

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Konsep Dasar Sistem

Menurut Sutabri (2012:16) “Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung satu sama lain dan terpadu”.

Menurut Hartono (2013:9) “Sistem adalah suatu himpunan dari berbagai bagian atau elemen yang saling berhubungan secara terorganisasi berdasar fungsi – fungsinya menjadi satu kesatuan”.

Menurut Taufiq (2012:2) “Sistem adalah kumpulan dari sub-sub sistem abstrak maupun fisik yang saling terintegrasi dan berkolaborasi untuk mencapai suatu tujuan tertentu”.

Berdasarkan definisi diatas, maka dapat disimpulkan sistem adalah sekelompok unsur yang saling terhubung satu sama lain yang berfungsi bersama-sama untuk mencapai suatu tujuan tertentu.

2.1.2 Karakteristik Sistem

Menurut Sutabri (2012:20) menyimpulkan bahwa:

Model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari jumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen - komponen sistem tersebut dapat berupa suatu bentuk sub sistem. Setiap subsistem memiliki sifat dari sistem yang melanjutkan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan *Supra System*.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah - pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut, disebut dengan lingkungan luar sistem. Dimana lingkungan luar ini dapat bersifat menguntungkan merupakan energi bagi sistem tersebut, dengan demikian lingkungan luar tersebut harus dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem tersebut.

4. Penghubung Sistem (*Interface*)

Sebagai media yang menghubungkan sistem dengan subsistem yang lainnya disebut dengan penghubung sistem atau *interface* melalui penghubung ini memungkinkan

sumber-sumber daya mengalir dari suatu subsistem akan menjadi masukan untuk subsistem yang lainnya dengan melalui penghubung tersebut.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan kedalam sistem tersebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan signal masukan (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna, keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objektive*)

Suatu sistem memiliki tujuan dan sasaran yang pasti akan bersifat determinan kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila menggunakan sasaran atau tujuan yang direncanakan.

2.1.3 Klasifikasi Sistem

Menurut Sutabri (2012:15) menyimpulkan bahwa:

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi didalam sistem tersebut. Oleh karena itu sistem dapat dikasifikasikan dari beberapa sudut pandang, seperti contoh sistem yang bersifat abstrak, sistem alamiah, sistem yang bersifat *deterministic*, dan sistem yang bersifat terbuka dan tertutup.

2.1.4 Definisi Informasi

Menurut Sutarman (2012:14) “Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.

Menurut Darmawan (2013:2) “Informasi adalah hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi, hasil dari pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut”.

2.1.5 Pengertian Sistem Informasi

Menurut Hutahaean (2015:13) ”Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Sistem Informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*Building Block*), yaitu :

1. Blok Masukan (*input block*)

Input mewakili data yang termasuk ke dalam sistem informasi. *Input* disini termasuk metode - metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah diinginkan.

3. Blok Keluaran (*output blok*)

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

4. Blok Teknologi (*technology block*)

Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan.

Teknologi terdiri dari unsur utama:

- a. Teknisi (*human ware* atau *brain ware*)
- b. Perangkat lunak (*software*)
- c. Perangkat keras (*hardware*)

5. Blok Kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi , misalnya bencana alam, api, temperature tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan dan menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

2.1.6 Definisi Inventaris

Menurut Gima.A (2013:173) “Inventaris adalah serangkaian kegiatan untuk melakukan pendataan, pencatatan, pelaporan hasil pendataan asset dan

mendokumentasikannya, baik aset berwujud maupun aset tidak berwujud pada suatu waktu tertentu. Inventarisasi aset dilakukan untuk mendapatkan data seluruh aset yang dimiliki, dikuasai sebuah organisasi, perusahaan, atau instansi pemerintah”.

Menurut Chabib dkk (2010:180) “Inventarisasi merupakan kegiatan atau tindakan untuk melakukan penghitungan, pengurusan, penyelenggaraan peraturan, pencatatan data dan pelaporan barang milik daerah dalam unit pemakaian”.

Inventaris adalah daftar memuat semua barang milik kantor yang dipakai untuk kegiatan mencatat dan menyusun barang dengan ketentuan dan tugas berlaku. Inventaris kantor sangatlah penting bagi kelangsungan sebuah perusahaan dan instansi. Gangguan pada salah satu atau beberapa perlengkapan pasti akan menghambat jalannya roda perekonomian perusahaan atau instansi tersebut.

