



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Komputer

Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Pratama (2014:16), “Komputer merupakan mesin elektronik untuk perhitungan cepat yang mampu menerima inputan data digital, melakukan pemrosesan di memori sesuai dengan aplikasi yang digunakan, untuk kemudian dihasilkan menjadi sebuah informasi”.

2.1.2 Pengelolaan Data

Yogiyanto:2003 (dalam iken surya buana dan yonal supid,dkk:2017), “Pengelolaan data adalah pengolahan pikiran dengan bantuan tangan atau peralatan yang mengikuti serangkaian langkah, perumusan atau pola tertentu untuk mengubah data, sehingga data tersebut, susunan, sifat atau isinya menjadi lebih berguna. Pengelolaan data di atas memberikan gambaran bahwa pengelolaan data merupakan aktifitas dalam mengolah data baik berupa bentuk, susunan, sifat atau isinya menjadi informasi yang bermanfaat dalam mencapai tujuan”.

Jurnal Agung Wahana dan Asep Ririh Riswaya:2014 “Pengelolaan data dengan menggunakan komputer dikenal dengan sebutan pengolahan data elektronik (PDE). Data adalah kumpulan kejadian yang diambil dari suatu fakta dan data dapat berupa angka-angka, huruf-huruf atau simbol-simbol khusus atau gabungan dari kesemuanya. Suatu data mentah merupakan data yang belum dapat memberi informasi yang banyak dan lengkap, sehingga perlu diolah terlebih dahulu”.



2.2 Teori Judul

2.2.1 Aplikasi

Kadir (2017:3), “Perangkat lunak adalah kumpulan intruksi yang ditujukan kepada komputer. Istilah program dan aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa intruksi-intruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi”.

Juansyah (2015:2), “Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju”.

2.2.2 Data

Rusmawan (2019:34), “Data adalah catatan atau kumpulan fakta. Data merupakan bentuk jamak dari datum, berasal dari bahasa latin yang berarti “sesuatu yang diberikan”.

Indrajani (2015:69), “Data merupakan fakta mentah tentang orang, tempat, kejadian, dan apapun yang penting bagi perusahaan, dimana data itu sendiri tidak memiliki arti”.

2.2.3 Stok

kamus KBBI:2018,”Stok adalah Persediaan keperluan untuk dagang atau perbekalan”.

Raymond Mcleod,Jr:1998 (dalam Hanik Mujiati dan Sukajadi:1979)
“Stok atau persediaan adalah sebagai suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam periode usaha yang normal.



2.2.4 Peralatan

Kamus KBBI:2018, "Peralatan adalah suatu benda yang dipakai untuk mengerjakan sesuatu".

Kamus KBBI:2018, "Alat adalah barang yang dipakai untuk mengerjakan sesuatu.

2.2.5 Pelaporan

Kamus KBBI:2018, "Pelaporan adalah suatu pembetulan dari hasil perbuatan atau usaha memperbaiki".

Kamus KBBI:2018, "Laporan adalah segala sesuatu yang dilaporkan.

2.2.6 Perbaikan

Kamus KBBI:2018, "Perbaikan adalah suatu pembetulan dari hasil perbuatan atau usaha memperbaiki".

2.2.7 Komputer

Puspitosari (2013:1), "Komputer adalah sebuah alat yang digunakan untuk mengolah data menurut perintah yang telah dirumuskan. Komputer dapat didefinisikan sebagai sekumpulan alat elektronik yang saling terkoordinasi satu sama lain sehingga dapat menerima data, kemudian mengolah data, dan pada akhirnya akan menghasilkan suatu keluaran yang berupa informasi (Input > Proses > Ouput)".

Kadir (2013:2), "Komputer merupakan peralatan elektronik yang biasa dipakai orang untuk membantu pelaksanaan pekerjaan".

2.2.8 Website

Abdulloh (2018:1), "*Website* atau disingkat *web* dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet".

Sidik (2017:1), "Situs *Web (web site)* awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan surfer



(sebutan bagi pemakai komputer yang melakukan penelusuran informasi di Internet) untuk mendapatkan informasi, dengan cukup mengklik suatu link berupa teks atau gambar, maka informasi dari teks atau gambar akan ditampilkan secara lebih rinci (*detail*)”.

2.2.9 Metode PIECES

Menurut Gordon B Davis dan Margrethe H. Olson (dalam Rahmat Tullah dan Muhammad Iqbal Hanafri: 2014) “*PIECES Framework* adalah kerangka yang dipakai untuk mengklasifikasi-kan suatu *problem, opportunities*, dan *directives* yang terdapat pada bagian *scope definition* analisa dan perancangan sistem.

