

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Dalam pendefinisian sistem terbagi mejadi dua kelompok, yang lebih menekankan pada prosedur dan juga pada komponen atau elemennya. Pendekatan sistem adalah “*Suat bentuk jaringan kerja dari prosedur-prosedur yang saling berhubungan, yang berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu tujuan tertentu*” (Tata Sutabri, 2014). Pendekatan sistem yang merupakan suatu bentuk jaringan kerja dari sebuah prosedur yang lebih menekankan pada urutan-urutan operasi yang berada di dalam sistem. Sedangkan menurut Gordon B.Davis sebagai berikut: “*Sistem adalah sekumpulan dari elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu*” (Tata Sutabri, 2014).

2.2 Informasi

Informasi merupakan proses lanjutan dari suatu data yang sudah memiliki nilai lebih. Informasi dikelompokkan menjadi 3 bagian, yaitu:

1. Informasi Strategis.

Informasi ini dibutuhkan untuk pengambilan sebuah keputusan jangka panjang, yang mencakup sebuah informasi *eksternal*, suatu rencana untuk memperluas sebuah perusahaan, dan sebagainya.

2. Informasi Taktis.

Informasi ini dibutuhkan untuk pengambilan sebuah keputusan jangka menengah, misalnya informasi tren penjualan yang dapat dimanfaatkan untuk membuat sebuah rencana penjualan.

3. Informasi Teknis.

Informasi ini dibutuhkan untuk suatu keperluan operasional dalam keseharian, seperti informasi persediaan stok barang, retur, dan laporan kas harian. Informasi adalah suatu data yang sudah diolah atau diinterpretasikan yang digunakan dalam proses pengambilan keputusan (Tata Sutabri, 2014).

2.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan–laporan yang diperlukan. (Halimah, 2015).

2.4 Basis Data

Basis data adalah sebuah sistem yang terkomputerisasi yang mempunyai tujuan utama yaitu pemeliharaan data atau informasi yang sudah diolah dan membuat informasi menjadi tersedia saat dibutuhkan. Basis data pada dasarnya adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan.

(Rz Abdul Aziz, 2019).

Kebutuhan suatu basis data didalam sistem informasi meliputi:

1. Memasukkan, menyimpan, dan mengambil data
2. Membuat laporan yang berdasarkan pada data yang telah disimpan

Tujuan dari pembuatan tabel tersebut adalah untuk menyimpan data kedalam tabel-tabel agar dapat mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka akan membutuhkan pola piker penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana pada setiap baris terdapat beberapa kolom, (Rosa A.S.M.Shalahuddin, 2016).

2.5 Pengertian Analisis dan Desain Berorientasi Objek (OOAD)

OOAD adalah suatu metode untuk menganalisis dan memeriksa *requirements* dari sudut pandang kelas-kelas dan objek yang dapat ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Metode OOAD adalah metode baru dalam menyelesaikan suatu masalah yang menggunakan model berdasarkan pada konsep pada dunia nyata. Dasar pembuatannya adalah objek, yang merupakan kombinasi dari struktur data dan perilaku dalam satu entitas.

2.5.1 Konsep Dasar OOAD

OOAD memiliki cakupan analisis dan desain dari sebuah sistem dengan pendekatan suatu objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa *requirement* (syarat/keperluan yang harus dipenuhi dalam sistem) dari sudut pandang kelas-kelas dan objek-objek yang dapat ditemukan dalam ruang lingkup suatu perusahaan. Sedangkan OOD adalah metode yang lebih menekankan kepada arahan arsitektur software yang berdasar pada manipulasi objek-objek sistem atau subsistem. Terdapat beberapa konsep dalam metode OOAD, yaitu:

1. Kelas (*Class*)

Kumpulan objek-objek yang memiliki karakter yang sama. Suatu kelas memiliki sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan mempunyai arti. Suatu kelas dapat diturunkan dari kelas yang lain. Dimana atribut dan kelas awalnya dapat diwariskan ke kelas yang baru. Kelas adalah sebuah struktur untuk membuat sebuah perangkat lunak. Kelas merupakan bentuk dari struktur yang terdapat pada kode program dengan menggunakan metode berorientasi objek.

