



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Perangkat Lunak

Menurut Gary dan Misty (dikutp Mahesa 2019:7) “Perangkat lunak adalah seperangkat perintah terkait, disusun untuk tujuan yang umum, memerintahkan kepada komputer apa yang harus dilakukan dan bagaimana melakukannya.”

menurut Sukamto dan Shalahuddin (2018:2) “Perangkat lunak (*Software*) adalah program komputer terasosiasi dengan dokumentasi perangkat lunak seperti doukumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa perangkat lunak merupakan cara ditujukan kepada komputer sesuai kebutuhan pemakai yang digunakan untuk memproses informasi dan berfungsi melakukan operasi yang mengurus tentang segala aktifitas komputer seperti mendukung sistem aplikasi dan mengendalikan semua perangkat komputer agar dapat berjalan dengan selaras.

2.1.2 Komputer

Menurut Kadir (dikutip Akbar 2019:6) “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan pekerjaan yang dilakukan oleh manusia.”

Menurut Paramytha (dikutip Akbar 2019:6) “Menyatakan bahwa computer berasal dari bahasa latin yaitu *Computer* yang berarti menghitung (*to compute* atau *reckon*). Kata komputer pada awalnya dipergunakan untuk menggambarkan pekerjaan orang yang melakukan perhitungan arimatika dengan atau tanpa alat bantu.”



Berdasarkan pendapat di atas, dapat disimpulkan bahwa definisi dari komputer adalah sebuah alat elektronik yang dapat memproses data digital dan informasi yang bermanfaat bagi *user* sehingga dapat menghasilkan keluaran informasi yang dihasilkan setelah diolah.

2.1.3. Perangkat Keras (*Hardware*)

Menurut Sutabri (2014:32), “Perangkat keras sebagai subsistem dari sistem komputer juga mempunyai komponen, yaitu

1. Komponen peranti masukan (*input device*) yaitu alat yang digunakan untuk menerima masukan data atau masukan program misalnya *keyboard, scanner, sensor, dan pengenalan suara*.
2. Komponen peranti pemroses (*processing device*), yaitu alat di mana instruksi-instruksi program dieksekusi untuk memproses data yang dimasukkan lewat alat masukan yang hasilnya nanti akan ditampilkan dialat output misalnya *CPU (central processing unit)*.
3. Komponen peranti keluaran (*output device*), yaitu alat untuk menampilkan pengolahan data seperti tulisan (huruf, kata, angka, karakter, dan simbol-simbol), *image* (bentuk grafik atau gambar) dan suara (bentuk musik) misalnya *hardcopy device (printer)* dan *softcopy device (monitor)*.
4. Komponen peranti penyimpanan (*storage*) eksternal adalah penyimpanan luar, karena terletak diluar alat prosesnya atau disebut *mass storage* atau penyimpanan masal karena penyimpanan umumnya lebih besar dari *main memory*”. Lain halnya menurut Kadir (2017:2), “Perangkat keras adalah peranti-peranti yang terkait dengan komputer dan terlihat secara fisik. *Monitor, hard disk, dan mouse* adalah contoh perangkat keras”.

Dari beberapa definisi perangkat keras penulis menyimpulkan bahwa perangkat keras adalah beberapa komponen yang bekerja saling mendukung sesuai dengan instruksi *software*.



2.1.4 Data

Menurut Akbar (2019:7) “Data disimpan secara terintegrasi (integrated), artinya basis data merupakan gabungan dari berbagai macam file aplikasi yang

Menurut Siregar (dikutip 2019:8) “Data adalah kumpulan fakta atau segala sesuatu yang dapat dipercaya kebenarannya sehingga dapat digunakan sebagai dasar untuk menarik suatu kesimpulan.”

Dari pendapat diatas dapat disimpulkan bahwa data adalah kumpulan data baik berupa angka, kata-kata atau citra yang didddapatkan melalui pengukuran atau pengamatan yang memerlukan pengolahan selanjutnya untuk menghasilkan informasi yang akurat.