Menurut Whitten dan Bentley:2007 (dalam Yuli Asbar dan Mochamad Ari Saptari:2017) “mengembangkan suatu kerangka kerja yang berguna untuk mengklasifikasikan permasalahan yang disebut *PIECES (Performance Information Economics Efficiency Service)*. Kerangka kerja *PIECES* dapat dijabarkan sebagai berikut :

1. *Performance*

Produksi jumlah kerja selama periode waktu tertentu Waktu respon penundaan rata-rata antara transaksi atau permintaan dengan respons ke transaksi atau permintaan tersebut

2. *Information Input* :

- (1) Kurangnya informasi.
- (2) Kurangnya informasi yang diperlukan.
- (3) Kurangnya informasi yang relevan.
- (4) Terlalu banyak informasi – “kelebihan informasi.
- (5) Infomasi yang tidak akurat.
- (6) Infomasi yang tidak tepat waktunya untuk penggunaan selanjutnya.

Penyimpanan data-data disimpan secara berlebihan dalam banyak file dan/atau database Item-item data sama memiliki nilai-nilai berbeda dalam file-file berbeda (integrasi data yang jelek). Data tersimpan tidak akurat Data tidak diorganisasikan dengan baik. Data tidak fleksibel – tidak



mudah untuk memenuhi kebutuhan informasi baru dari data tersimpan
Data tidak dapat diakses.

3. *Economy*

Biaya tidak dapat dilacak ke sumber Biaya terlalu tinggi Keuntungan
Pasar-pasar baru dapat dieksplorasi Pemasaran saat ini dapat diperbaiki

2. *Control*

Keamanan/control terlalu lemah Input data tidak diedit dengan cukup
Kejahatan (misalnya, penggelapan atau pencurian) terhadap data

3. *Efficiency*

Orang, mesin, atau komputer membuang waktu, Data secara berlebihan di-
input atau disalin. Data secara berlebihan diproses Informasi secara
berlebihan dihasilkan Orang, mesin, atau komputer membuang material
dan persediaan

4. *Services*

Sistem menghasilkan produk yang tidak akurat Sistem menghasilkan
produk yang tidak konsisten Sistem menghasilkan produk yang tidak
dapat dipercaya, Sistem tidak mudah dipelajari. Analisis *PIECES*
(*Performance, Information, Economy, Control, Eficiency, dan Service*)
merupakan teknik untuk mengidentifikasi dan memecahkan permasalahan
yang terjadi pada kualitas pelayanan. Dari analisis ini akan menghasilkan
identifikasi masalah utama dari suatu kinerja serta memberikan solusi dari
permasalahan tersebut.

2.2.6 Aplikasi Pengelolaan Data Stok Peralatan dan Pelaporan Perbaikan Komputer pada Badan Pengelolaan Keuangan dan Aset Daerah (BPKAD) menggunakan Metode *Pieces*

Aplikasi Pengelolaan Data Stok dan Pelaporan Perbaikan Komputer ini merupakan aplikasi yang digunakan untuk melakukan pengajuan proposal perbaikan komputer dari instansi badan pemerintahan contohnya BPPD ke bagian admin BPKAD dan bagian IT melakukan pengecekan data stok peralatan



pada Badan Pengelolaan Keuangan dan Aset Daerah dengan menggunakan metode *Pieces*

Badan Pengelolaan Keuangan dan Aset Daerah (BPKAD) merupakan instansi pemerintahan yang memiliki unsur pelaksana urusan pemerintahan bergerak di bidang pengelolaan keuangan dan aset daerah, dipimpin oleh seorang kepala badan yang berkedudukan di bawah dan bertanggung jawab kepada Walikota melalui Sekretariat Daerah. Badan Pengelolaan Keuangan dan Aset Daerah (BPKAD) ayat (1) mempunyai tugas membantu walikota melaksanakan urusan pemerintahan di bidang pengelolaan keuangan dan aset daerah beserta tugas pembantuan dari masing-masing bidang .

2.3 Teori Khusus

2.3.1 Unified Modeling Language

2.3.1.2. Pengenalan *Unified Modeling Language* (UML)

Unified Modeling Language (UML) 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. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2018:133).

2.3.1.3. Sejarah *UML*

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman *Smalltalk* pada awal 1980-an



yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, *Eiffel*, dan CLOS.

Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi- metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep *Object Modeling Technique (OMT)* dari *Rumbaugh* dan *Booch* (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari *Rebecca Wirfs-Brock* (1990), konsep pemikiran *Ivar Jacobson*, dan beberapa konsep lainnya dimana *James R. Rumbaugh*, *Grady Booch*, dan *Ivar Jacobson* bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada tahun 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *UML* diakomodasi oleh *OMG* sehingga sampai saat ini *UML* telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (*Rosa A.S dan M. Shalahudin, 2014:138*).

2.3.1.4. Diagram UML

Rosa A.S dan M. Shalahudin (2018:140), pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.



2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

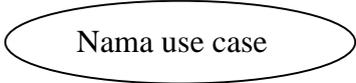
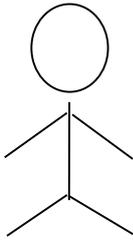
2.3.1.5. Use Case Diagram

Rosa dan M. Shalahudin (2018:155), *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 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.



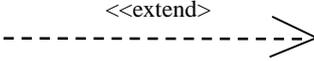
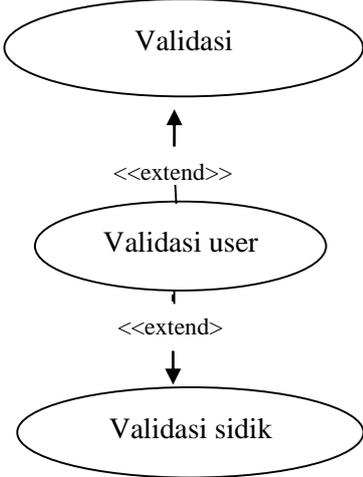
Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
1.	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan Dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.	<i>Aktor/actor</i> 	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 jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda di awal frasa nama aktor.
3.	<i>Assosiasi/association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

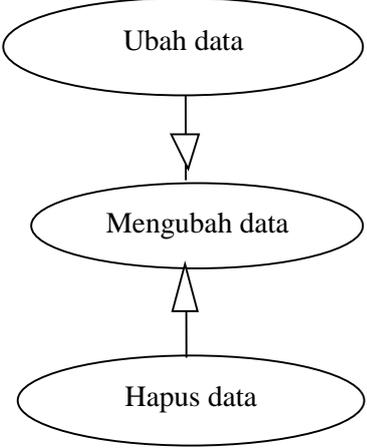
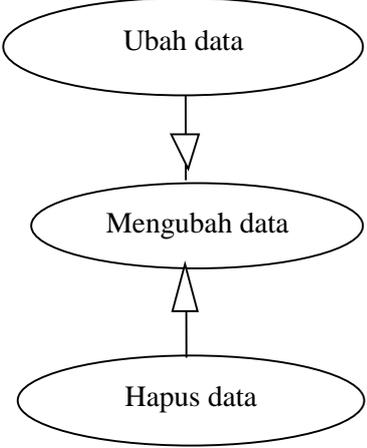
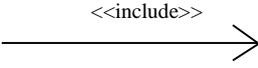
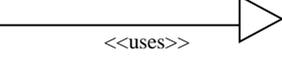
Lanjutan Tabel 2.1. Simbol-simbol diagram *use case*



No.	Simbol	Deskripsi
4.	Ekstensi / <i>extend</i> 	<p>Relasi <i>use case</i> tambahan ke— sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use</i> <i>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>

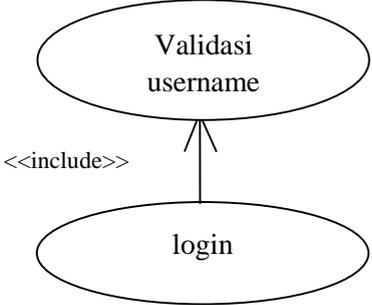
Lanjutan Tabel 2.1. Simbol-simbol diagram *use case*



No.	Simbol	Deskripsi
5.	<p data-bbox="440 371 815 405">Generalisasi / <i>generalization</i></p>  	<p data-bbox="927 371 1414 622">Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p data-bbox="927 1234 1414 1317">arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum.</p>
6.	<p data-bbox="440 1368 815 1402">Menggunakan / <i>include</i> / <i>uses</i></p>  	<p data-bbox="927 1368 1430 1619">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>



Lanjutan Tabel 2.1. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
		<p>ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none">• <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan nasal pada kasus berikut  <pre>graph BT; login([login]) -- "<<include>>" --> validasi([Validasi username]);</pre> <ul style="list-style-type: none">• <i>include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:



Lanjutan Tabel 2.1. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
		<pre>graph TD; U1(Validasi username); U2(Ubah data); U2 -- <<include>> --> U1;</pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Rosa A.S dan M. Shalahudin (2018:156)