2. Objek (*Object*)

Abstraksi dan sesuatu yang mewakilkan suatu kejadian pada dunia nyata. Objek merupakan suatu entitas yang mampu untuk menyimpan sebuah informasi (status) dan memiliki operasi (kelakuan) yang mampu diterapkan.

3. Metode (*Method*)

Metode atau operasi pada suatu kelas hampir sama dengan fungsi atau prosedurnya yang terdapat pada metode terstruktur. Operasi adalah fungsi atau transformasi yang dapat dilakukan pada objek.

4. Atribut

Suatu variabel global yang dipunyai oleh sebuah kelas. Atribut dapat juga berupa nilai atau elemen-elemen data yang dipunyai oleh objek dalam sebuah kelas. Atribut dipunyai setiap individu pada suatu objek contohnya berat, jenis dan lain-lain.

5. Abstraksi

Suatu prinsip untuk merepresentasikan kejadian pada dunia nyata menjadi suatu bentuk model yang begitu sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan yang terjadi.

6. Enkapsulasi

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai pada setiap objek untuk menyembunyikan implementasi dan objek sehingga objek-objek lain tidak dapat mengetahui cara kerjanya.

7. Pewarisan

Mekanisme yang memungkinkan suatu objek yang mewarisi sebagian ataupun seluruh objek lain sebagai bagian dari dirinya.

8. Antarmuka

Biasanya digunakan agar kelas yang lain tidak bisa mengakses langsung ke dalam suatu kelas.

9. *Reusability*

Pemanfaatan kembali suatu objek yang sudah didefinisikan pada suatu permasalahan yang ada pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi&spesialisasi

Menunjukkan hubungan antar kelas dan objek yang umum dengan kelas dan objek khusus.

11. Komunikasi antar objek

Dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. *Polimorfisme*

Kemampuan yang dimiliki suatu objek yang digunakan ke banyak tujuan yang berbeda-beda dengan nama yang sama sehingga dapat menghemat sebuah program.

13. *Package*

Sebuah container atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas yang memiliki kesamaan nama disimpan dalam package yang berbeda.

Object Oriented Analysis and Design (OOAD) dapat di definisikan menjadi dua bagian yaitu:

1. *Object Oriented Analysis*

Adalah suatu metode analisis yang memeriksa requirements (syarat/keperluan yang harus dipenuhi sistem) dilihat dari kelas dan objek yang dapat ditemui dalam pengembangan sistem. OOA mempunyai sebuah aktivitas utama yaitu:

- a. Menganalisa masalah domain
- b. Menjelaskan sebuah proses pembentukan sistem
- c. Mengidentifikasi objek
- d. Menentukan atribut
- e. Mengidentifikasi metod atau operasi
- f. Komunikasi antar objek

2. *Object Oriented Design*

Adalah metode yang dapat mengarahkan arsitektur *software* sehingga sesuai dengan manipulasi sistem maupun sub sistemnya. OOD mempunyai sebuah aktivitas utama yaitu:

- a. Mengidentifikasi konteks dan model dari penggunaan sistem
- b. Mendesain arsitektur sistem
- c. Identifikasi objek dari sistem utama
- d. Menentukan interface objek

2.6 Definisi Unified Modeling Language (UML)

Menurut Rosa A.S.M. Shalahudin (2016, p.137) UML merupakan sebuah bahasa visual yang diperuntukan untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukungnya. UML hanya dapat berfungsi untuk melakukan pemodelan, jadinya penggunaan UML tidak hanya terbatas pada sebuah metodologi tertentu, meskipun dalam kenyataannya UML juga banyak digunakan pada metodologi yang berorientasi objek. Perkembangan penggunaan UML tergantung pada level abstraksi dalam penggunaannya, didalam penggunaan UML adas uatu hal yang dapat menjadi kesalahan, jadi perlu ditelaah lagi dimanakah UML bisa digunakan dan apa saja yang ingin divisualkan. Menurut Rosa A.SM. Shalahudin (2016, p.140).

