

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem secara keseluruhan.

e. Blok basis data (*database block*)

Basis data (database) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk manipulasinya.

f. Blok Kendali (*control block*)

Beberapa Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal- hal yang dapat merusak sistem dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi. (tata sutabri, 2004 : 37).

II.2. Pengertian Sistem Informasi Akuntansi

Sistem Informasi Akuntansi merupakan kumpulan sumber daya, seperti manusia dan peralatan, yang dirancang untuk mengubah data keuangan dan data lainnya kedalam informasi. Informasi tersebut dikomunikasikan kepada para pembuat keputusan. Sistem informasi akuntansi melakukan hal tersebut baik dengan sistem manual atau melalui sistem terkomputerisasi. (George, 2006 : 3).

II.3. Pengertian pengendalian internal

Pengendalian internal merupakan bagian yang sangat penting dalam suatu perusahaan karena merupakan alat untuk mengendalikan aktivitas-aktivitas yang dilakukan pada akhirnya dapat mencapai tujuan yang telah ditetapkan.

II.3.1 Unsur unsur pengendalian internal

Menurut Ikatan Akuntan Indonesia dalam standar Profesional Akuntan Publik (2001:319.8.8-11)

1. Lingkungan pengendalian menetapkan corak suatu organisasi dan mempengaruhi kesadaran orang-orangnya. Lingkungan pengendalian merupakan dasar untuk semua komponen pengendalian internal yang lain, menyediakan disiplin dan struktur.
2. Penaksiran Resiko

Penaksiran resiko entitas untuk tujuan pelaporan keuangan merupakan identifikasi, analisis dan manajemen terhadap resiko yang relevan dengan

penyusunan laporan keuangan yang wajar sesuai dengan prinsip akuntansi yang berlaku umum di Indonesia.

3. Aktivitas Pengendalian

Aktivitas pengendalian adalah kebijakan dan prosedur yang membantu arahan manajemen dilaksanakan aktivitas tersebut membantu memastikan bahwa tindakan yang diperlukan untuk menanggulangi risiko dalam pencapaian tujuan entitas.

4. Informasi Dan Komunikasi

Sistem informasi yang relevan dengan tujuan pelaporan keuangan yang meliputi sistem Akuntansi, terdiri dari metode dan catatan yang dibangun untuk mencatat, mengolah, meringkas dan melaporkan transaksi entitas dan untuk memelihara akuntabilitas bagi aktiva, utang dan ekuitas yang bersangkutan, Komunikasi mencakup pemberian pemahaman atas peran dan tanggung jawab individual berkaitan dengan pengendalian intern terhadap pelaporan keuangan

5. Pemantauan

Adalah proses penentuan kualitas kinerja pengendalian intern sepanjang waktu. Pemantauan ini mencakup penentuan desain dan operasi pengendalian internal tepat waktu. (Sri Winarni, 2013 :51)

II.4.VB.NET

II.4.1. Sejarah VB.NET

Pada zaman dahulu ada sebuah bahasa pemrograman yang diberi nama basic (Beginner's All-purpose Symbolic Instruction Code). Sesuai dengan namanya, Basic ditunjukkan sebagai bahasa yang paling sederhana bagi mereka yang tidak terlalu familiar dengan dunia pemrograman.

Pada zaman 1991 Microsoft mengeluarkan Visual Basic, pengembangan dari Basic yang merubah dari sisi pembuatan antarmukanya. Visual Basic sampai sekarang masih menjadi salah satu bahasa pemrograman terpopuler di dunia.

Pada akhir tahun 1999, Teknologi.NET diumumkan. Microsoft memosisikan teknologi tersebut sebagai platform untuk membangun XML Web Services. XML Web Services memungkinkan aplikasi tipe apa pun dapat berjalan pada system komputer dengan tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apa pun melalui internet.



