

BAB II

LANDASAN TEORI

2.1. Pengertian Konsep Dasar Sistem

Sistem menurut Soeherman dalam Sumirat dan Jakaria (2017:2) diartikan sebagai, “Serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu”.

Sedangkan menurut Rohmat dalam Sumirat dan Jakaria (2017:2) “Sistem berasal dari bahasa Yunani yaitu ‘Sistema’ berarti kesatuan, yakni keseluruhan dari bagian-bagian yang mempunyai hubungan satu dengan yang lainnya”.

Menurut Pratama dalam Sumirat dan Jakaria (2017:2) Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware* dan *brainware*. Ketiga Komponen tersebut berkaitan satu sama lain.

Jadi, sistem merupakan sebuah kumpulan komponen-komponen yang saling berinteraksi untuk melakukan suatu kegiatan sehingga kegiatan tersebut dapat berjalan lebih optimal dan dapat mencapai tujuan tertentu.

Menurut Hutahaean (2014:3) mengemukakan, supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu:

1. Komponen (*component*)

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

2. Batasan sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan sistem (*input*)

Masukan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Sedangkan menurut Hutahacan (2014:6) sistem dapat diklasifikasikan dalam beberapa sudut pandang:

1. Sistem diklasifikasikan sebagai:

a. Sistem abstrak (*abstract system*)

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.

b. Sistem fisik (*Physical system*)

Sistem fisik adalah sistem yang ada secara fisik.

2. Sistem diklasifikasikan sebagai:

a. Sistem alamiah (*natural system*)

Sistem alamiah adalah sistem yang terjadi melalui proses awal, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi.

b. Sistem buatan manusia (*human made system*)

Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine system*).

3. Sistem diklasifikasikan sebagai:

a. Sistem tertentu (*deterministic system*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi sebagai keluaran sistem yang dapat diramalkan.

b. Sistem tak tertentu (*probalistic sytem*)

Sistem tak tentu adalah sistem yan kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probablistik.

4. Sistem diklasifikasikan sebagai:

a. Sistem tertutup (*close system*)

Sistem tertup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar. Sistem bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*.

b. Sistem terbuka (*open system*)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem itu menerima *input* dan *output* dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik.

2.1.1. Pengertian Sistem Informasi

Menurut Sutabri dalam Sumirat dan Jakaria (2017:2), “Sistem merupakan suatu bentuk integrasi antara satu komponen dengan lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi dalam sistem tersebut”.

Menurut Davis dalam Kadir (2014:45) “informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang”.

Menurut McFadden dalam Kadir (2014:45) “mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut”.

Jadi, sistem informasi adalah serangkaian komponen-komponen yang saling berinteraksi yang dapat menghasilkan data yang akan diolah untuk membentuk sebuah informasi yang akurat untuk oranglain.

Menurut Hutahaean (2014:13) Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu:

1. Blok masukan (*input block*)

Input mewakili data yang masuk 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, logik 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 block*)

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:

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

5. Blok basis data (*database block*)

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

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

2.1.2. Pengertian Sistem Persediaan

Persediaan merupakan unsur yang paling aktif dalam operasi perusahaan dagang dan perusahaan industri serta perusahaan jasa. Tanpa adanya persediaan, para pengusaha akan dihadapkan pada keadaan bahwa perusahaannya pada suatu waktu tidak dapat memenuhi keinginan para pelanggannya sehingga kontinuitas perusahaan dapat terganggu karena sumber utama pendapatan perusahaan berasal dari penjualan persediaan. Ini berarti perusahaan akan kehilangan kesempatan untuk memperoleh keuntungan yang seterusnya didapatkan. Istilah persediaan memberikan pengertian yang berbeda-beda tetapi pada dasarnya maksud dan tujuannya adalah sama.

Menurut C. Rolln Niwwonger, Philip E. Fess dan Carl S. Wareen dalam Sumirat dan Jakaria (2017:2) “istilah persediaan (*inventory*) merupakan barang dagangan yang disimpan untuk dijual dalam operasi perusahaan dan merupakan barang yang terdapat dalam proses produksi atau yang disimpan untuk tujuan itu”.