2.1.7 WEB

Menurut Hidayat (2010:2) menyimpulkan bahwa:

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

Menurut Kustiyahningsih (2011:113) “*Web* adalah layanan yang didapat oleh pemakai komputer yang terhubung ke *internet*. *Browser* adalah perangkat lunak untuk mengakses halaman *web* seperti *Internet Explorer*, *Mozilla Firefox*, *Opera*, *Safari* dan lain-lain”.

Dapat disimpulkan *website* adalah sebuah tempat di *internet*, yang menyajikan informasi dengan berbagai macam format data seperti *text*, *image*, bahkan video dan

dapat diakses menggunakan berbagai aplikasi klien sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi.

2.1.8 PHP (*Hypertext Preprocessor*)

Menurut Sibero (2011:49) menyimpulkan bahwa:

PHP adalah pemograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemograman *Server side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya.

2.1.9 Hyper Text Markup Language (HTML)

Menurut Sibero (2011:19) menyimpulkan bahwa:

HTML (*Hyper Text Markup Language*) adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*, sebuah bahasa *markup* yang digunakan untuk membuat halaman *web*, menampilkan berbagai informasi di dalam sebuah *internet web browser* dan *formatting hypertext* yang sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan ke dalam format ASCII normal sehingga menjadi *home page* dengan perintah- perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan yang disebut SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standard yang digunakan secara luas untuk menampilkan halaman *web*, HTML saat ini merupakan standard *internet* yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

2.1.10 HTTP Server dan Web Browser

Menurut Sibero (2011:12) menyimpulkan bahwa:

Aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*. Sumber informasi *web* diidentifikasi dengan *Uniform Resource Identifier (URL)* yang dapat terdiri dari halaman *web*, video, gambar, ataupun konten lainnya. Sejarah *Web Browser* dimulai pada tahun 1991 saat Tim Berners-Lee membuat aplikasi *Web Browser* pertama pada komputer NeXT

dengan nama *World Wide Web Browser*, kemudian pada tahun 1993 NCSA (National Center Supercomputing Application) mengembangkan *Web Browser* grafis bernama NCSA Mosaic, yang kemudian dilanjutkan pada tahun 1994 merilis Netscape Navigator dan pada tahun 1998 berubah menjadi *Mozilla Firefox*.

2.1.11 MYSQL (*My Structure Query Language*)

Menurut Anhar (2010:45) “MySQL (*My Structure Query Language*) adalah salah satu database *management system* (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postagre SQL* dan lainnya. Mysql berfungsi untuk mengolah *database* menggunakan bahasa *SQL*. Mysql bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman php juga sangat mendukung atau *support* dengan *database mysql*”.

2.1.12 XAMPP

Menurut Puspitasari (2011:1) berpendapat bahwa:

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia *database server mysql* dan *support php programming*. xampp merupakan *software* yang mudah digunakan gratis dan mendukung instalasi di *linux* dan *windows*. Keuntungan lainnya adalah cuma menginstal 1 kali sudah tersedia *apache web server*, *mysql database server*, *php support* dan beberapa modul lainnya hanya bedanya kalau versi windows selalu dalam bentuk instalasi grafis dan yang linux dalam bentuk file terkompresi tar.gz. kelebihan lain yang berbeda dari versi untuk windows adalah memiliki fitur untuk mengaktifkan sebuah server secara grafis, sedangkan linux masih berupa perintah-perintah didalam *console*. oleh karena itu versi untuk linux sulit untuk dioperasikan.

2.1.13 UML (*Unified Modeling Language*)

Menurut Widodo (2012:10) “Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan

diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram Use Case (*Use Case Diagram*)

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalan itu *apa yang dilakukan* bukan *bagaimana melakukannya*. Diagram Use Case dekat kaitannya dengan kejadian-kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. Untuk lebih memperjelas lihat gambaran suatu peristiwa untuk sebuah klinik kesehatan di bawah ini :

“Pasien menghubungi klinik untuk membuat janji (*appointment*) dalam pemeriksaan tahunan. *Receptionist* mendapatkan waktu yang luang pada buku jadwal dan memasukkan janji tersebut ke dalam waktu luang itu”.



Gambar II.1
Contoh kegiatan pasien yang membuat janji

Diagram Use Case berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada (*requirements*)

Use Case baru selalu menghasilkan fasilitas baru ketika sistem di analisa dan *design* menjadi lebih jelas.

2. Komunikasi dengan klien

Penggunaan notasi dan simbol dalam diagram *Use Case* membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.