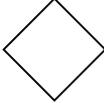
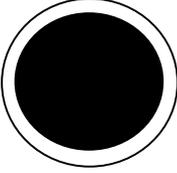
2.4.1.6 Activity Diagram

Rosa dan M. Shalahudin (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 di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

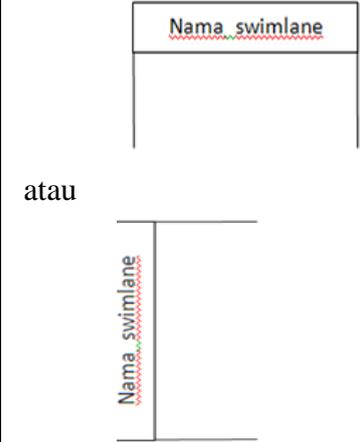
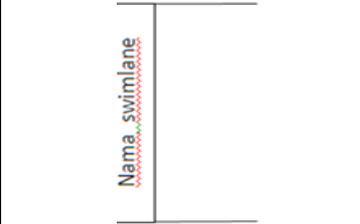


Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2. 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 dimana 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 oleh sistem, sebuah diagram Aktivitas memiliki sebuah status akhir.

Tabel 2.2. Lanjutan simbol-simbol *activity diagram*

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

Sumber : Rosa A.S dan M. Shalahudin (2018:162)

2.4.1.7 Class Diagram

Rosa dan M. Shalahudin (2014:141), 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 *method* atau operasi. Berikut penjelasan atribut dan *method* :

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



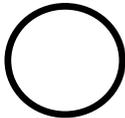
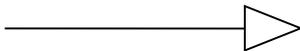
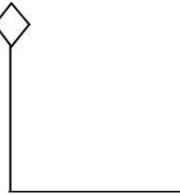
Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.3. Simbol-simbol *class diagram*

No.	Simbol	Deskripsi			
1.	Kelas <table border="1" data-bbox="563 618 863 808"><tr><td data-bbox="563 618 863 689">nama_kelas</td></tr><tr><td data-bbox="563 689 863 748">+atribut</td></tr><tr><td data-bbox="563 748 863 808">+operasi()</td></tr></table>	nama_kelas	+atribut	+operasi()	Kelas pada struktur sistem
nama_kelas					
+atribut					
+operasi()					

Tabel 2.3. Lanjutan Simbol-simbol *class diagram*



No.	Simbol	Deskripsi
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam perograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna 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 antarkelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

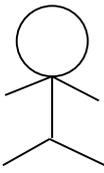
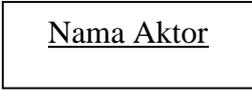
Sumber : Rosa A.S dan M. Shalahudin (2018:146)

2.4.1.8 Sequence Diagram

Rosa dan M. Shalahudin (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *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 diinstansiasi 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 interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

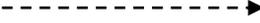
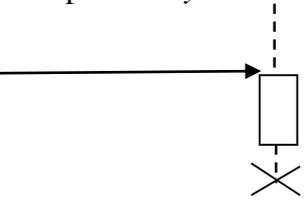
Tabel 2.5. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
1.	Aktor  Atau  tanpa waktu aktif	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 aktor.
2.	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek.

Tabel 2.3. Lanjutan Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
3.	Objek <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> Nama objek : nama kelas </div>	Menyatakan objek yang berinteraksi pesan.
4.	Waktu Aktif <div style="border: 1px solid black; width: 40px; height: 100px; margin: 10px auto;"></div>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan, yang dilakukan di dalamnya, misalnya
5.	Pesan tipe <i>create</i> <div style="text-align: center; margin: 10px auto;"> <<create>> </div>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> <div style="text-align: center; margin: 10px auto;"> 1 : nama_metode() </div>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,

Tabel 2.3. Lanjutan Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
		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.
7.	Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i>  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.
9.	Pesan tipe <i>destroy</i> 	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 : Rosa A.S dan M. Shalahudin (2018:165)



2.4 Teori Program

2.4.1 *Bootstrap*

Bootstrap merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah *web* menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik (Abdulloh,2018:261).

2.4.2 CSS (*Cascading Style Sheet*)

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *web* yang berfungsi mengatur elemen HTML dengan berbagai *property* yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan (Abdulloh,2018:45).

2.4.3 *Database*

Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Hidayatullah dan Kawistara, 2014:147).

2.4.4 HTML (*Hypertext Markup Language*)

Hypertext Markup Language (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan HTML yaitu (Hidayatullah dan Kawistara, 2014:13).

