



---

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Menurut Ermawita, dan Ahmad Zainy (2019:1), Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (input), mengolah data (proses), dan memberikan informasi (output) serta terkoordinasi dibawah kontrol program yang tersimpan di memori komputer. Jika pada zaman sebelumnya penggunaan komputer hanya terbatas pada beberapa aktivitas saja maka pada zaman sekarang komputer dapat digunakan dalam berbagai aspek kehidupan seperti perusahaan, sekolah, perkantoran, rumah dan tempat-tempat umum sebagai layanan informasi.

Menurut Kadir, Abdul (2017:2), Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia. Meskipun komputer berasal dari kata “Komputasi”, komputasi yang memang dilaksanakannya mungkin tidak terlihat secara eksplisit. Ketika orang menggunakan komputer untuk membuat dokumen, berbagai perhitungan yang dilakukan tidak terlihat. Sebagai contoh, ketika pemakai memilih pengaturan “*justify*” atau rata di batas kiri dan batas kanan dokumen, perhitungan untuk menambahkan karakter-karakter spasi sebenarnya terjadi.

Dari definisi di atas penulis menyimpulkan bahwa komputer adalah perangkat keras yang dapat digunakan untuk membantu dalam berbagai pekerjaan manusia seperti pembuatan dokumen.

##### 2.1.2. Pengertian Perangkat Lunak

Menurut Kadir, Abdul (2017:2), Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada *computer* agar dapat melaksanakan tugas sesuai kehendak pemakai. System operasi seperti *Windows*, *Mac OS*, dan *Linux*, dan aplikasi seperti *Microsoft Word* dan *Microsoft Excel* adalah contoh perangkat lunak.



Dari definisi di atas penulis menyimpulkan bahwa perangkat lunak adalah suatu program aplikasi yang digunakan sesuai dengan perintah yang ditujukan untuk menjalankan tugasnya.

### 2.1.3. Pengertian Perangkat Keras

Menurut Ansori, Z dan Yulmaini, (2019:2), Perangkat keras komputer (*hardware*) adalah semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, seperti : *Motherboard, Power Supply, Processor, RAM, Hardisk, CD Drive, Battery CMOS, VGA Card, Sound Card*, dll. Untuk dapat digunakan pada saat memasukkan data, memproses data, dan menghasilkan informasi maka paling sedikit perangkat komputer harus terdiri dari *keyboard, CPU (Central Processing Unit), monitor, dan mouse*.

Berdasarkan dari fungsinya, peralatan komputer dapat dibedakan menjadi dua macam, yaitu :

a. Peralatan masukan (*Input*), seperti :

- 1) *Keyboard*
- 2) *Mouse*
- 3) *Microphone*
- 4) *Scanner*
- 5) *Joystick*

b. Peralatan keluaran (*Output*), seperti :

- 1) *Monitor*
- 2) *Speaker*
- 3) *Printer*

Lain halnya menurut Kadir, Abdul (2017:2), “Perangkat keras adalah peranti-peranti yang terkait dengan komputer dan terlihat secara fisik. *Monitor, hard disk, dan mouse* adalah contoh perangkat keras”.

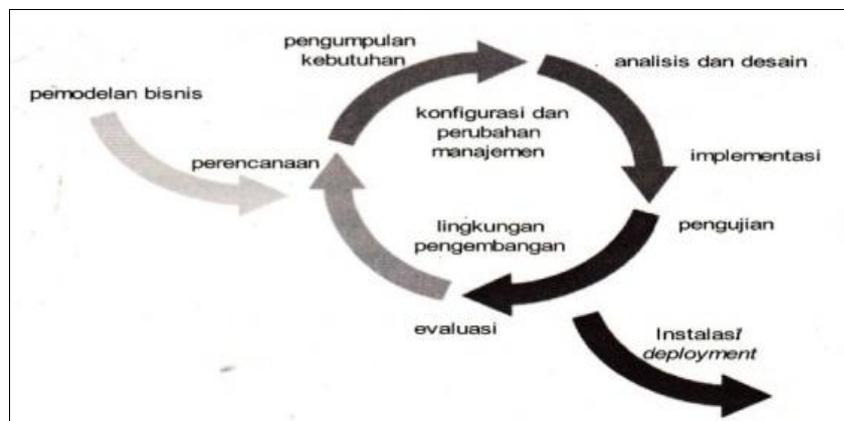
Dari beberapa definisi perangkat keras penulis menyimpulkan bahwa perangkat keras adalah beberapa komponen yang bekerja saling mendukung sesuai dengan instruksi *software*.