3. Membuat test dari kasus-kasus secara umum

Kumpulan dari kejadian - kejadian untuk *Use Case* bisa dilakukan test kasus layak untuk kejadian - kejadian tersebut.

2. Diagram Kelas (*Class Diagram*)

Diagram Class memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka. Diagram Class bersifat statis, menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.

Diagram Class mempunyai 3 macam *relationships* (hubungan), sebagai berikut:

1. ***Association***

Suatu hubungan antara bagian dari dua kelas. Terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.

2. **Aggregation**

Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian. Sebagai contoh : Order Detail merupakan kumpulan dari Order.

3. **Generalization**

Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *super Class* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *super Class*.

Package dan *Object* untuk mengatur pengorganisasian diagram Class yang *kompleks*, dapat dilakukan pengelompokan kelas-kelas berupa *package* (paketpaket). *Package* adalah kumpulan elemen-elemen logika UML.

3. **Diagram Interaksi dan Sequence (Sequence Diagram)**

Diagram Class dan diagram Object merupakan suatu gambaran *model statis*. Namun ada juga yang bersifat *dinamis*, seperti Diagram Interaction. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

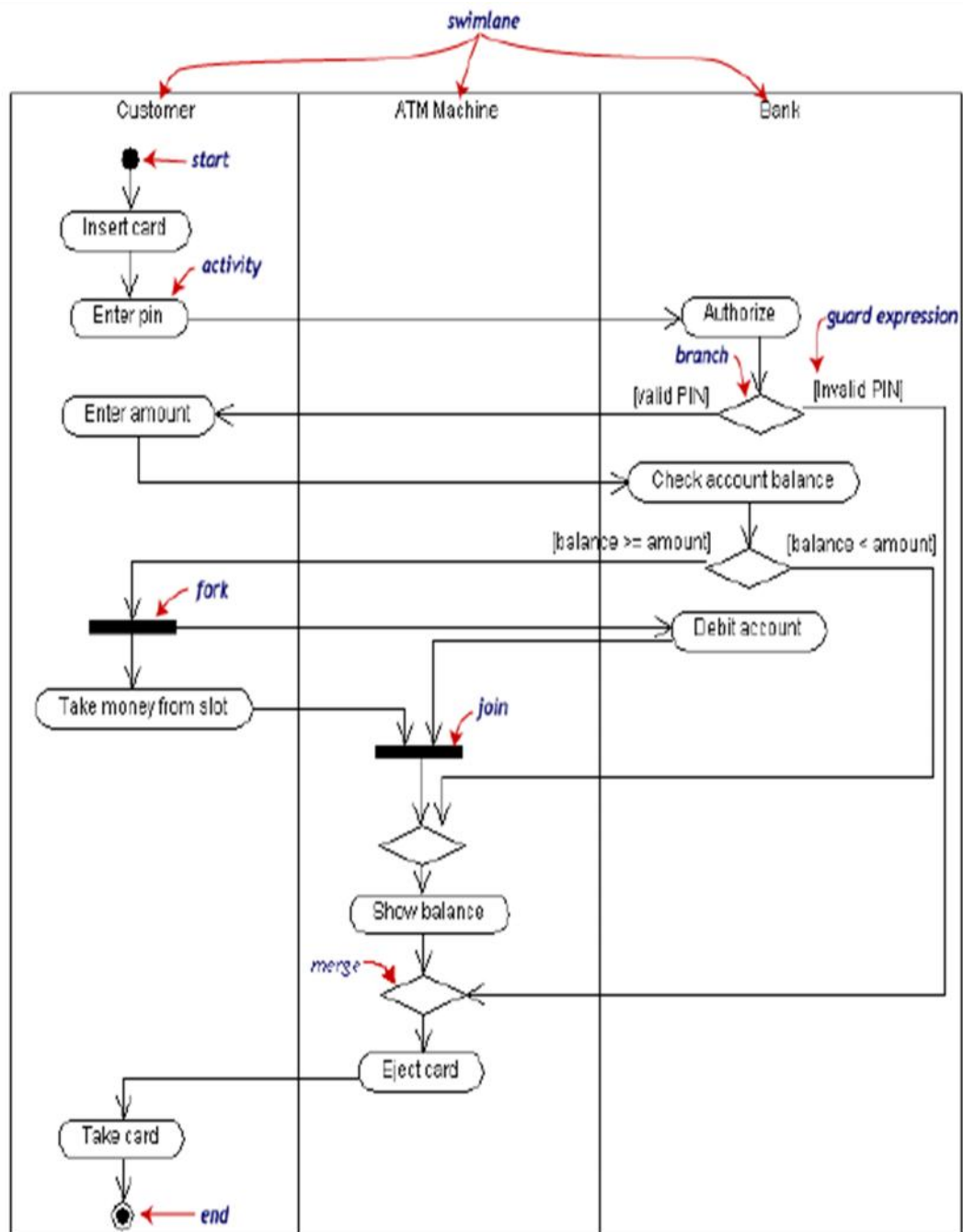
4. **Diagram StateChart (StateChart Diagram)**