2.1.3. Konsep Dasar Program

Menurut Sukmaindrayana dan Sidik (2017:32) “Program adalah kumpulan intruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu. Tanpa adanya program komputer hanyalah perangkat keras (*hardware*) yang tidak bisa melakukan apa-apa”.

Adapun definisi program menurut Fakhri dalam Sukmaindrayana dan Sidik (2017:32) ”Program adalah algoritma yang ditulis dalam salah satu bahasa komputer yang dapat dijalankan pada komputer”.

Jadi, program merupakan kata, ekspresi, atau pernyataan yang disusun dan dirangkai menjadi satu kesatuan prosedur, yang berupa urutan langkah, untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer.

2.1.4. Pengertian Bahasa Pemrograman

Menurut Setiawan dalam Sukmaindrayana dan Sidik (2017:33) mengemukakan bahwa, bahasa pemrograman adalah teknik komando atau intruksi standar untuk memerintah komputer yang merupakan suatu himpunan dari aturan sintaks dan sistematik yang dipakai untuk mendefinisikan program komputer.

2.1.5. Pengertian Bahasa Pemrograman Java

Menurut Sukamto dan Shalahuddin (2016:101) mendefinisikan “Java merupakan bahasa pemrograman yang paling konsisten dalam mengimplementasikan paradigma pemrograman berorientasi objek.

Menurut Sukamto dan Shalahuddin (2016:103) “Java merupakan bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas.”

2.1.6. Pengertian Pemrograman Berorientasi Objek

Menurut Sukamto dan Shalahuddin (2016:100) mengemukakan bahwa metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

2.1.7. Pengertian Pemrograman Berbasis Desktop

Menurut Hilman (2016) Aplikasi desktop merupakan aplikasi yang dioperasikan dengan cara menginstalnya terlebih dulu di sistem operasi. Sesuai namanya, aplikasi ini mengacu pada program yang diinstal pada perangkat desktop (PC dan laptop) mampu beroperasi secara *offline*, tetapi kita harus menginstalnya sendiri pada laptop atau komputer.

2.1.8. Pengertian Basis Data (*Database*)

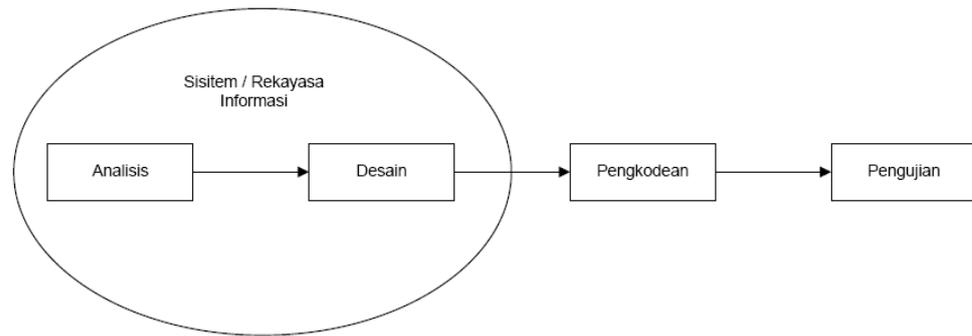
Menurut Mandalamaya dalam Sukmaindrayana dan Sidik (2017:33) mengemukakan bahwa *Database* adalah sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan penggunaan dalam mengelolanya juga memudahkan memperoleh informasi.

Menurut Sukanto dan Shalahuddin (2016:43) mendefinisikan bahwa Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Jadi, basis data (*database*) adalah sebuah tempat penyimpanan data yang sudah terkomputerisas dan terdapat beberapa tabel yang saling berelasi.