2.1.5 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (Rational Unified Process). Sukanto dan Shalahuddin (2016:125), “RUP (Rational Unified Process) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (iterative), fokus pada arsitektur (architecture-centric), lebih diarahkan berdasarkan penggunaan kasus (use case driven)’. Memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif.berikut merupakan penjelasam untuk setiap fase pada RUP.

a. Fase Inception (Permulaan)

Tahap dimana kita memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahap yang dibutuhkan pada tahap ini:

- a. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya)
- b. Membangun kasus bisnis yang dibutuhkan

Hasil yang diharapkan pada tahap ini adalah memenuhi *Lifecycle Objective Milestone* (batas/tonggak objektif dari siklus) dengan kriteria berikut:



- a. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, dan perkiraan jadwal
- b. Kebutuhan dimengerti dengan pasti (dapat dibuktikan) dan sejalan dengan kasus primer yang dibutuhkan.
- c. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, resiko dan proses pengembangan.
- d. Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan
- e. Membangun garis besar dengan membandingkan perencanaan yang direncanakan.

Pada fase ini, penulis menyimpulkan beberapa kegiatan yang dilakukan berdasarkan hal-hal diatas, diantaranya penyelidikan awal, studi kelayakan hingga kebutuhan fungsional dan non fungsional.

1. Menurut Akbar (2019:44) “Studi Kelayakan sendiri merupakan. Suatu proses mempelajari dan menganalisa masalah yang telah ditentukan sesuai dengan tujuan yang akan dicapai meliputi mempelajari struktur sistem dan mengembangkan alternatif pemecahan masalah yang telah ditentukan.”
2. Menurut Muhidin dkk (2017:61) “Kebutuhan fungsional Dari hasil analisis kebutuhan fungsional dibutuhkan suatu sistem informasi yang bermanfaat bagi guru, siswa dan pengunjung yaitu media informasi berbasis website. Dimana informasi yang disajikan dapat diakses dengan menggunakan media internet dan admin dapat manajemen database pada sistem.”

b. Fase Elaboration (Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini lebih pada analisis dan desain sistem. Hasil yang diharapkan dari tahap ini memenuhi *Lifeycl Ojjective Milestone* (batas/tonggak objektif dari siklus) dengan kriteria, model kasus yang digunakan (*use case*) dimana kasus dan



aktor yang terlibat telah diidentifikasi dan sebagian besar harus dikembangkan.

c. Fase Construction (Konstruksi)

Tahap dimana kita mengembangkan komponen dan fitur-fitur sistem. Implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Hasil dari fase ini yaitu implementasi sistem informasi monitoring berbasis website dan hasil pengujian sistem yaitu dengan blackbox validation testing dan compatibility testing. Menurut Hariyanto (2018:270), terdapat dua teknik pengujian black-box yaitu validation testing dan compability testing.

Menurut Hidayat dan Muttaqin (2018:27) “*Black box Testinnng* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.”

d. Fase Transition (Transisi)

Tahap dimana kita deployment atau Instalasi sistem agar dapat dimengerti oleh user. Aktifitas pada tahap ini termasuk pada pelatihan user dan pemeliharaan. Produk perangkat lunak juga disesuaikan dengan kebutuhan yang didefinisikan pada tahap *inception*. Jika semua kriteria objektif maka dianggap sudah memenuhi *Product Release Milestone* (batas/tonggak peluncuran produk) dan pengembangan perangkat lunak selesai dilakukan.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap, keempat fase pada RUP dijalankan secara urut dan berulang, dengan setiap iterasi digunakan untuk memperbaiki iterasi berikutnya.

2.2 Teori Khusus

2.2.1 Kamus Data

Menurut Sukamto dan Shalahuddin (2018:73) ”Kamus Data (alah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan



(*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Kamus data memiliki beberapa simbol sebagai berikut :

Tabel 2.1 Simbol-simbol Kamus Data

No	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau...
4.	{ } ⁿ	n kali diulang/bernilai banyak
5.	()	Data opsional
6.	*..*	Batas komentar

Sumber: Sukamto dan Shalahuddin (2016:74)

2.2.2 Unified Modeling Language (UML)

Menurut Sukamto dan Shalahuddin (2018:133) “UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat alisis dan desain, serta menggambarkan aritektur dalam pemerograman berorientasi objek”.