Behaviors dan *state* dimiliki oleh obyek. Keadaan dari suatu obyek bergantung pada kegiatan dan keadaan yang berlaku pada saat itu. Diagram StateChart

menunjukkan kemungkinan dari keadaan obyek dan proses yang menyebabkan perubahan pada keadaannya.

5. Diagram Aktivitas (*Activity Diagram*)

Pada dasarnya diagram Activity sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada *obyek yang dalam suatu proses* (atau proses menjadi suatu obyek), diagram Activity berfokus pada *aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal*. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Sebagai contoh, perhatikan proses yang terjadi. “Pengambilan uang dari bank melalui ATM.” Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait, yaitu Customer, ATM dan Bank. Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran stop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi. Lihat gambar dibawah ini agar lebih jelas.



Gambar II.2
Contoh Diagram Activity 'Pengambilan Uang melalui ATM'

Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek yang mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal (*single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungi pada aktifitas berikutnya. Sebuah *transition* (transisi) dapat membuat cabang ke dua atau lebih percabangan *exclusive transition* (transisi eksklusif).

Label *Guard Expression* (ada di dalam []) yang menerangkan output (keluaran) dari percabangan. percabangan akan menghasilkan bentuk menyerupai bentuk intan. *transition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut **Fork**. *Fork* beserta *join* (gabungan dari hasil output *fork*) dalam diagram berbentuk *solid bar* (batang penuh).

6. Diagram komponen (*Component Diagram*)

Bersifat statis, Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen - komponen yang telah ada sebelumnya.

7. Diagram deployment (*deployment diagram*)

Bersifat statis, Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram dan sebagainya.

8. Diagram komunikasi (*Communication Diagram*)

Bersifat dinamis, diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

9. Diagram paket (*Package Diagram*)

Bersifat statis, diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model dan mempelajari 3 (tiga) elemen utama dari UML, seperti *building block*, aturan - aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

a. Building blocks

Tiga macam yang terdapat dalam building block adalah :

1. Benda *Things*

Adalah abstraksi yang pertama dalam sebuah model

2. Hubungan/*Relationships*

Sebagai alat komunikasi dari benda-benda

3. Bagan/*Diagrams*

Sebagai kumpulan / group dari benda-benda/things

b. Benda/*Things*

Adalah hal yang sangat mendasar dalam model UML juga merupakan bagian paling statik dari sebuah model serta menjelaskan elemen elemen lainnya dari

sebuah konsep dan atau fisik. Bentuk dari beberapa benda atau thing adalah sebagai berikut :

- 1) **Classes**, yang diuraikan sebagai sekelompok dari *object* yang mempunyai *atribute*, operasi, hubungan yang semantik. Sebuah kelas mengimplementasikan 1 atau lebih *interfaces*. Sebuah kelas dapat digambarkan sebagai sebuah persegi panjang, yang mempunyai sebuah nama, *atribute* dan metoda pengoperasiannya.
- 2) **Interfaces**, merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. *Interface* atau antar-muka mendefinisikan sebuah set atau kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah antar-muka berdiri sendiri dan umumnya merupakan pelengkap dari kelas atau komponen.
- 3) **Collaboration**, yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas - kelas atau elemen - elemen yang bekerja secara bersama-sama. *Collaborations* mempunyai struktur dan dimensi. Pemberian sebuah kelas memungkinkan berpartisipasi didalam beberapa *collaborations* dan digambarkan dengan sebuah 'elips' dengan garis terpotong-potong.
- 4) **Use cases**, adalah rangkaian/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 atau *things* dalam sebuah model serta di realisasikan oleh sebuah *collaboration*.

Umumnya ‘*use case*’ digambarkan dengan sebuah ‘*elips*’ dengan garis yang solid, biasanya mengandung nama.