Pada UML terdapat 13 macam diagram yang dibagi menjadi 3 kategori.

1. *Use Case Diagram*


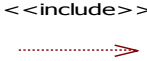


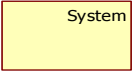

Use case atau diagram *Use case* merupakan suatu pemodelan untuk kelakuan (*behavior*) pada sistem informasi yang akan dibuat. *Use case* mendefinisikan sebuah interaksi yang terjadi antar satu atau lebih actor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi itu, (Rosa A.SM. Shalahuddin, 2016).

Ada 2 hal utama yang terdapat pada use case pendefinisian yang disebut aktor dan juga *use case*.

1. Aktor merupakan orang, proses, atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat diluar konteks sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai suatu unit yang saling bertukar kesan antar unit atau aktor.

Berikut adalah tabel simbol-simbol yang ada pada diagram *use case*:

Tabel 2.1 Simbol Use Case Diagram

No.	Gambar	Nama	Keterangan
1		Aktor	Mendefinisikan himpunan peran yang dimainkan pengguna ketika sedang berinteraksi dengan use case.
2		<i>Include</i>	Mendefinisikan bahwa use case sumber secara eksplisit.
3		<i>Extend</i>	Mendefinisikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4		<i>Association</i>	Sesuatu yang menghubungkan antarobjek satu dengan objek lainnya.
5		Sistem	Mendefinisikan suatu paket yang menampilkan sistem secara terbatas.
6		<i>Use Case</i>	Definisi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor







2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan suatu aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau suatu proses bisnis atau menu yang terdapat pada perangkat lunak. Perlu diperhatikan disini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang telah dilakukan aktor, jadi aktivitas yang telah dilakukan oleh sistem. aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut ini:

1. Rancangan proses bisnis dimana pada setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang telah didenfinisikan.
2. Urutan pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan oleh perangkat lunak.

Berikut adalah tabel simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol Activity Diagram

No.	Gambar	Nama	Keterangan
1		Swimlane	Menunjukkan bahwa siapa yang bertanggung jawab dalam melakukan aktivitas dalam suatu diagram.
2		Action	Langkah-langkah dalam sebuah aktivitas. Action bisa terjadi saat memasuki activity, meninggalkan activity, atau pada event yang spesifik.
3		Initial State	Menunjukkan dimana aliran kerja dapat dimulai.
4		Activity Final Node	Menunjukkan dimana aliran kerja akan diakhiri.
5		Decision Node	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi dan dua atau lebih transisi sesuai suatu kondisi.
6		Control Flow	Menunjukkan bagaimana kendali suatu aktivitas yang terjadi pada aliran kerja dalam suatu tindakan tertentu.

3. Class Diagram

Class Diagram bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

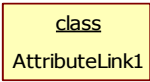
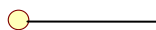





(Dona Yuliawati, 2018).

Kelas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Atribut adalah variabel-variabel yang dipunyai dalam suatu kelas.
2. Operasi atau metode adalah fungsi yang dipunyai dalam suatu kelas.

Berikut adalah table simbol-simbol yang ada pada diagram kelas:

Tabel 2.3 Simbol Class Diagram



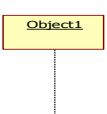



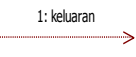
No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem.
2		Antarmuka/interface	Sama dengan konsep antarmuka dalam pemrograman berorientasi objek.
3		Asosiasi	Relasi antar kelas yang memiliki makna umum, asosiasi biasanya disertai dengan multiplicity.
4		Asosiasi berarah	Relasi antar kelas dengan makna kelas pada saat digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan multiplicity.
5		Generalisasi	Relasi antar kelas yang mempunyai makna generalisasi dan spesialisasi (umum dan khusus).
6		Kebergantungan	Relasi antar kelas yang mempunyai makna kebergantungan antar kelas.
7		Agregasi	Relasi antar kelas yang mempunyai makna semua bagian (whole-part).