2.1.9. Model Pengembangan Perangkat Lunak

Model pengembangan sistem yang digunakan dalam penulisan Tugas Akhir ini penulis menggunakan SDLC model *Water fall*. Menurut Sukanto dan Shalahuddin (2016:28) menjelaskan bahwa “model SDLC air terjun (*water fall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). Berikut ini adalah gambar model air terjun:



Sumber : Sukamto dan Shalahuddin (2016:29)

Gambar II.1. Ilustrasi Model *Waterfall*

1. Analisa Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari tahap analisis spesifikasi untuk perubahan perangkat lunak baru.

2.1.10. *My SQL (My Structure Query Language)*

Menurut Kadir dalam Sumirat dan Jakaria (2017:5), “*MySQL* adalah salah satu jenis *database server* yang sangat populer, hal ini disebabkan karena *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*”.

MySQL bersifat *Open Source*, software ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), bentuk *executablenya* atau kode yang dapat dijalankan secara langsung dalam sistem operasi. *SQL (structure query language)*, kriteria kecepatan, pemakaian memori, mudah tidaknya program, dan daya tampung data menjadi kriteria utama.

Selain keutamaan *SQL Server* sebagai penampung *database* cukup besar dan dukungannya terhadap bahasa *SQL*. *SQL Server* memiliki banyak kemiripan

dengan *Microsoft Access* dalam hal fasilitas-fasilitas yang dimilikinya tetapi menyediakan fasilitas-fasilitas tambahan karena *SQL Server* ditujukan untuk aplikasi-aplikasi berskala besar dimana data-data dengan jumlah yang sangat banyak perlu diorganisasi dengan seksama.

2.1.11. Xampp

Xampp adalah *software web server apache* yang di dalamnya tertanam *serverMySQL* yang didukung dengan bahasa pemrograman *PHP* untuk membuat *website* yang dinamis. *Xampp* mendukung dua sistem operasi yaitu *windows* dan *Linux*.

Menurut Yogajiwanjaya dalam Noor (2016:78) “ Untuk *linux* proses penginstalannya menggunakan *command line* sedangkan untuk *windows* dalam proses penginstalannya menggunakan *interface* grafis sehingga lebih mudah dalam penggunaan *xampp* di *Windows* dibanding dengan *Linux*.”

2.1.12. Jurnal Terkait

1. Amiennia Rektiani dan Anak Agung Gde Agung, (2015: Vol.02, No.02 ISSN : 2087 2429), dengan judul aplikasi pencatatan persediaan, penjualan dan pesanan dengan metode perpetual. Kesimpulan dari penelitian ini adalah Aplikasi dapat melakukan pencatatan transaksi penjualan dan pesanan, Aplikasi dapat menghitung dan menampilkan jumlah stok barang melalui kartu persediaan dengan metode *average* secara otomatis, Aplikasi dapat menampilkan laporan berupa jurnal dan buku besar secara otomatis. Dengan adanya aplikasi ini, selain dapat mencatat transaksi penjualan dan pesanan, manajemen dapat mengetahui

persediaan barang yang dimilikinya secara *real time*, sehingga diharapkan dapat meningkatkan keunggulan kompetitif perusahaan.

2. Margareta Nawang, Laela Kurniawati dan Dudi Duta, (2017: Vol. 13, No. 2 P-ISSN:1978-1946|E-ISSN: 2527-6514), dengan judul rancang bangun sistem informasi pengolahan data persediaan barang berbasis dekstop dengan model *waterfal*. Kesimpulan dari penelitian ini adalah banyaknya pengeluaran barang yang tidak sesuai dengan pencatatan dikarenakan banyaknya pekerjaan yang dilakukan dengan manual; adanya sistem informasi pengelolaan data persediaan barang pada PT. Sakura Yasa Prima maka sistem yang telah terkomputerisasi; sistem yang terkomputerisasi akan mempermudah segala aktifitas operasional perusahaan; kesalahan-kesalahan yang terjadi dengan menggunakan sistem manual dapat diminimalkan dengan sistem yang telah terkomputerisasi; sistem yang sudah terkomputerisasi mampu menghasilkan informasi atau keluaran yang lebih tepat dan akurat guna membantu kelancaran pekerjaan sehari-hari, dan model *waterfall* cukup efektif digunakan sebagai model pengembangan system karena langkah-langkahnya mudah untuk diterapkan.
3. Ani Oktarini Sari dan Elan Nuari, (2017 : Vol. 13, No.2 P-ISSN: 1978-1946 | E-ISSN: 2527-6514), dengan judul rancang bangun sistem informasi persediaan barang berbasis web dengan metode *fast (framework for the applications)*. Kesimpulan dari penelitian ini adalah Dengan adanya sistem terkomputerisasi untuk persediaan barang pada PT. Solusi Aksesindo Pratama diharapkan permasalahan-permasalahan yang telah dibahas dapat terselesaikan. Penggunaan hasil perancangan ini juga membutuhkan partisipasi aktif dari pemakai sistem, terutama kedisiplinan para pelaksana yang menangani secara langsung pada