Gambar II.4.1 . Web Servis

Sumber :Priyanto Hidayatullah (2014:4)

II.4.2. Pengertian VB.NET

Visual Basic .Net adalah Visual Basic yang direkayasa kembali untuk digunakan pada platform .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal .NET Framework. (Priyanto Hidayatullah, 2014 : 5)

II.5. Microsoft SQL Server

Database Management System (DBMS) adalah aplikasi yang dipakai untuk mengelola basis data. DBMS biasanya menawarkan beberapa kemampuan yang terintegrasi seperti :

1. Membuat, menghapus, menambah dan memodifikasi basis data
2. Pada beberapa DBMS pengelolaannya berbasis *Windows* (berbentuk jendela-jendela) sehingga lebih mudah digunakan
3. Tidak semua orang bisa mengakses basis data yang ada sehingga memberikan keamanan bagi data
4. Kemampuan berkomunikasi dengan program aplikasi yang lain. Misalnya dimungkinkan untuk mengakses basis data SQL Server menggunakan aplikasi yang dibuat menggunakan VB .NET
5. Kemampuan pengaksesan melalui komunikasi antarkomputer (client server)

Microsoft SQL server adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi basis data. Contoh DBMS

lainnya adalah: MySQL (*freeware*), MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro, dsb.(Priyanto Hidayatullah, 2014 ;175)

II.5.1. Manajemen Basis Data

Manajemen basis data adalah seperangkat program komputer yang didesain untuk mengatur sebuah basis data sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi data.

Didalam manajemen basis data akan dibahas beberapa hal, yaitu :

1. Informasi

Hasil dari pengolahan data sehingga menjadi sesuatu yang berarti. Misalnya, grafik produksi komputer perbulan mengalami kenaikan sebanyak 50%, hal itu merupakan data yang sudah diolah menjadi sebuah informasi.\

2. Basis Data

Basis data merupakan sekumpulan data yang disusun secara logis dan dikendalikan secara sentral basis data memiliki bagian bagian penting, misalnya tabel yang digunakan atau kolom dan record atau data perbaris.

3. Tabel

Tabel merupakan perpaduan antara baris dan kolom yang digunakan untuk menyimpan data data .

4. Field

Merupakan elemen dari tabel yang berisikan informasi tertentu yang spesifik seperti nama seseorang atau nama kota.

5. *Record*

Record atau disebut juga basis merupakan sekumpulan data yang berkaitan dengan sebuah subjek tertentu dan *record* merupakan kumpulan dari *fields*.

6. *Null*

Null berarti tidak memiliki nilai sama sekali, untuk diingat dalam terminologi komputer angka nol (0) berarti masih memiliki nilai.

7. *Relationship*

Hubungan yang dibuat antara tabel-tabel yang memiliki persamaan data dari dua *field* diantara tabel tersebut. (Ema utami,2012 : 66)

II.5.2. Perancangan Basis Data

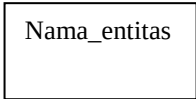
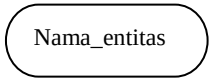
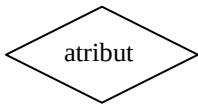
Proses perancangan ini bersifat konseptual.tujuan perancangan basis data adalah mendapatkan skema basis data yang meminimalisasi terjadinya redundansi dan duplikasi dan serta menjaga integritas data.keanyakan metode perancangan berbasis pada model basis data relasional. Pada basis data relasional, data diatur melalui pembuatan tabel – tabel dan terdapat keterkaitan antara tabel yang satu dengan yang lainnya (relasi).(Priyanto Hidayatullah, 2014:139)

II.5.3. ERD (Entity Relationship Diagram)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika.ERD digunakan untuk pemodelan basis data relasional.Sehingga penyimpana basis data tidak perlu menggunakan ERD. ERD

memiliki beberapa aliran notasi seperti notasi Chen (Dikembangkan oleh Peter Chen), Barker (Dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's foot, dan beberapa notasi lain, Namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen. Simbol-simbol yang digunakan ERD :