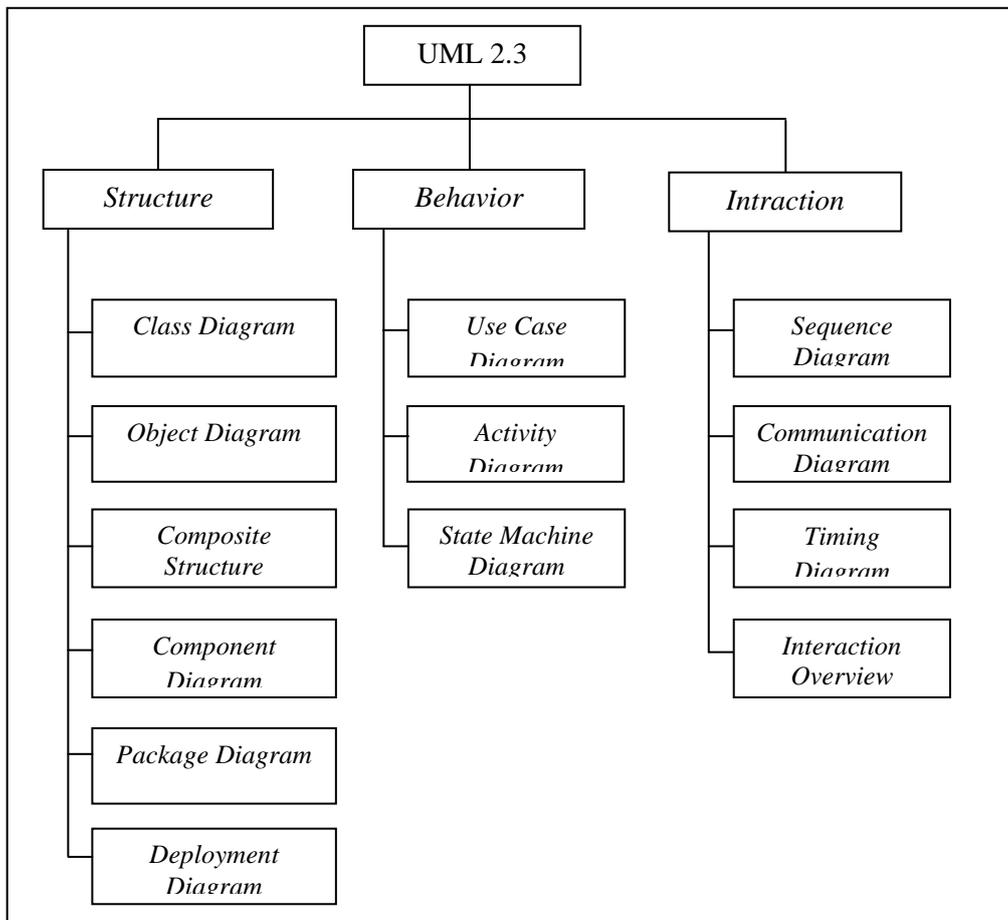
UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.



Gambar 2.1. Tampilan Logo UML

2.2.3 Kategori *Unified Modeling Language* (UML)

Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



Gambar 2.2 Kategori dan Macam-macam Diagram UML

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukanto dan Shalahuddin (2018:141) :

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk me
- 2) nggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 3) *Behavior diagrams* yaitu kumpulam diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahanana yang terjadi pada sebuah sistem.
- 4) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

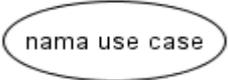
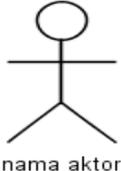


2.2.4 Jenis-Jenis Diagram UML

2.2.4.1 Use case Diagram

Menurut Sukamto dan Shalahuddin (2016:155), “*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. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

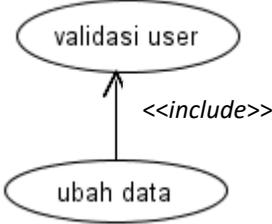
Tabel 2.2 Simbol-simbol Use case Diagram

No	Simbol	Deskripsi
1.		<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2.		<p>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; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
3.		<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.</p>
4.		<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat</p>

