



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Paramytha (2016: 4) menyatakan bahwa komputer berasal dari bahasa latin yaitu *Computer* yang berarti menghitung (*to compute* atau *to reckon*). Kata komputer pada awalnya dipergunakan untuk menggambarkan pekerjaan orang yang melakukan perhitungan aritmatika dengan atau tanpa alat bantu.

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

Berdasarkan pengertian diatas dapat disimpulkan bahwa komputer adalah alat elektronik yang dipergunakan untuk menggambarkan pekerjaan orang yang melakukan perhitungan aritmatika dengan atau tanpa alat bantu.

2.1.2. Pengertian *Smartphone*

Muflih (2017: 13) menyatakan “*smartphone* adalah operasi aplikasi berbasis internet. Dengan demikian, penggunaan *smartphone* secara fungsional harus menggunakan internet.”

Fitri (2017: 1) menyatakan “*smartphone* merupakan salah satu hasil dari perkembangan teknologi yang memiliki kemampuan canggih.”

Berdasarkan pengertian diatas dapat disimpulkan bahwa *smartphone* adalah salah satu hasil dari perkembangan teknologi yang memiliki kemampuan canggih dengan operasi aplikasi berbasis internet.

2.1.3. Pengertian Perangkat Lunak

Rosa (2018: 2) menyatakan “perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).”

Kadir (2017: 2) menyatakan “perangkat lunak adalah intruksi-intruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai.”



Berdasarkan pengertian diatas dapat disimpulkan bahwa perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak untuk melaksanakan tugas sesuai kehendak pemakai.

2.1.4. Pengertian Program

Menurut Kamus Besar Bahasa Indonesia (<https://kbbi.web.id/data>, diunduh pada 29 April 2019, pukul 15.22 WIB) menyatakan bahwa program adalah urutan perintah yang diberikan komputer untuk membuat fungsi atau tugas tertentu.

Kadir (2017: 4) menyatakan bahwa istilah program biasa digunakan untuk menyatakan karya para profesional yang berupa intruksi-intruksi untuk mengendalikan komputer.

Berdasarkan pengertian diatas dapat disimpulkan bahwa program adalah urutan perintah atau intruksi-intruksi untuk mengendalikan komputer untuk membuat fungsi atau tugas tertentu yang merupakan karya para profesional .

2.1.5. Basis Data (*Database*)

2.1.5.1. Pengertian Basis Data (*Database*)

Kristanto (2018: 79) menyatakan “basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi.”

Rosa (2018: 43) menyatakan “basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.”

Dari kedua pendapat diatas dapat disimpulkan bahwa basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis agar dapat diakses dengan mudah dan cepat.

2.1.5.2. Komponen Basis Data

Setyaningrum (2013: 2—4) mengatakan bahwa sebuah basis data tersusun atas beberapa komponen, yaitu sebagai berikut:

1. Data

Data disimpan secara terintegrasi (*integrated*), artinya basis data merupakan gabungan dari berbagai macam file aplikasi yang berbeda yang disusun dengan menghilangkan bagian yang rangkap. Sebagai alat penghubung digunakan kunci (*key*).



Data dipakai secara bersama-sama dalam waktu yang bersamaan oleh pemakai untuk aplikasi yang berbeda. Komponen data dibagi menjadi 3 (tiga), yaitu:

- a. Data Operasional, yaitu data yang disimpan dalam basis data baik itu berupa data master maupun data transaksi.
- b. Data Masukan (*Data input*), data dari luar sistem yang dimasukkan melalui peralatan input yang dapat mengubah data operasional.
- c. Data Keluaran (*Data output*), yaitu hasil data laporan melalui peralatan output sebagai hasil proses dari dalam sebuah sistem yang mengakses data operasional.

2. *Hardware* (Perangkat Keras)

Perangkat keras yang dibutuhkan untuk pengolahan basis data, berupa komputer beserta kelengkapannya seperti *monitor, keyboard, mouse* dll.

3. *Software* (Perangkat Lunak)

Aplikasi yang digunakan sebagai *interface* (antar muka) antara pemakai dengan data fisik basis data. Misalnya: program aplikasi maupun *DBMS*.