1. Mengatur tampilan dari halaman *web* dan isinya.
 2. Membuat tabel dalam halamn *web*.
 3. Mempublikasikan halaman *web* secara *online*.
 4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
 5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*.
 6. Menampilkan area gambar (*canvas*) di *browser*.
-



2.4.5 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi *web*. Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah (Hidayatullah dan Kawistara, 2014: 180).

2.4.6 PHP (*Hypertext Preprocessor*)

PHP *Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server* (Hidayatullah dan Kawistara, 2014: 231).

2.4.7 XAMPP

Nugroho (2013 : 1) menjelaskan, XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legas.”

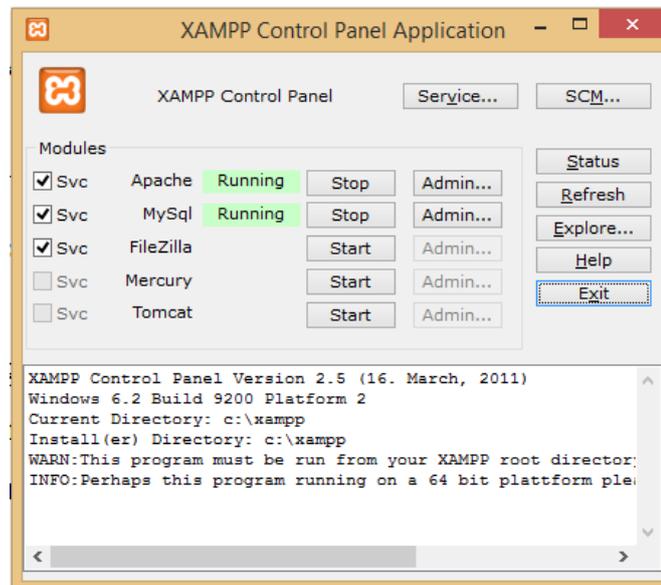
Dibawah folder utama *xampp*, terdapat beberapa folder penting yang perlu diketahui. Penjelasan fungsinya sebagai berikut

Tabel 2.4 Folder-Folder Penting Dalam XAMPP

No	Nama Folder	Fungsi
1.	Apache	Folder utama dari <i>Apache Webserver</i> .
2.	Htdocs	Folder utama untuk menyimpan data-data latihan <i>web</i> , baik PHP maupun HTML biasa. Pada folder ini, anda dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua folder dan file program di htdocs bisa diakses dengan



		mengetikkan alamat http://localhost/ di browser.
3.	Manual	Berisi subfolder yang di dalam terdapat manual program dan <i>database</i> , termasuk manual PHP dan MYSQL.
4.	Mysql	Folder utama untuk <i>database</i> MYSQL server. di dalamnya terdapat subfolder data (lengkapnya: C:\xampp\mysql\data) untuk merekam semua nama <i>database</i> , serta subfolder bin yang berisi <i>tools</i> klien dan server MYSQL.
5.	Php	Folder utama untuk program PHP.



Gambar 2.2 Tampilan XAMPP

2.5 Metode *Rapid Application Development (RAD)*

Menurut Rosa A.S. dan M. Shalahuddin (2018:34) “Metode pengembangan sistem yang dipakai yakni Metode *Rapid Application Development*. Metode



Rapid Application Development merupakan model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang singkat. RAD juga merupakan adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak

Jika kebutuhan perangkat lunak dipahami dengan baik dan lingkup perangkat lunak dibatasi dengan baik sehingga tim dapat menyelesaikan pembuatan perangkat lunak dengan waktu yang pendek. Menurut Kendall (2010), *RAD* adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. *RAD* bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi.

Terdapat tiga fase dalam *RAD* yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan dan penerapan. Adapun ketiga fase tersebut adalah *requirements planning* (perencanaan syarat-syarat), *RAD design workshop* (*workshop* desain *RAD*), dan *implementation* (implementasi). Sesuai dengan metodologi *RAD* menurut Kendall (2010), berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi.

1. *Requirements Planning* (Perencanaan Syarat-Syarat)

Dalam *fase* ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. *Orientasi* dalam *fase* ini adalah menyelesaikan masalah-masalah perusahaan.

2. *RAD Design Workshop* (*Workshop* Desain *RAD*)

Fase ini adalah *fase* untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan *representasi visual desain* dan pola kerja



kepada pengguna. *Workshop desain* ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon *prototipe* yang ada dan menganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Apabila seorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi.

3. *Implementation* (Implementasi)

Pada fase implementasi ini, menganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi.