#### 2.1.4. Metode Pengembangan Sistem

Menurut Sukamto dan Shalahuddin (dikutip Fitria dan Widiowati, 2017:28), RUP (*Rational Unified Process*) merupakan pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*welldefined*) dan penstrukturan yang baik (*wellstructured*)”.

Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada gambar berikut:



(Sumber: Sukamto dan Shalahuddin (dikutip Fitria dan Widiowati, 2017:28))

**Gambar 2.1** Proses iteratif RUP (*Rational Unified Process*)

##### 2.1.4.1. Fase *Rational Unified Process* (RUP)

RUP adalah metodologi pengembangan perangkat lunak, yang diformulasikan oleh *Rational Software Corporation* (sekarang menjadi salah satu divisi IBM), RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language (UML)* sebagai bahasa permodelan selama periode pengembangan dan *iterative incremental* sebagai model siklus pengembangan perangkat lunak (Santoso, E. dkk. 2016:2). Metode RUP mempunyai empat fase, yaitu:

###### a. Fase *Inception*

Tahap dimana kita memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan



dibuat (*requirements*). Hasil dari fase ini berupa Analisa pemodelan proses bisnis *as-is* dan *to-be*, analisis persyaratan, dan pemodelan *usecase*.

b. *Fase Elaboration*

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini lebih pada analisis dan desain sistem.

c. *Fase Construction*

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*.

d. *Fase Transition*

Tahap dimana kita deployment atau Instalasi sistem agar dapat dimengerti oleh *user*. Aktifitas pada tahap ini termasuk pada pelatihan user dan pemeliharaan.

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.

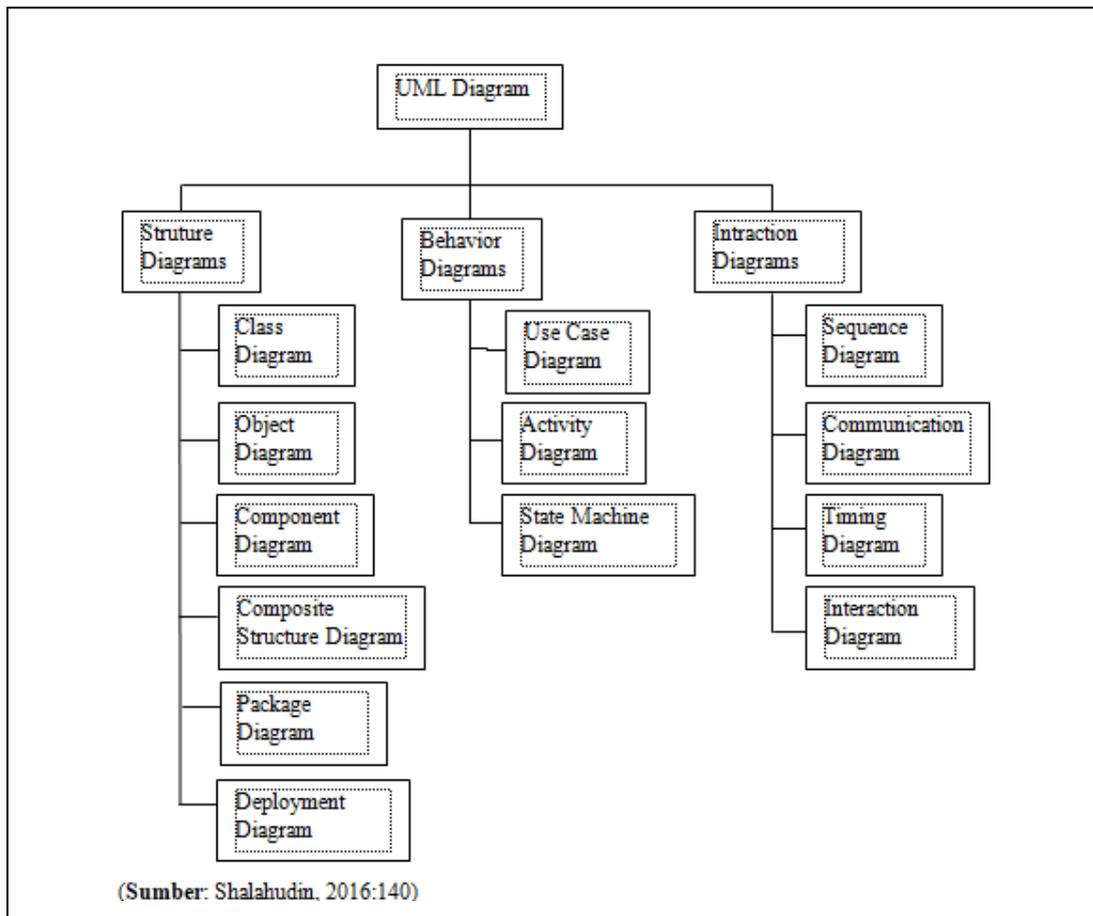
## 2.2. Teori Khusus

### 2.2.1 Pengertian UML (*Unified Modeling Language*)

A.S, Rosa dan M. Shalahuddin (2016:133), menyatakan bahwa UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Hal ini disebabkan karena UML menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