4. *User* (Pemakai)

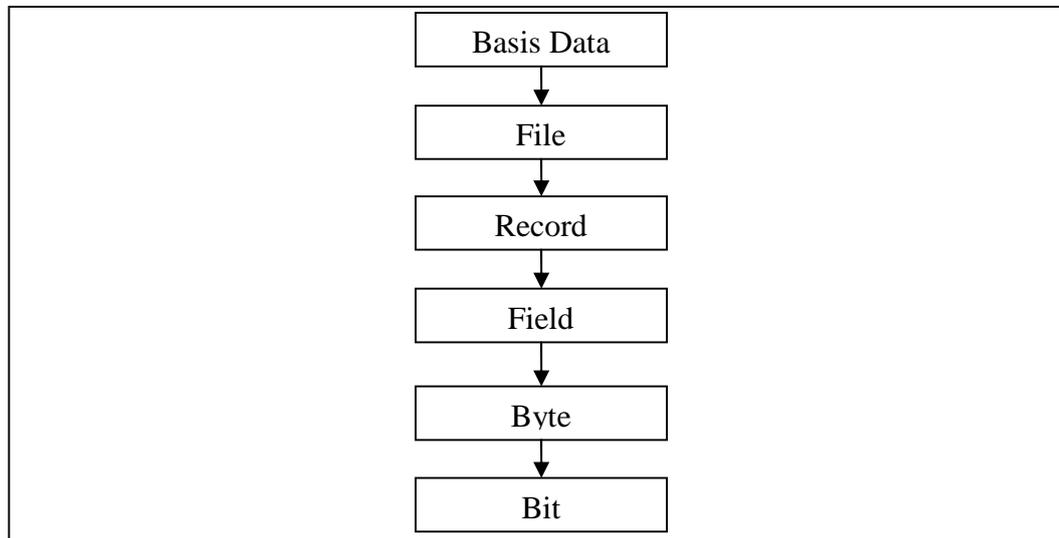
User basis data dibagi menjadi 4 (empat), yaitu:

- a. Pemakai umum (*End User/ Naive User*), yaitu pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program permanen yang telah disediakan sebelumnya.
- b. Pemakai khusus (*Sophisticated User*), yaitu pemakai yang berinteraksi dengan basis data tidak melalui program melainkan menggunakan bahasa *query*.
- c. Pemrograman Aplikasi (*Application Programmer*), yaitu pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language (DML)*, yang disertakan (*embedded*) dalam program yang dituliskan pada bahasa pemrograman induk.
- d. *Data Base Administrator (DBA)*, yaitu tenaga ahli yang bertugas untuk mengontrol sistem basis data secara keseluruhan, meramalkan kebutuhan akan sistem basis data, merencanakan serta mengaturnya.



5. Hierarki Data

Dalam basis data, urutan atau hierarki data sangatlah penting. Adapun struktur hierarki data dalam basis data tertinggi sampai terendah sebagai berikut:



(Sumber: Setyaningrum 2013: 4)

Gambar 2.1 Hierarki Data

Keterangan:

- Basis Data : Kumpulan dari file yang membentuk sebuah basis data.
- File* atau tabel/ berkas adalah kumpulan dari *record* yang saling berkaitan dan memiliki *format field* yang sama, namun berbeda isi datanya.
- Record* atau baris adalah gabungan dari sejumlah elemen data yang saling berkaitan.
- Field* atau elemen data atau atribut adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna.
- Bit* adalah bagian terkecil dari data secara keseluruhan, yaitu berupa karakter *ASCII* nol atau satu merupakan komponen *byte*.



2.2. Teori Khusus

2.2.1. Metode Pengembangan

2.2.1.1. Pengertian *Rational Unified Process (RUP)*

Rosa (2018: 125) menyatakan “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus terhadap arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*).”

2.2.1.2. Fase RUP

Rosa (2018: 129—131) menyatakan bahwa RUP memiliki 4 (empat) buah tahap atau fase yang dilakukan pula secara iteratif, yaitu sebagai berikut:

1. *Inception* (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut ini 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

2. *Elaboration* (Perluasan/ Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*)

3. *Construction* (Kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadu syarat dari *Initial Operational Capability Milestone* atau batas/ tonggak kemampuan operasional.