Tabel 2.1 Simbol pada ERD

No.	Simbol	Keterangan
1.	Entitas/entity 	Entitas Merupakan data inti yang akan disimpan; bakal tabel pada basis data: benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
2.		Field atau kolom data yang butuh disimpan dalam suatu entitas
3.		Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
4.	Asosiasi / association 	Penghubung antara relasi dan entitas dimana kedua ujungnya multiple kemungkinan jumlah pemakaian jumlah maksimum keterhubungan antar entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke n atau sering disebut dengan one to many menghubungkan entitas A dan entitas B

(Priyanto Hidayatullah, 2013:50)

II.5.4. Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel yang menunjukkan entitas sekaligus relasinya. Melalui normalisasi, kita akan mendesain basis data relasional menjadi suatu set data yang memenuhi kriteria berikut ini :

1. Memuat semua data penting yang dapat disediakan oleh basis data.
2. Memiliki *redundancy* data yang sesedikit mungkin.
3. Akomodasi multi *value* untuk tipe data yang diperlukan.
4. Mengizinkan *update* data yang efisien dalam basis data.
5. Terhindar dari bahaya kehilangan data yang tidak dikenal.

Tahapan-Tahapan normalisasi :

a. Bentuk Normalisasi Pertama (1NF)

Bentuk Normal pertama memiliki ciri :Data berbentuk flat file (file datar).record disusun sesuai kedatangan.masih mungkin terjadi penyimpangan data (anomaly data). data dapat berupa *insert anomaly,delete anomaly, update anomaly dan redundace anomaly*.

b. Bentuk Normalisasi Kedua (2NF)

Bentuk Normal kedua memiliki cirri: Tidak terjadi anomali data.setiap field/atribut bukan kunci harus tergantung fungsi (*Funcional Dependency*) masih mungkin terjadi *transitive depency* (*field* bukan kunci tergantung pada *field* bukan kunci suatu *table*).

c. Bentuk Normalisasi Ketiga (3NF)

suatu relation sudah berada pada 3NF bila sudah berada dalam 2NF dan

setiap atribut yang bukan key tidak dependent terhadap atribut lain (tidak transitif) kecuali terhadap primary key.(Anggit Dwi Hartanto,2012 :40)

II.6. UML

II.6.1 Pengenalan UML

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang diberbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunanperangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *state Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).(Priyanto Hidayatullah,2013 ;137)

II.6.2. Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama simula-67 yang dikembangkan pada tahun 1967. Bahasa

pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya.

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman smaltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian pada saat itu. Sekitar lima tahun setelah smaltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shlaer dan Stephen Mellor (Shlaer-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Boch (Boch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premierlani (Rumbaugh-Blaha-Premierlani-Eddy-Lorensen, 1991), dan masih banyak lagi.

Buku terkenal yang juga berkembang selanjutnya adalah karangan Ivar Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang fokus pada *use case* dan proses pengembangan. Sekitar lima tahun kemudian muncul buku yang membahas mengenai metodologi berorientasi objek yang diikuti dengan buku-buku yang lainnya. Di dalamnya juga membahas mengenai konsep, definisi, notasi, terminologi, dan proses mengenai metodologi berorientasi objek.