- 5) **Nodes**, merupakan fisik dari elemen - elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalah sebuah komputer, umumnya mempunyai sedikitnya *memory* dan *processor*. Sekelompok komponen mungkin terletak pada sebuah node dan juga mungkin akan berpindah dari node satu ke node lainnya. Umumnya node ini digambarkan seperti kubus serta hanya mengandung namanya.

c. Hubungan / *Relationship*

Ada 4 macam hubungan didalam penggunaan UML, yaitu;

- 1) **Dependency**, adalah hubungan semantik antara dua benda atau *things* yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah *dependency* digambarkan sebuah panah dengan garis terputus - putus.
- 2) **Association**, hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *association* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.
- 3) **Generalizations**, adalah menggambarkan hubungan khusus dalam obyek anak atau *child* yang menggantikan obyek *parent* atau induk . Dalam hal ini, obyek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada obyek induk. Digambarkan dengan garis panah.

- 4) **Realizations**, merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau *elements*, serta antara *use cases* dan *collaborations*. Model dari sebuah hubungan *realization*.

2.1.14 ERD (Entity Relationship Diagram)

Menurut Rosa (2013:50) “ERD (*Entity Relationship Diagram*) adalah dikembangkan berdasarkan teori himpunan dalam dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD”.

A. Komponen ERD:

1. Entitas (Entity)

Entitas adalah sesuatu yang nyata atau abstrak di mana kita kan menyimpan data.

2. Relasi (Relationship)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai.

3. Atribut (Attribute)

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

4. Kardinalitas

Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan.

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

1. One-to-One, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A .
2. One-to-Many, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B . Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A .
3. Many-to-One, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B . Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A .
4. Many-to-many, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan nol atau lebih entitas pada A .

2.1.15 Pengujian Sistem

Blackbox testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar.

Jenis testing ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Sebagai contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi inventory di sebuah perusahaan. Maka pada jenis white box testing, perangkat lunak tersebut akan berusaha dibongkar listing programnya untuk kemudian dites menggunakan teknik-teknik yang telah dijelaskan sebelumnya. Sedangkan pada jenis black box testing, perangkat lunak tersebut akan dieksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar listing programnya (Rizky.2011:264).

2.2 Penelitian Terkait

Sistem Inventory pada obat-obatan di Rumah Sakit Umum Daerah (RSUD) Padang pengolahan datanya masih menggunakan sistem manual. Penanganan data dengan sistem manual ini mempunyai beberapa kendala, diantaranya menyebabkan terjadinya kesalahan dalam pengecekan stok obat dan setiap laporannya terjadi redundancy data, serta kurang telitinya dalam pencatatan stok obat yang masuk maupun keluar sehingga terjadinya kekeliruan dalam pencatatan stok akhir. Hal-hal tersebut menyebabkan proses kerja tidak efektif dan efisien. Proses inventory obat yang baik akan mengurangi kesulitan dalam mengontrol inventory maupun pengeluaran obat. dengan di rancanganya sebuah sistem informasi, diharapkan semua obat-obat yang masuk dan keluar yang ada digudang dapat diketahui dengan jelas tanpa adanya kesalahan-kesalahan yang sifatnya mengganggu dalam proses penyediaan obat nantinya, Dan dengan menggunakan web akan lebih mudah dalam menyajikan informasi dan dapat memonitor keadaan persediaan obat-obatan di Rumah Sakit umum Daerah Padang (Susanti,Minarni.2014:104).

Di Laboratorium SMK Negeri 1 Rembang saat ini sudah menggunakan komputer sebagai sarana untuk mengerjakan kegiatan-kegiatan yang berkaitan dengan administrasi maupun manajemen, namun masih sebatas untuk mengetik mendata inventaris laboratorium (Meja, Kursi, AC, JPS, CPU, LCD Monitor, Sound, LCD proyektor, Papan tulis dan alat-alat praktikum) dan mengetik laporan-laporan lainnya, kegiatankegiatan tersebut belum menggunakan program khusus untuk

membuat laporan yang berhubungan dengan pengolahan data maupun inventaris sehingga mengakibatkan keterlambatan dalam inventarisasi, maka diperlukan sistem baru dengan menggunakan program khusus untuk mengolah data dan untuk memasukan data yang meliputi data inventaris alat dan bahan, data laporan kegiatan dan perawatan laboratorium. Dengan adanya sistem baru akan mempermudah informasi yang cepat dan akan membantu kepala laboratorium maupun pimpinan lainnya dalam menerima laporan data inventaris serta dapat menyajikan informasi yang cepat dan akurat kepada pengguna lainnya (Lutfhi.Riasti, 2011:69).