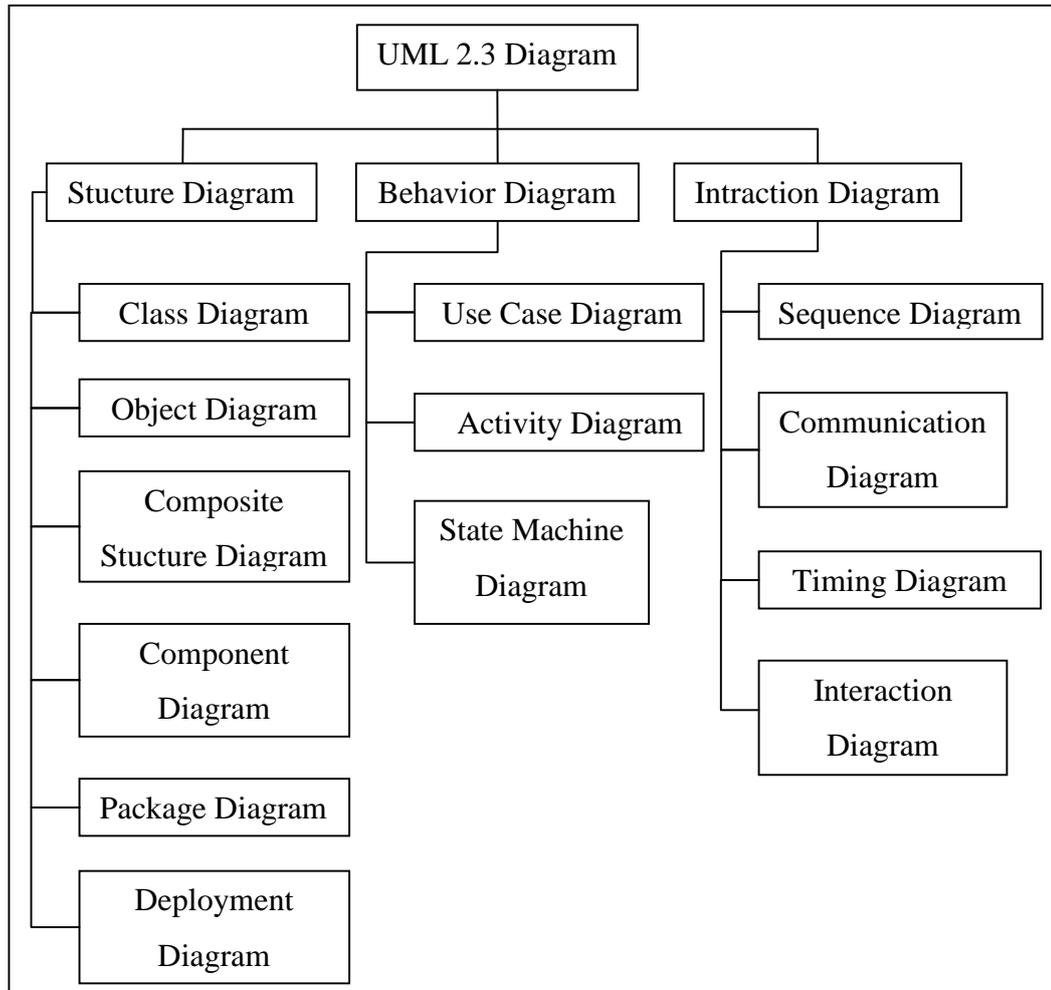
4. Trasiotion (Trasnsisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari Initial Operational Capability Milestone atau batas/ tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.



2.2.2. Pengertian UML (*Unified Modeling Language*)

Rosa (2018: 140—141) menyatakan bahwa pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah:



(Sumber: Rosa 2018: 140)

Gambar 2.2 Macam-macam Diagram UML



Penjelasan singkat dari pembagian kategori pada diagram UML, yaitu:

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

2.2.3. Use Case Diagram

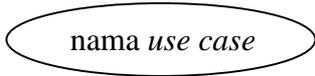
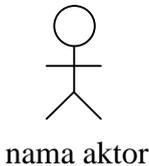
2.2.3.1. Pengertian Use Case Diagram

Rosa (2018: 155) menyatakan “*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.”