Lanjutan Tabel 2.2 Simbol-simbol *Use case Diagram*

No	Simbol	Deskripsi
	<p data-bbox="448 495 719 562"><<extend>> </p>	<p data-bbox="754 439 1358 745">berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p> <div data-bbox="783 792 1091 1272" style="text-align: center;"> <pre> graph TD A([validasi user]) -.-> <<extend>> B([validasi username]) C([validasi sidik jari]) -.-> <<extend>> A </pre> </div> <p data-bbox="754 1323 1358 1518">arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>
5.	<p data-bbox="411 1547 727 1630">Generalisasi / <i>generalization</i></p> <p data-bbox="427 1666 687 1697"></p>	<p data-bbox="754 1547 1358 1798">hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>

Lanjutan Tabel 2.2 Simbol-simbol *Use case Diagram*

No	Simbol	Deskripsi
		<p>akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT UC1(ubah data) -- <<include>> UC2(validasi user) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Rosa dan Shalahuddin (2018:156-158)

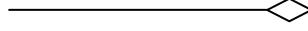
2.2.4.2 Class Diagram

Menurut Sukamto dan Shalahuddin (2018:141), “diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. 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.”

Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol *Class Diagram*



No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2	antarmuka / interface  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
4.	asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: Sukamto dan Shalahuddin (2018:146-

2.2.4.3 Activity Diagram

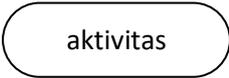
Menurut Sukamto dan Shalahuddin (2018:161), “Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan



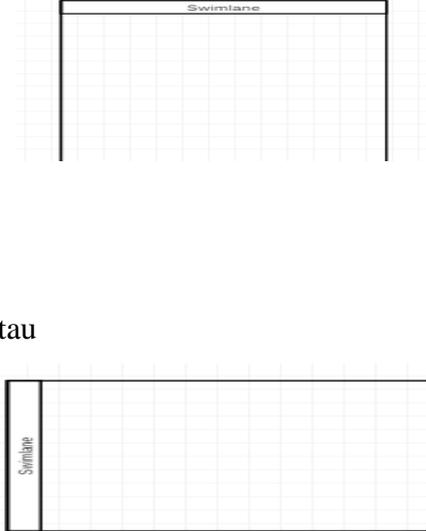
aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.”

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

Lanjutan Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
6.	<p data-bbox="391 432 523 465">Swimlane</p>  <p data-bbox="391 824 448 857">atau</p>	<p data-bbox="885 432 1364 582">Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

Sumber: Sukanto dan Shalahuddin (2018:162-163)

2.2.4.4 Sequence Diagram

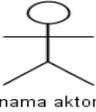
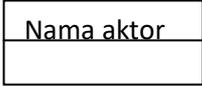
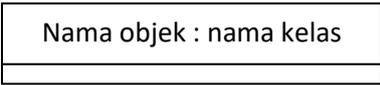
Menurut Sukanto dan Shalahuddin (2018:165). “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”.

Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

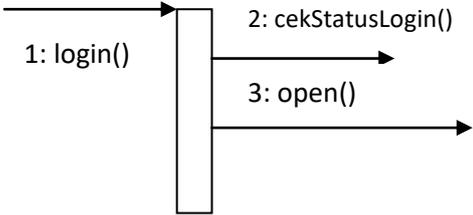
Berikut simbol-simbol pada Sequence Diagram :



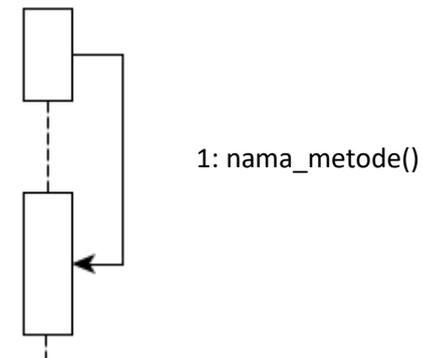
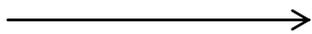
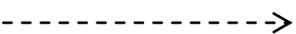
Tabel 2.5 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
1.	<p>Actor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>menyatakan objek yang berinteraksi pesan</p>

Lanjutan **Tabel 2.5** Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
4.	Waktu aktif 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5.	Pesan tipe create <<create>> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe call 1: nama metode() 	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,

Lanjutan **Tabel 2.5** Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
		 <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
7.	Pesan tipe send 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe return 1: keluaran 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian

Lanjutan **Tabel 2.5** Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
9.		menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

Sumber: Sukamto dan Shalahuddin (2018:165-

2.3 Teori Judul

2.3.1 Implementasi

Menurut Pressman dan Wildavsky (dikutip Signe 2017:10) “Kemampuan untuk menjalin hubungan berikutnya dalam rantai sebab-akibat sehingga mendapatkan hasil yang diinginkan”.