### 2.2.2 Macam-macam diagram *Unified Modeling Language (UML)*

Menurut A.S, Rosa dan M. Shalahuddin (2016:140), Pada *UML* terdapat 13 buah diagram yang selanjutnya dikelompokkan menjadi 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah ini:



**Gambar 2.2** Macam-macam diagram *Unified Modeling Language (UML)*

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan struktur statis dari sebuah sistem yang dimodelkan.

b. *Behavior Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan rangkaian perubahan yang terjadi pada sebuah sistem.



### c. *Interaction Diagram*

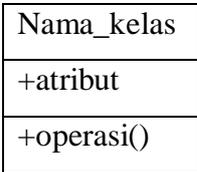
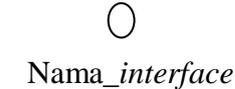
Adalah sekumpulan diagram yang digunakan untuk menggambarkan sebuah interaksi sistem dengan sistem lain nya ataupun interaksi antar subsistem pada suatu sistem.

### 2.2.3. Pengertian *Class Diagram*

A.S, Rosa dan M. Shalahuddin (2016:133), menyatakan bahwa 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 atau operasi.

- a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

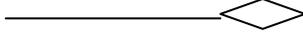
**Tabel 2.1** Simbol-simbol dalam *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur system
2.		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.		Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>




---

**Lanjut Tabel 2.1** Simbol-simbol dalam *Class Diagram*

No.	Simbol	Deskripsi
5.		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
6.		Relasi antarkelas dengan makna kebergantungan antarkelas
7.		Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

(Sumber: Shalahudin, 2016:146-147)

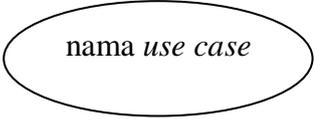
#### 2.2.4. Pengertian *Use Case Diagram*

Menurut A.S, Rosa dan M. Shalahuddin (2016:133), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

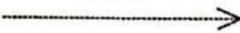
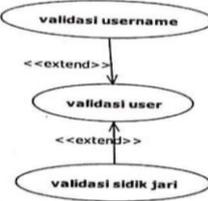
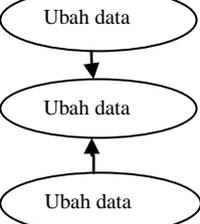
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. 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.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

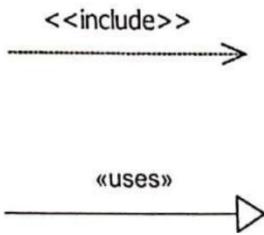
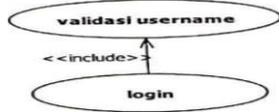
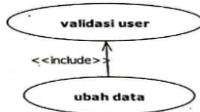
Tabel 2.2 Simbol-simbol dalam *Use Case*

No.	Simbol	Deskripsi
1.	<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal <i>frase</i> nama <i>use case</i>.</p>
2.	<p>Aktor/<i>actor</i></p> 	<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 <i>frase</i> nama aktor.</p>
3.	<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

Lanjutan Tabel 2.2 Simbol-simbol dalam Use Case

No.	Simbol	Deskripsi
4.	<p>Ekstensi/<i>extend</i></p> <p style="text-align: center;">&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun 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, misal;</p>  <p>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>Generalisasi/<i>generalization</i></p> 	<p>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>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>



Lanjutan 2.2 Simbol-simbol dalam Use Case		
No.	Simbol	Deskripsi
6.	Menggunakan/include/uses  	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di use case:</p> <ol style="list-style-type: none"> <li>1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut:           <div style="text-align: center;">  </div> </li> <li>2. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan selalu dijalankan sebelum <i>use case</i> tambahan dijalankan, missal pada kasus berikut:           <div style="text-align: center;">  </div> </li> </ol> <p>Kedua interpretasi di atas dapat dianut salah satu keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Shalahudin, 2016:156-158)



### 2.2.5. Pengertian *Activity Diagram*

A.S, Rosa dan M. Shalahuddin (2016:133), menyatakan bahwa *Activity Diagram* atau Diagram Aktivitas 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.