2.2.3.2. Simbol-simbol pada Use Case Diagram

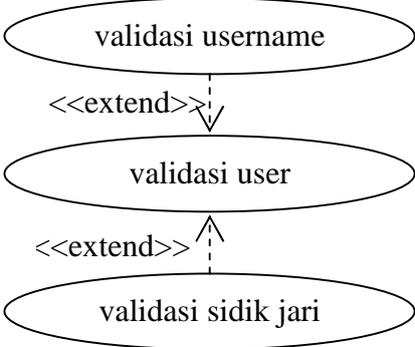
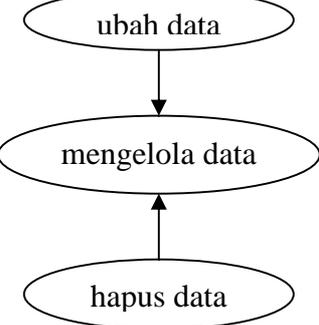
Rosa (2018: 156—158) menyatakan bahwa berikut adalah simbol-simbol yang sering digunakan dalam *use case diagram*, yaitu sebagai berikut:

Tabel 3.1 Simbol-simbol pada *Use Case Diagram*

No.	Simbol	Deskripsi
1.		<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di 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 aktor</p>

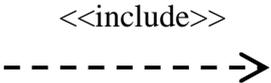
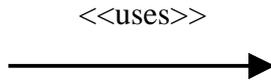
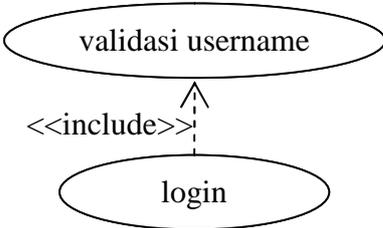
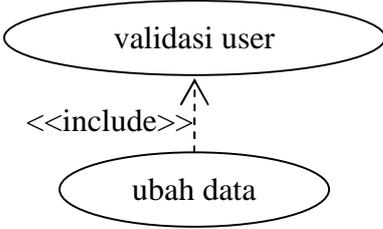


Lanjutan Tabel 3.1 Simbol-simbol pada *Use Case Diagram*

3.	Asosiasi/ <i>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
4.	Ekstensi/ <i>extend</i> <<extend>> 	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 case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal 
5.	Generalisasi/ <i>generalization</i> 	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:  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)



Lanjutan Tabel 3.1 Simbol-simbol pada *Use Case Diagram*

<p>6. Menggunakan/ <i>include/ uses</i></p>	<p style="text-align: center;">   </p>	<p>relasi use case tambahan ke sebuah use case dimana use case ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di use case:</p> <ul style="list-style-type: none"> • Include berarti use case yang ditambahkan akan selalu di panggil saat use case tambahan dijalankan, misalnya pada kasus berikut: <p style="text-align: center;">  </p> <ul style="list-style-type: none"> • Include berarti berarti use case yang ditambahkan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah dijalankan sebelum use case tambahan di jalankan, misal pada kasus berikut: <p style="text-align: center;">  </p> <p>Kedua intrepretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
---	---	--

(Sumber: Rosa 2018: 156—158)



2.2.4. Activity Diagram

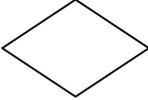
2.2.4.1. Pengertian Activity Diagram

Rosa (2018: 161) menyatakan “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.”

2.2.4.2. Simbol-simbol pada Activity Diagram

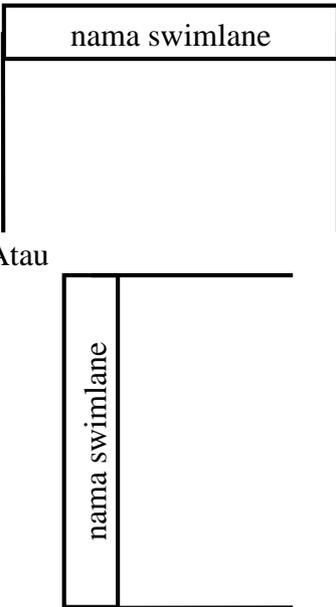
Rosa (2018: 162—163) menyatakan bahwa berikut adalah simbol-simbol yang ada pada *activity diagram*, yaitu sebagai berikut:

Tabel 3.2 Simbol-simbol pada *Use Case Diagram*

No.	Simbol	Deksripsi
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 sistem, sebuah diagram aktivitas memiliki sebuah status akhir



Lanjutan Tabel 3.2 Simbol-simbol pada *Use Case Diagram*

6.	<p>Swimlane</p> 	<p>memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
----	--	--

(Sumber: Rosa 2018: 162—163)

2.2.5. Class Diagram

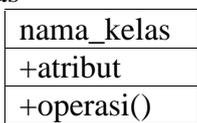
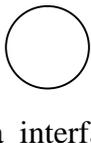
2.2.5.1. Pengertian Class Diagram

Rosa (2018: 141) menyatakan “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.”