Menurut Fixsen (dikutip Signe 2017:10) “Seperangkat kegiatan tertentu yang dirancang untuk mempraktikkan suatu kegiatan atau program dari dimensi yang diketahui, prosesnya bertujuan dan dijelaskan secara cukup rinci sehingga pengamat independen dapat mendeteksi keberadaan dan kekuatan”.

Dari pendapat diatas dapat disimpulkan bahwa pengertian Implementasi adalah suatu tindakan atau rencana yang telah disusun secara rinci dan cermat (matang).

2.3.2 Pengendalian

Menurut Hery (dikutip Angkasa dkk, 2019:) “Seperangkat kebijakan dan prosedur untuk melindungi asset atau kekayaan perusahaan dari segala bentuk tindakan dan penyalahgunaan, menjamin tersedianya informasi akuntansi perusahaan yang akurat, serta memastikan bahwa semua ketentuan (peraturan)



hukum/undang-undang serta kebijakan manajemen telah dipatuhi atau dijalankan sebagaimana mestinya oleh seluruh karyawan perusahaan”.

Menurut Zahri (2019:) ”Adalah proses dalam menetapkan ukuran kinerja dan pengambilan tindakan yang dapat mendukung pencapaian hasil yang diharapkan sesuai dengan kinerja yang telah ditetapkan tersebut”.

Dari pendapat diatas jadi dapat disimpulkan bahwa pengertian pengendalian alalah menjamin sumber yang diperoleh dan digunakan dengan efektif dan efisien dalam rangka pencapaian.

2.3.3 Stok

Menurut Raymond McLeod (dikutip Mujiati 2014:25) “Adalah sebagai suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam periode usaha yang normal”.

Memurut Kamus Besar Bahasa Indonesia (<https://kbbi.web.id/stok>, diunduh pada tanggal 07 Juli 2020, pukul 11.02 WIB) Menyatakan bahwa stok adalah persediaan barang keperluan untuk perbekalan.

Berdasarkan pengertian diatas dapat disimpulkan bahwa stok merupakan sejumlah barang yang disediakan untuk memenuhi kebutuhan.

2.3.4 Obat

Menurut Nuryati (2017:21) “Merupakan suatu zat atau bahan-bahan yang berguna dalam menetapkan diagnose, mencegah, mengurangi menghilangkan, menyembuhkan penyakit atau atau gejala penyakit, luka atau kelainan fisik dan



rohani pada manusia atau hewan, termasuk memoercantik tubuh atau bagian tubuh manusia”.

Menurut Indijah dan Fajri (2016:2) “Obat didefinisikan sebagai senyawa yang digunakan untuk mencah, mengobati, mendiagnosis penyakit atau gangguan, atau menimbulkan kondisi tertentu”.

Dari pendapat diatas jadi dapat disimpulkan bahwa pengertian obat alalah senyawa yang dapat digunakan untuk mencegah atau mengobati penyakit agar tidak lebih parah.

2.3.5 Puskesmas

Menurut Ruliyandari (2018:8) “Pusesmas merupakan salah satu tempat pelayanan kesehatan utama bagi sebagian besar masyarakat di daerah di seluruh indonesia”.

Menurut Irmawati dkk (2019:189) “Puskesmas adalah fasilitas pelayanan kesehatan yang menyelenggarakan upaya kesehatan masyarakat dan upaya kesehatan perseroangan tingkat pertama, dengan lebih mengutamakan upaya promotif dan preventif, untuk mencapai derajat kesehatan masyarakat yang setinggi-tingginya diwilayah kerjanya”.

Kesimpulannya dari pendapat diatas adalah puskesmas merupakan pusat pengembangan kesehatan masyarakat yang juga membina peran serta masyarakat dan memberikan pelyanan secara menyeluruh dan terpadu kepada masyarakat diwilayah kerjanya.

2.3.6 Website

Menurut Krisnawati dkk (2018:38) “Salah satu yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang



menggunakan protokol *HTTP* (*hypertext transfer protocol*) dan mengaksesnya menggunakan perangkat lunak yang disebut browser”.