sistem yang dirancang. Beberapa keuntungan sistem terkomputerisasi ini antara lain: Pengolahan data untuk keluar masuk barang lebih efektif dan efisien; pencarian data dapat lebih efisien karena data sudah terorganisir dengan baik sesuai dengan level akses admin, sales dan gudang; data persediaan barang/stock barang dapat terpantau dengan baik; file data barang masuk dan keluar tersimpan dengan baik dan dapat diakses dengan mudah ketika dibutuhkan; dan proses pelaporan menjadi lebih baik, karena dapat di akses dan dicetak langsung.

4. Sifa Fauziah dan Ratnawati, (2018 : Vol. 4, No. 1 P-ISSN: 2442-2436, E-ISSN: 2550-0120), dengan judul penerapan metode fifo pada sistem informasi persediaan barang. Kesimpulan Perancangan Sistem Informasi persediaan barang di bagian gudang PT Fivalco Indonesia merupakan perancangan dari sistem yang sedang berjalan. Aplikasi sistem informasi persediaan barang dengan menerapkan metode FIFO ini dapat mempermudah dan mempercepat kinerja petugas bagian gudang dalam mengakomodasi perhitungan stok persediaan data barang. Selain itu Dengan adanya aplikasi ini perusahaan dapat melihat persediaan barang dalam jangka waktu tertentu serta informasi tentang stok minimum dan maksimum sehingga dapat lebih teratur dalam proses transaksi persediaan barang sehingga waktu proses menjadi relatif lebih cepat dan dapat meningkatkan kinerja bagian gudang dalam pencatatan barang.
5. Oky Irnawati, (2017 : Vol. 3. No. 1 E-ISSN: 2527-4864), dengan judul perancangan program persediaan barang dengan *java desktop* pada PT. Pakartel Dengan digantinya sistem manual menjadi sistem terkomputerisasi untuk persediaan barang pada PT. Pakartel diharapkan permasalahanpermasalahan yang telah dibahas akan terselesaikan. Kecepatan dan ketepatan hasil

perancangan ini juga membutuhkan partisipasi aktif dari pemakai sistem, terutama kedisiplinan para pelaksana yang menangani secara langsung pada sistem yang dirancang. Dengan adanya jasa komputer sebagai alat bantu, beberapa keuntungan sistem terkomputerisasi ini antara lain, data persediaan barang lebih akurat, pengolahan data untuk keluar masuk barang lebih efektif dan efisien, memudahkan petugas untuk memantau dan mengontrol data dalam *database*, pengawasan terhadap pelaksanaan kegiatan lebih mudah, karena adanya laporan atau informasi kegiatan yang tepat waktu dan dapat disajikan setiap saat bila dibutuhkan.

2.2. Teori Pendukung (*Tools System*)

2.2.1. *Unified Modelling Language (UML)*

Menurut Sukamto dan Shalahuddin (2016:137) mendefinisikan bahwa UML Bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. 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.

Menurut Gata dan Grace dalam Hendini (2016:108), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

1. *Activity Diagram*

Menurut Sukamto dan Shalahuddin (2016:161) diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau 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.

Menurut Hendini (2016:109) “*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis”.