2.2.5.2. Simbol-simbol pada Class Diagram

Rosa (2018: 146—147) menyatakan bahwa berikut adalah simbol-simbol yang ada pada *class diagram*, yaitu sebagai berikut:

Tabel 3.3 Simbol-simbol pada *Class Diagram*

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	<p>kelas pada struktur sistem</p>
2.	<p>antarmuka/ <i>interface</i></p> 	<p>sama dengan konsep interface dalam pemrograman berorientasi objek</p>



Lanjutan Tabel 3.3 Simbol-simbol pada *Class Diagram*

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 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 antarkelas
7.	agregasi/ <i>aggregation</i> 	relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

(Sumber: Rosa 2018: 146—147)

2.2.6. Sequence Diagram

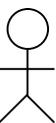
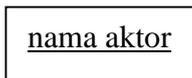
2.2.6.1. Pengertian Sequence Diagram

Rosa (2018: 165) menyatakan “diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.”

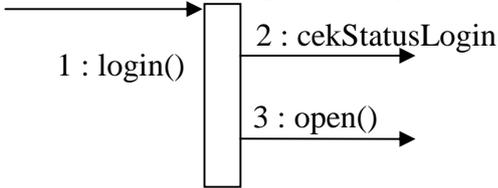
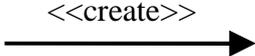
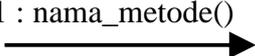
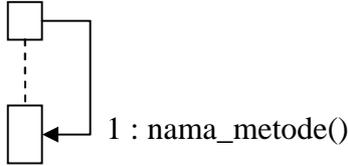
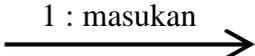
2.2.6.2. Simbol-simbol pada Sequence Diagram

Rosa (2018: 165—167) menyatakan bahwa berikut adalah simbol-simbol yang ada pada *class diagram*, yaitu sebagai berikut:

Tabel 3.4 Simbol-simbol pada *Sequence Diagram*

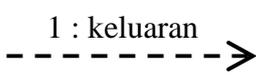
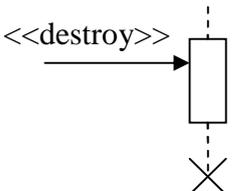
No.	Simbol	Deskripsi
1.	aktor  nama aktor atau  tanpa waktu aktif	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor

Lanjutan Tabel 3.4 Simbol-simbol pada *Sequence Diagram*

2.	garis hidup/ <i>lifeline</i> 	menyatakan kehidupan suatu objek
3.	Objek <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">nama objek: nama kelas</div>	menyatakan objek yang berinteraksi pesan
4.	waktu aktif 	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 create 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	pesan tipe call 	menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, 
7.	pesan tipe send 	menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim



Lanjutan Tabel 3.4 Simbol-simbol pada *Sequence Diagram*

8.	pesan tipe return 	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 destroy 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

(Sumber: Rosa 2018: 165—167)

2.2. Pengertian Judul

2.3.1. Pengertian Aplikasi

Indrajani (2018: 3) menyatakan “aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.”

Paramytha (2016: 56) menyatakan bahwa “aplikasi (*application software*) merupakan perangkat lunak yang dikembangkan untuk menyelesaikan suatu aplikasi tertentu.”

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa aplikasi adalah program yang dikembangkan dalam menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.

2.3.2. Pengertian Baca

Menurut Kamus Bahasa Indonesia (2015: 44) menyatakan baca adalah melihat dan memahami isi dari apa yang tertulis (dengan melisankan atau hanya dalam hati) atau mengeja apa yang tertulis.

Haryadi dikutip Widiyanto (2015: 2) menyatakan bahwa membaca merupakan interaksi antara pembaca dan penulis. Interaksi tersebut tidak langsung, namun bersifat komunikatif.