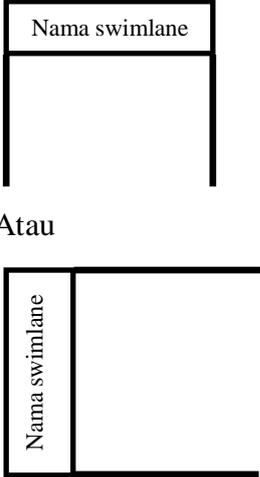
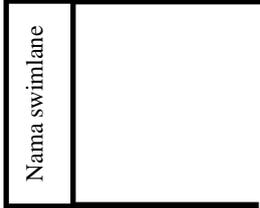
Diagra aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

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

**Tabel 2.3** Simbol-simbol dalam *Activity Diagram*

No.	Simbol	Deskripsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

Lanjutan Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

No.	Simbol	Deskripsi
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
6.	Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Sumber: Shalahudin, 2016:162-163)

### 2.2.6. Pengertian Sequence Diagram

Menurut A.S, Rosa dan M. Shalahuddin (2016:133), Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstantiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

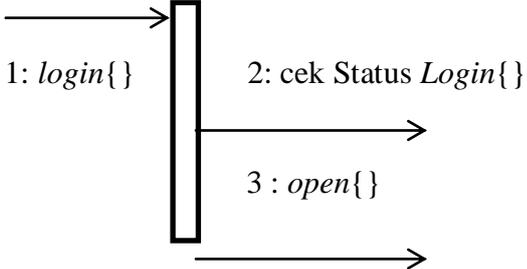


Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

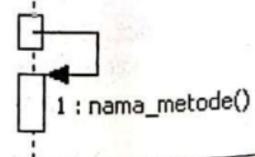
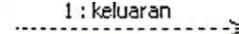
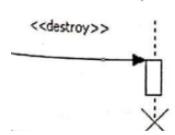
**Tabel 2.4** Simbol-simbol dalam *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Nama aktor</p> <p>Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Namaaktor</div> <p>tanpa waktu aktif</p>	<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 <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i></p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama objek</u>  <u>:nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

No.	Simbol	Deskripsi
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya  Maka cek Status <i>Login</i> { } dan <i>open</i> { } dilakukan di dalam metode <i>login</i> { } Aktor tidak memiliki waktu aktif
5.	Pesan tipe <i>create</i> << <i>create</i> >> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada arah objek yang dibuat
6.	Pesan tipe <i>call</i> 1 :nama_metode{ }	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

No.	Simbol	Deskripsi
7.		 <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>
8.	<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
9.	<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	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
10.	<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

(Sumber: Shalahudin, 2016:165-167)



### **2.3. Teori Judul**

#### **2.3.1. Pengertian Aplikasi**

Menurut Koyuko, dkk (2016:2), aplikasi merupakan perangkat lunak yang berfungsi untuk melakukan berbagai pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan OpenOffice yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

#### **2.3.2. Pengertian Persediaan**

Menurut Indrajit dan Djokopranoto (Dikutip Yunita, Putri dan Dwi Maya Sari, 2020:2), “Persediaan merupakan serangkaian kebijaksanaan dan pengendalian yang memonitor tingkat persediaan dan menentukan tingkat persediaan yang harus dijaga, kapan persediaan harus diisi, dan berapa besar pesanan yang harus dilakukan”.

#### **2.3.3. Pengertian Obat**

Menurut Ariani (Dikutip Yunita, Putri dan Dwi Maya Sari, 2020:2), “Obat dalam bahasa Inggris disebut drug sedangkan dalam bahasa Prancis disebut droque yang berarti rempah kering. Menurut bagian Farmakologi obat didefinisikan sebagai substansi yang digunakan untuk pencegahan, diagnosa dan pengobatan pada manusia maupun binatang”.



### **2.3.4. Pengertian Aplikasi Persediaan Obat Masuk dan Keluar Pada Puskesmas Merdeka Palembang**

Dimana sistem ini digunakan untuk melakukan pencatatan persediaan obat masuk dan keluar dalam hal ini secara tidak langsung aplikasi juga dapat menghitung obat yang telah digunakan. Pada penelitian ini akan dibangun aplikasi yang dapat digunakan untuk memudahkan pegawai dalam pencatatan obat masuk dan keluar. Aplikasi yang dibuat haruslah dapat diakses melalui web sehingga dapat diakses secara langsung oleh pegawai.