Menurut Abdullah (dikutip Destiningrum dan Adrian, 2017:8) “Sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

Dari pendapat diatas dapat disimpulkan bahwa *website* sebuah *software* yang berfungsi menampilkan dokumen-dokumen pada suatu web yang membuat pengguna dapat mengakses internet melalui *software* yang terkoneksi dengan internet.

2.3.7 Implementasi pada Pengendalian Stok Obat di Puskesmas Tanjung Lago berbasis *website*.”

Aplikasi yang mengelola stok obat yang keluar dan masuk, stok barang berupa obat-obatan pada puskesmas Tanjung Lago.

2.4 Teori Program

2.4.1 Basis Data

Menurut Sukanto dan Shalahuddin (2018:43) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Menurut Kristanto (dikutip Akbar 2019:7) “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”.

Kesimpulannya, Basis data (*database*) adalah kumpulan informasi yang disimpan dalam komputer secara sistematis agar dapat diakses dengan mudah dan cepat, sebagai aktifitas dari satu atau lebih organisasi.



2.4.2 MySQL

Menurut Palit (dikutip Akbar 2019:22) “MySQL adalah sebuah basis data yang mengandung satu atau jumlah table. Table terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah table”.

Menurut Solichin (dikutip Niovianty 2019:24) “HTML merupakan bahasa pemrograman *web* yang memberitahukan peramban *web* (*web browser*) bagaimana menyusun dan menyajikan konten dihalaman *web*”.

Dari beberapa pendapat di atas dapat disimpulkan bahwa *MySQL* adalah salah satu jenis *database* yang siapa saja boleh menggunakannya yang mengimplementasikan dari sistem manajemen basis data relasional.

2.4.3 Xampp

Menurut Riyanto (dikutip Akbar 2019:22) “XAMPP merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*”.

Menurut Palit (dikutip Akbar 2019:22) “XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program”.

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa XAMPP yaitu aplikasi web server dengan kebutuhan akan program-program telah terinstal dan digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*.

2.4.4 JavaScript

Winano dikutip Akbar (2019:24) mengatakan bahwa *JavaScript* biasanya digunakan untuk memberikan efek program di halaman, juga menjadi dasar yang



digunakan teknologi lainnya seperti *jQuery*, *Ajax*, *jQuery Mobile*, *JavaScript* adalah bahasa *scripting client site*,

Menurut Novianty (dikutip Akbar) “*Javascript* adalah untuk menambah fungsionalitas dan kenyamanan halaman *web*, *Javascript* lebih focus pada proses pengolahan data di sisi *client* serta menyajikan komponen *web* yang lebih interaktif”

Berdasarkan kedua pendapat diatas, maka dapat disimpulkan bahwa java adalah Bahasa pemrograman yang sangat populer untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat internet atau jaringan komunikasi.

2.4.5 Framework

Menurut Sidik (dikutip Destiningrum dan Adrian 2017:32) “*Framework* adalah kumpulan intruksi-intruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu”.

“Sebuah perangkat lunak kerangka kerja yang membantu membuat dalam pengembangan situs web secara mudah dan cepat”

Berdasarkan kedua pendapat diatas, adalah sebuah *software* untuk mempermudah membuat aplikasi web yang didalamnya ada beberapa fungsi, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dterstruktur dengan rapih.

2.4.6 Codeigniter

Menurut sidik (dikutip Destiningrum dan Adrian 2017:32) “*Codeigniter* adalah sebuah *framework php* yang bersifat *open source* dan menggunakan metode *MVC (Model, View, Contoller)* untuk memudahkan *developer* atau



programmer dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuatnya dari awal”

Destiningrum dan Adrian (2017:8) “Menyebutkan bahwa *codeigniter* merupakan *framework* PHP yang kuat dan sedikit bug. *Codeigniter* ini dibangun untuk para pengembang dengan Bahasa pemrograman PHP yang membutuhkan alat untuk membuat *web* dengan fitur lengkap. ”

Berdasarkan kedua pendapat diatas, maka dapat disimpulkan bahwa *codeigniter* adalah *framework* yang dibuat berdasarkan *design pattern* model *view controller* atau bisa disingkat *MVC*.