Jadi dapat berdasarkan pengertian diatas dapat disimpulkan bahwa baca atau membaca adalah interaksi antara pembaca dan penulis dimana kegiatan melihat serta memahami isi dari apa yang tertulis baik dengan melisankan atau hanya dalam hati.

2.3.3. Pengertian Koran

Suharyanto (2016: 126) menyatakan bahwa surat kabar adalah sinonim dari koran, dalam Kamus Besar Bahasa Indonesia dinyatakan bahwa surat kabar merupakan lembaran-lembaran kertas yang bertuliskan kabar atau berita yang terbagi-bagi atas kolom-kolom, terbit setiap hari atau secara periodik.

2.3.4. Pengertian Online

Romli (2012) menyatakan bahwa media *online* adalah media massa yang tersaji secara *online* di situs *web (website)* internet. Media online adalah media massa “generasi ketiga” setelah media cetak (*printed media*) dan media elektronik (*electronic media*).

2.3.5. Pengertian Palembang Ekspres (Palpres)

Palembang Ekspres atau disingkat Palpres adalah sebuah surat kabar harian yang terbit di Palembang, Indonesia. Saat ini Palpres sudah berkembang menjadi koran Provinsi, dengan cakupan seluruh kabupaten dan kota di Palembang, sebaran Palpres juga mencakup wilayah Sumatera Bagian Selatan (Sumbagsel). Pada umumnya Palpres memberitakan tentang hiburan, nasional, olahraga dan masih banyak lagi.

2.3.6. Pengertian Android

Sifauttori (2017: 309) menyatakan “*android* merupakan sebuah hasil sistem operasi modifikasi dari linux, yang digunakan untuk ponsel (*smartphone*)/ tablet hingga perangkat jam tangan sampai televisi pintar, dalam perkembangannya dalam bidang bidang handphone (*smartphone*).”

Surahman (2017: 36) menyatakan “*android* adalah sistem operasi yang bersifat open source berbasis Linux yang dirancang untuk perangkat seluler layar.”



Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian *android* adalah sebuah hasil sistem operasi modifikasi dari linux yang digunakan untuk ponsel/ tablet hingga perangkat jam sampai televisi pintar yang sangat berguna dalam berkomunikasi dan mencari informasi.

2.3.7. Pengertian Aplikasi Baca Koran *Online* pada Palembang Ekspres

Pengertian Aplikasi Baca Koran *Online* pada Palembang Ekspres adalah aplikasi yang dibuat untuk membantu para pembaca Palembang Ekspres dalam membaca koran online dengan menggunakan *smartphone* berbasis *android*.

2.3. Teori Program

2.4.1. Pengertian *XAMPP*

Riyanto (2015: 1) menyatakan “*XAMPP* merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*. *XAMPP* mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket.”

Palit (2015: 2) mengatakan bahwa *XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL Database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. *XAMPP* singkatan dari *X* (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*.

Jadi dapat disimpulkan bahwa *XAMPP* adalah perangkat lunak yang digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL Database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.

2.4.2. Pengertian *MySQL*

Kustiyahningsih dikutip Palit (2015: 3) menyatakan bahwa “*MySQL* adalah sebuah basis data yang mengandung satu atau jumlah table. Table terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah table.”



Wahana Komputer dikutip Palit (2015: 3) menyatakan bahwa “*MySQL* adalah *database server open source* yang cukup populer keberadaannya.”

Berdasarkan kedua pengertian diatas bahwa *MySQL* adalah *database server open source* yang mengandung satu atau jumlah table. Table terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah table.

2.4.3. Pengertian PHP

Menurut Madcoms (2016: 2) “*PHP* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam *HTML*.”

Raharjo (2016: 38) menyatakan “*PHP* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*”

Berdasarkan pengertian tersebut bahwa *PHP* adalah bahasa pemrograman yang disisipkan ke dalam *HTML* untuk membuat aplikasi *web*.

2.4.4. Pengertian phpMyAdmin

Madcoms (2016: 12) menyatakan bahwa *phpMyAdmin* adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*. *phpMyAdmin* dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan meng-*update* data dengan *GUI* dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara manual.