4. Sequence Diagram

Sequence Diagram melambangkan interaksi antar objek didalam dan disekitar termasuk pengguna (*user*), *display*, dan sebagainya berupa *message* yang menggambarkan waktu. *Sequence diagram* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan sebuah scenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan sebuah output. *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima oleh objek. Oleh karena itu, untuk menggambarkan *sequence diagram* harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta dengan metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek. Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Sugiarti, 2013).

Berikutnya adalah tabel simbol-simbol yang ada pada Sequence Diagram:

Tabel 2.4 Simbol Sequence Diagram

No.	Gambar	Nama	Keterangan
1		Aktor	Objek <i>entity</i> , antarmuka yang saling berinteraksi satu sama lain.
2		Garis hidup	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi sebuah pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan sedang aktif dan berinteraksi pesan.
5		Object Message	Mendeskripsikan pesan (<i>message</i>) antar dua objek.
6		Message to self	Mendeskripsikan pesan (<i>message</i>) kepada dirinya sendiri.
7		Pesan tipe return	Mendeskripsikan pengembalian dari pemanggilan prosedur

2.7 Android

Android adalah sebuah sistem operasi untuk smartphone dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (device) dan penggunanya, sehingga pengguna bisa berinteraksi dengan device-nya dan menjalankan aplikasi-aplikasi yang tersedia pada device. Mobile phone adalah salah satu perangkat yang bergerak seperti telepon seluler atau komputer bergerak yang digunakan untuk mengakses jasa jaringannya. Pada mobile application juga digunakan untuk mendeskripsikan aplikasi internet yang berjalan pada smartphone serta piranti mobile lainnya.

(Yuni Puspita Sari, 2019)

2.8 Android SDK

Menurut (Nadia Firly, 2018) Android SDK adalah merupakan sebuah kit yang berfungsi untuk mengembangkan berbagai aplikasi berbasis Android oleh para *developer*. Didalam SDK telah terdapat berbagai tools yang bertujuan untuk proses pengembangan aplikasi seperti proses *debugger*, *emulator*, *software libraries*, dan dokumentasi.

2.9 PHP (Hypertext Preprocessor)

Menurut (Jubilee Enterprise, 2018) PHP merupakan sebuah Bahasa untuk pemrograman biasanya digunakan untuk membuat website yang dinamis dan interaktif. Dinamis memiliki arti bahwa website tersebut dapat berubah-ubah tampilan dan juga kontennya sesuai dengan kondisinya. Misalkan, PHP dapat menampilkan tanggal dan hari pada saat ini secara berganti-ganti di dalam suatu website. Interaktif artinya PHP dapat member timbal balik bagi *user* (misalkan menampilkan hasil pencarian suatu produk).

2.10 Java

Definisi java dari *Sun Microsystem* adalah nama dari kumpulan teknologi untuk menjalankan perangkat lunak pada komputer *standalone* yang beradapada jaringan. *Java 2* adalah generasi kedua dari *Platform Java*. *Java* berdiri atas sebuah mesin *interpreter* yang diberinama *Java Virtual Machine*

(JVM). JVM inilah yang akan membaca *bytecode* dalam *file class* dari suatu program sebagai representasi langsung program yang berisi sebuah bahasa mesin. Oleh karena itu bahasa *Java* disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan sistem operasi tersebut terdapat JVM. *Java* merupakan sebuah bahasa pemrograman objek murni karena semua kode programnya dibuat dalam sebuah kelas. Saat ini *Sun Microsystem* sudah diakuisisi *Oracle Corporation* sehingga pengembangan *Java* diteruskan oleh *Oracle Corporation*.