## **2.4. Teori Program**

### **2.4.1. Pengertian Basis Data**

Fathansyah (2018:2), Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

### **2.4.2. Pengertian MySQL**

Menurut Firman, A dkk (2016:30), *MySQL* adalah salah satu *database server open source* yang cukup populer keberadaannya saat ini. Dengan berbagai keunggulan yang dimiliki, membuat *software database* ini banyak digunakan oleh praktisi untuk membangun suatu *project*. Dengan adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh *MySQL*, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat dengan mudah mengakses basis data *MySQL*.



### 2.4.3. Pengertian *CodeIgniter*

Menurut Pujohardiyanto, A dan Syahbaniar Rofiah (2019:104), “*Codeigniter* adalah *framework* pemrograman berbasis web dan *Codeigniter* digunakan untuk memudahkan pembuatan website”.

Adapun menurut Jafar, dkk. (2020:3), “*CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua *code program* dari awal”.

Sehingga dapat disimpulkan *CodeIgniter* adalah *framework open source* yang memudahkan dalam membuat aplikasi tanpa harus membuat dari awal.

### 2.4.4. Pengertian Bahasa Pemrograman

Menurut Gusti, N, dkk (2018:21), Bahasa pemrograman merupakan sebuah alat komunikasi antara manusia dengan perangkat komputer. Saat ini perkembangan teknologi. Bahasa pemrograman telah menjadi sebuah primadona tersendiri. Bahasa dibagi menjadi pemrograman tiga bagian, bahasa pemrograman tingkat rendah (low level programming language), bahasa pemrograman tingkat menengah (middle level programming language), bahasa pemrograman tingkat tinggi (high level programming language). Saat ini yang paling banyak mengalami perkembangan adalah bahasa pemrograman tingkat tinggi. Tercatat sejak kemunculan bahasa C yang dibuat oleh Dennis Ritchie, yang kemudian menjadi pondasi dari beberapa bahasa pemrograman seperti Java (James Gosling), C++ (Bjarne Stroustrup), PHP (Rasmus Lerdorf), dan sebagainya.

### 2.4.5. Pengertian HTML

Abdulloh (2018:7), *HTML* merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *web* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*. *HTML* berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai layout yang diinginkan.



*HTML* biasanya disimpan dalam sebuah file berekstensi *.html*. Untuk mengetikkan skrip *HTML* dapat menggunakan *text editor* seperti Notepad sebagai bentuk paling sederhana atau *text editor* khusus yang dapat mengenali setiap unsur skrip *HTML* dan menampilkannya dengan warna yang berbeda sehingga mudah dibaca, seperti Notepad++, *Sublime Text*, dan masih banyak lagi aplikasi lain yang sejenis.

#### 2.4.6. Pengertian PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi *HTML*. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari.

PHP merupakan bahasa scripting *server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server*lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

Adapun pengertian lain PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode *HTML*".

Menurut Firman, A dkk (2016:30), "PHP atau lebih resminya (*Hypertext Preprocessor*) adalah skrip bersifat *server-side* yang ditambahkan ke dalam *HTML*". Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan *internet*, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*.

Selanjutnya *webserver* akan mencari berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera



menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh *user* adalah halaman yang mengandung *script* PHP? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe file yang diminta *user*. Jika tipe file yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut.

Apabila dalam file tersebut tidak mengandung *script* PHP, permintaan *user* akan langsung ditampilkan ke *browser*, namun jika dalam file tersebut mengandung *script* PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script-script* PHP dan mengolah *script* tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke *browser user*.

#### **2.4.7. Pengertian Web Server**

Menurut Prayitno dan Safitri (2015:2), Menerangkan bahwa pengertian *web server* adalah “*Server Web (Web Server)* merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”.

#### **2.4.8. Pengertian Xampp**

Menurut Prayitno dan Safitri (2015:2), “XAMPP adalah salah satu paket instalasi apache, PHP, dan MySQL secara *instant* yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasi-kannya secara otomatis untuk anda. XAMPP adalah sebuah webserver (Kartini dan Riya Widayanti. 2020:72).



#### **2.4.9. Pengertian phpMyAdmin**

Menurut Firliana, R, dkk. (2019:46), PhpMyadmin adalah sebuah software yang berbentuk seperti halaman situs yang terdapat pada web server. Fungsi dari halaman ini adalah, sebagai pengendali database MySQL. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman PhpMyadmin.