Hidayatullah (2017: 180) menyatakan “*phpMyAdmin* adalah *tool open source* yang ditulis dalam bahasa *PHP* untuk menangani administrasi *MySQL* berbasis *World Wide Web*.”

Jadi berdasarkan pengertian tersebut dapat disimpulkan bahwa *phpMyAdmin* adalah sebuah aplikasi *database server MySQL* berbasis *web* yang berfungsi untuk membuat *database*, membuat tabel, meng-*insert*, menghapus dan meng-*update* data.

2.4.5. Pengertian HTML (Hyper Text Markup Language)

Raharjo (2016: 449) menyatakan “*HTML* adalah singkatan dari *Hyper Text Markup Language*. *HTML* merupakan *file* teks yang tulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke user melalui suatu aplikasi *web browser*.”



Achmad Solichin dikutip Novianty (2017: 42) menyatakan bahwa *HTML* merupakan bahasa pemrograman *web* yang memberitahukan peramban *web* (*web browser*) bagaimana menyusun dan menyajikan konten di halaman *web*.

Berdasarkan pengertian diatas dapat disimpulkan bahwa *HTML* (*Hyper text Markup Language*) adalah bahasa pemrograman atau *file* teks yang ditulis menggunakan aturan-aturan kode untuk menyusun dan menyajikan konten di halaman *web*.

2.4.6. Pengertian CSS

Abdulloh (2016: 2) menyatakan bahwa *CSS* singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Fungsi *CSS* adalah memberikan pengaturan yang lebih lengkap agar stuktur *website* yang dibuat dengan *HTML* terlihat lebih rapi dan elegan.

Achmad Solichin dikutip Novianty (2017: 42) menyatakan bahwa *CSS* memberikan cara yang mudah dan efisien bagi pemrogram untuk menentukan tata letak halaman web dan mempercantik halaman dengan elemen desain seperti warna, sudut bulat, gradien dan animasi.

Jadi dapat disimpulkan bahwa pengertian *CSS* (*Cascading Style Sheets*) adalah skrip yang digunakan untuk mengatur desain *website* dengan cara yang mudah dan efisien agar terlihat lebih rapi dan elegan.

2.4.7. Pengertian JavaScript

Winarno (2014: 129) mengatakan bahwa *JavaScript* adalah bahasa *scripting client site* yang sangat populer. *JavaScript* digunakan untuk memberikan efek pemrograman di halaman. *JavaScript* juga menjadi dasar yang bisa digunakan teknologi lainnya, seperti *Ajax*, *jQuery* dan *jQuery Mobile*.

Achamad Solichin dikutip Novianty (2017: 42) meyakini bahwa fungsi utama dari *javascript* adalah untuk menambah fungsionalitas dan kenyamanan halaman *web*. *Javascript* lebih fokus pada proses pengolahan data di sisi *client* serta menyajikan komponen *web* yang lebih interaktif.



Berdasarkan pengertian diatas dapat disimpulkan bahwa *JavaScript* adalah bahasa *scripting client site* yang digunakan untuk membuat program yang digunakan agar dokumen *HTML* yang ditampilkan dalam *browser* menjadi lebih interaktif.

2.4.8. Pengertian *jQuery*

Rohingun (2015: 1) menyatakan bahwa *jQuery* adalah salah satu dari sekian banyak *JavaScript library* yaitu kumpulan fungsi *JavaScript* yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode *JavaScript*. Dengan menggunakan *jQuery*, skrip *JavaScript* yang panjang dapat disingkat menjadi beberapa baris kode.

Aloysius Sigit dikutip Novianty (2017) menyatakan bahwa *jQuery* adalah *library* atau kumpulan kode *JavaScript* siap pakai. Keunggulan menggunakan *jQuery* dibandingkan *JavaScript* standar, yaitu menyederhanakan kode *JavaScript* dengan cara memanggil fungsi-fungsi yang disediakan *jQuery*.

Jadi dapat disimpulkan bahwa *jQuery* adalah kumpulan fungsi *JavaScript* yang siap pakai. Keunggulan menggunakan *jQuery* dibandingkan *JavaScript* standar, yaitu menyederhanakan kode *JavaScript* dengan cara memanggil fungsi-fungsi yang disediakan *jQuery*.