(Rosa A.S. M. Shalahuddin, 2016).

2.11 MySQL

MySQL adalah sebuah perangkat lunak (*software*) sistem manajemen basis data *SQL* atau *DBMS Multithread* dan *multiuser*. *MySQL* merupakan turunan dari salah satu konsep utama dalam database untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara otomatis. *MySQL* diciptakan oleh Michael "Monty" Widenius pada tahun 1979, seorang *programmer* komputer asal Swedia yang mengembangkan sebuah sistem *database* sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level *ISAM database engine* dengan indexing.

Menurut Rosa A.S.M. Shalahuddin (2016) mengemukakan bahwa “*MySQL* adalah salah satu *database* manajemen sistem (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MSSQL*, *Postgre SQL*, dan lainnya”. Dimana *MySQL* dalam operasi *client-server* melibatkan *server* daemon *MySQL* disisi *server* dan berbagai macam program serta *library* berjalan yang besar. *SQL* merupakan singkatan dari *Structure Query Language* dan sering disebut *Sequel*. *SQL* mulai dikembangkan tahun 70-an di laboratorium IBM, Stan Jose, California. Untuk mengakses sebuah *file database*, salah satu *server database* yang kecil dan mudah digunakan namun memiliki kehandalan dan *performa* tinggi.

2.12 GIS (*Geographic Information Sistem*)

SIG merupakan sebuah sistem informasi yang dibuat untuk bekerja dengan sumber data spasial. SIG merupakan suatu media yang sangat handal untuk

mempresentasikan data *Remote Sensing* (RS) menjadi informasi yang berguna untuk banyak orang untuk berbagai keperluan. Data spasial adalah data-data yang memiliki sistem koordinat geografis. Dengan kata lain SIG merupakan suatu sistem *database* yang memiliki kemampuan spesifik untuk melakukan operasi tertentu pada data. Teknologi SIG biasanya telah terintegrasi dengan teknologi *database* seperti *query* dan analisa statistik dengan tampilan yang unik, serta analisis geografis yang menguntungkan dengan peta. Kemampuan ini yang membuat SIG berbeda dengan sistem informasi lainnya, sehingga SIG menjadi barang yang sangat berharga bagi masyarakat umum dan perusahaan untuk menjelaskan peristiwa, memprediksi pendapatan dan perencanaan strategis. SIG meliputi pemahaman tentang: pola, ruang, dan proses (metodologi) yang dibutuhkan untuk penyelesaian masalah. Selain obyektif, SIG juga dapat dijadikan sebagai alat tindakan yang sangat cepat dan efisien. SIG akan sangat berguna bila pemakai memahami konsep spasial secara utuh (Indarto, 2013).

2.13 Virtual Reality

Virtual Reality adalah pemunculan gambar-gambar tiga dimensi yang dibuat komputer sehingga terlihat nyata dengan bantuan sejumlah peralatan tertentu, yang menjadikan penggunaannya seolah-olah terlibat langsung secara fisik dalam lingkungan tersebut. Virtual Reality membutuhkan perangkat yang dirancang untuk tujuan tertentu dalam teknologi ini, sehingga mampu menjadikan orang yang merasakan dunia maya terkecoh dan yakin bahwa yang dialaminya adalah nyata. Beberapa perangkat yang digunakan antara lain: Force balls/trackingballs, Controller wands, Voice recognition, Headset, Joysticks / gamepad, Data gloves, Treadmills, Motion trackers/bodysuits.

Saat berada dalam lingkungan virtual, pengguna akan merasa seolah menyatu dengan dunianya dan dapat berinteraksi dengan objek-objek yang ada di sana. Hal ini disebut dengan telepresence. Telepresence diartikan sebagai pengalaman keberadaan seseorang terhadap lingkungan melalui sebuah media. Penggunaan istilah telepresence digunakan untuk jenis komunikasi yang menggunakan media dalam tujuannya menghadirkan persepsi.

(Triowali Rosandy, 2020).