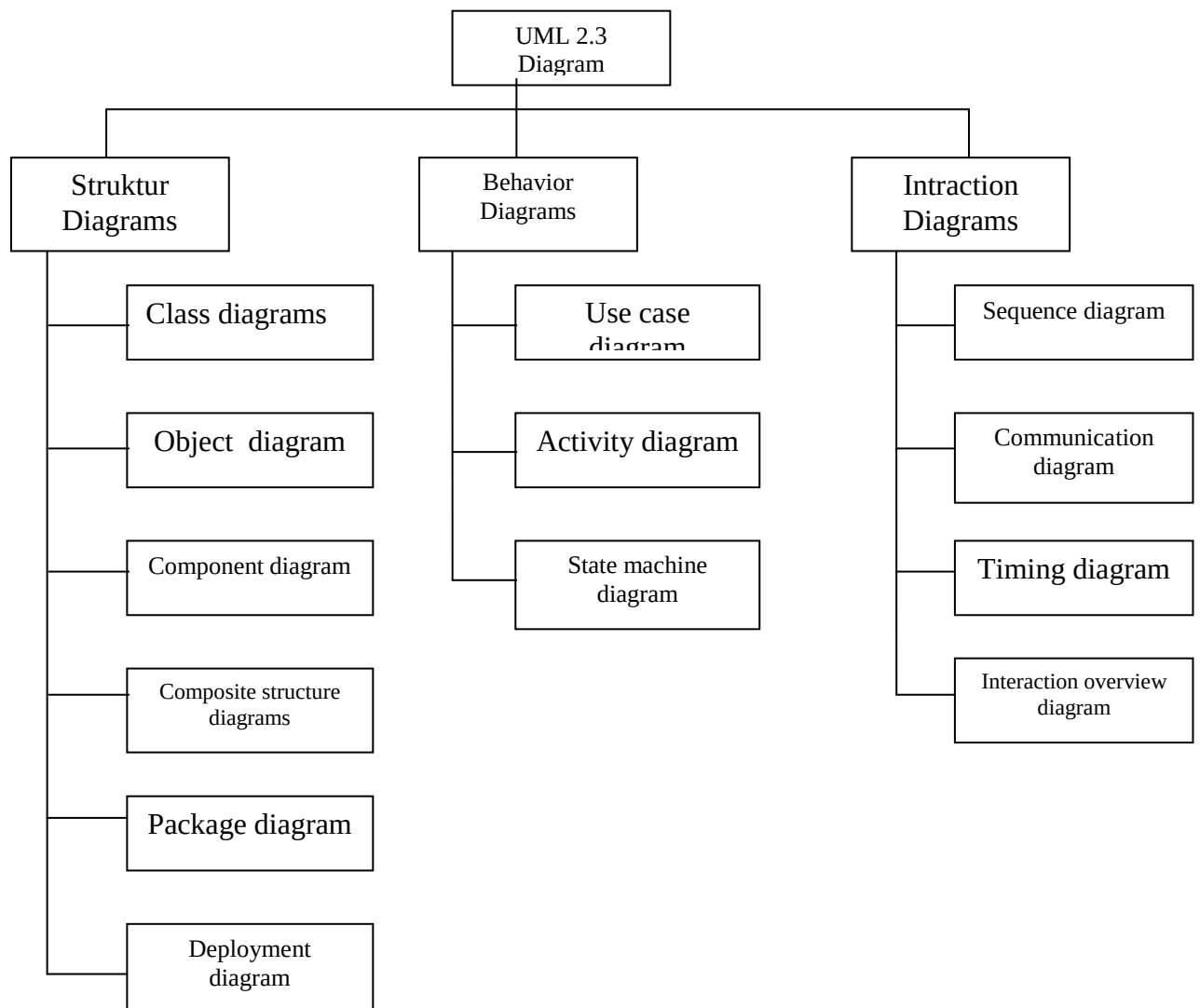
Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua

orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modelling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan Unified Modeling Language (UML).

Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya. Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, UML Infrastructure, UML Superstructure, dan Object Constraint Language(OCL).(Rosa A.S, 2014 138)

II.6.3. Diagram - Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah.



(Priyanto Hidayatullah,2013 ;137)

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 system.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. .(Rosa A.S, 2014 ;140)

1. 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 dengan 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

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas Main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem (*View*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian *use case*(*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. .(Rosa A.S, 2014 ;142)

2. Usecase Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) 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 didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal hal utama pada *use case* yaitu pendefinisian 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.

2. Activity Diagram.

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau sebuah proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak. (Priyanto Hasibullah, 2014 ;161)

3.Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen itu juga membutuhkan untuk melihat skenario yang ada pada use case.

Banyaknya diagramsekuen harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen harus dibuat juga semakin banyak.(Rosa A.S, 2014 ;167)