Jadi, *activity diagram* adalah bagaimana aktor melakukan aktivitas atau bisa dikatakan bagaimana interaksi antara aktor dengan sistem.

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

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.

2. *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2016:155) mendefinisikan bahwa *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi itu.

Menurut Sukamto dan Shalahuddin (2016:155) ada dua hal utama pada *use case* yaitu:

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 fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor

Jadi, *use case diagram* adalah apa yang sedang aktor lakukan untuk mengetahui sebuah informasi.

3. *Class Diagram*

Menurut Sukamto dan Shalahuddin (2016:141) “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 operasi yaitu:

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

Berdasarkan beberapa pendapat para ahli yang dikemukakan diatas, maka dapat disimpulkan bahwa *class diagram* memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek.

4. *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2016:165) mendefinisikan gambar 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.

Menurut Sukamto dan Shalahuddin (2016:165) banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

2.2.2. Entity Relationship Diagram (ERD)

Menurut Jogianto dalam Sumirat dan Jakaria (2017:15) “*Entity Relationship Diagram* atau ERD adalah suatu model jaringan yang menggambarkan *layout* (susunan) penyimpanan data dari sebuah sistem ER-Diagram yang menggambarkan data-data dalam keadaan diam (data yang disimpan)”.

Menurut Utomo dalam Rahmayu (2015:161) “ERD merupakan *tool* analisis sistem pertama yang memusatkan pada data dan keterkaitan antar data serta pengorganisasian data”.

Menurut Ladjamudin dalam Rahmayu (2015:161) “*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam secara abstrak.”

Adapun komponen-komponen ERD menurut Ladjamudin dalam Rahmayu (2015:161), hubungan entitas terdiri dari :

1. Entity

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

2. Relationship

Pada E-R diagram, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif

atau kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

3. *Relationship Degree*

Relationship degree atau derajat relationship adalah jumlah entitas yang berpartisipasi dalam satu *relationship*. Definisi derajat *relationship* menurut Ladjamudin dalam Rahmayu (2015:161) “*Relationship Degree* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.” Berikut ini derajat *relationship* yang sering dipakai didalam ERD:

a. *Unary Relationship*

b. *Binary Relationship*

c. *Tenary Relationship*

4. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *relationship*.

5. Kardinalitas Relasi

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Adapun kardinalitas relasi:

a. Satu ke satu (*One to One*)

Setiap elemen dari Entitas A berhubungan paling banyak dengan elemen pada Entitas B. Demikian juga sebaliknya setiap elemen B berhubungan paling banyak satu elemen pada Entitas A.

b. Satu ke banyak (*One to Many*)

Setiap elemen dari Entitas A berhubungan dengan maksimal banyak elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan paling banyak satu elemen di Entitas A.

c. Banyak ke satu (*Many to One*)

Setiap elemen dari Entitas A berhubungan paling banyak dengan satu elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan maksimal banyak elemen di entitas A.

d. Banyak ke banyak (*Many to Many*)

Setiap elemen dari Entitas A berhubungan maksimal banyak elemen pada Entitas B demikian sebaliknya.

UNIVERSITAS

2.2.3. Logical Record Structure (LRS)

LRS (*Logical Record Structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas.

Menurut Wulandari dalam Sumirat dan Jakaria (2017:3) “*Logical Record Structure* dibentuk dengan nomor dari *tipe record*. Beberapa *tipe record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik.”

Beda LRS dengan diagram *entity relationship diagram* nama *tipe record* berada diluar kotak *field tipe record* ditempatkan. LRS terdiri dari *link-link* diantara *tipe record*. *Link* ini menunjukkan arah dari satu *tipe record* lainnya. Banyak *link*

dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *linktipe record*. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan *Entity Relationship Diagram* dan langsung dikonversikan ke LRS.

Menurut Frieyadie dalam Rahmayu (2015:162) “LRS merupakan hasil dari pemodelan *Entity Relationshipship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”. Dalam pembuatan LRS terdapat tiga hal yang dapat mempengaruhi yaitu:

- a. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
- b. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
- c. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.