



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Perangkat Lunak

Menurut Sukanto dan Shalahuddin (2016:2), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Sedangkan Swara dan Febriadi (2016:8), “Perangkat lunak merupakan seluruh perintah yang digunakan untuk memproses informasi”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa perangkat lunak merupakan cara menggunakan yang ditujukan kepada komputer sesuai kebutuhan pemakai yang digunakan untuk memproses informasi.

2.1.2 Pengertian Komputer

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

Hal senada juga dikemukakan oleh Ikhsan dan Kurniawan (2015:13), “Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah.”

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



2.1.3 Pengertian Data

Menurut Kristanto (2018:8) menyatakan bahwa, “Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi”.

Selain itu menurut Rusdiana dan Irfan (2016:68), “Data adalah fakta-fakta mentah yang harus dikelola untuk menghasilkan informasi yang memiliki arti bagi suatu organisasi atau perusahaan.

Berdasarkan pendapat di atas, dapat disimpulkan bahwa definisi dari data adalah fakta yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model untuk menghasilkan informasi yang memiliki arti bagi suatu organisasi atau perusahaan.

2.1.4. Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2016:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2016:128-131) adalah sebagai berikut:

1. *Inception* (permulaan)

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

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 dan 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 menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

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 Teori Khusus

2.2.1 Pengertian Kamus Data

Sukanto dan Shalahuddin (2016:73) menjelaskan, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (ouput) dapat dipahami secara umum (memiliki standar cara penulisan).” Kamus data memiliki beberapa simbol sebagai berikut :

Tabel 2.1 Simbol-simbol Kamus Data

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



6.	*..*	Batas komentar
----	------	----------------

Sumber: Sukamto dan Shalahuddin (2016:74)

2.2.2 Pengertian *Unified Modeling Language* (UML)

Sukamto dan Shalahuddin (2016:133), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

“*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.”

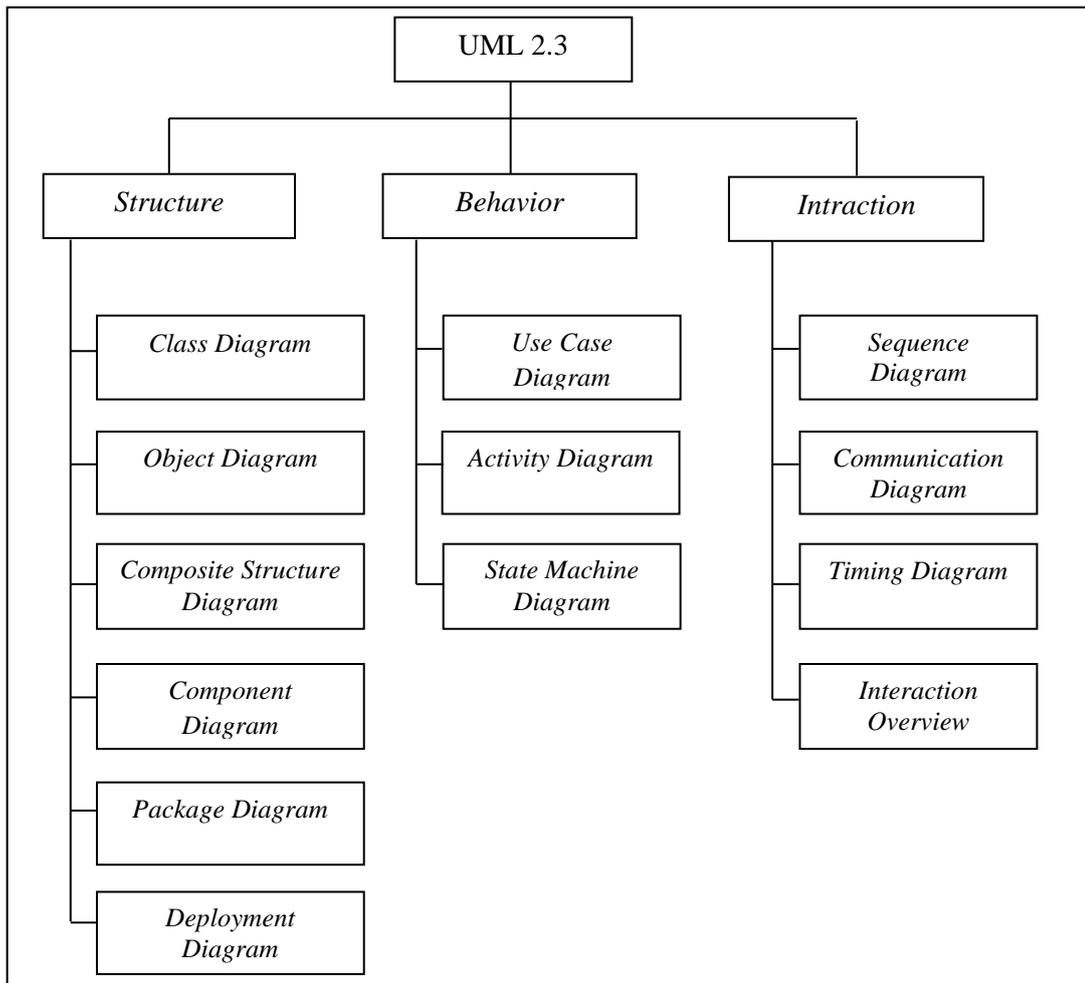
UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.



Gambar 3.1. Tampilan Logo UML

2.2.3 Kategori *Unified Modeling Language* (UML)

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



Gambar 2.2 Kategori dan Macam-macam Diagram UML

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

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk 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.



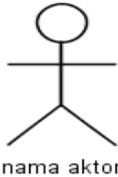
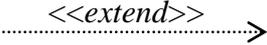
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.4 Jenis-Jenis Diagram UML

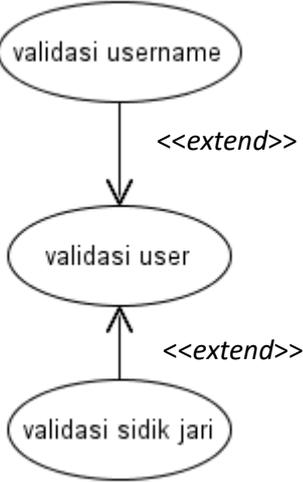
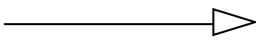
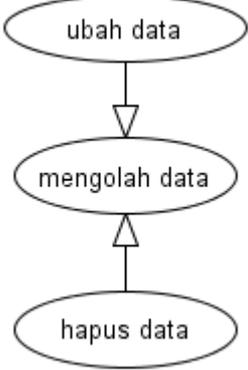
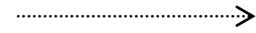
2.2.4.1 Pengertian *Use case Diagram*

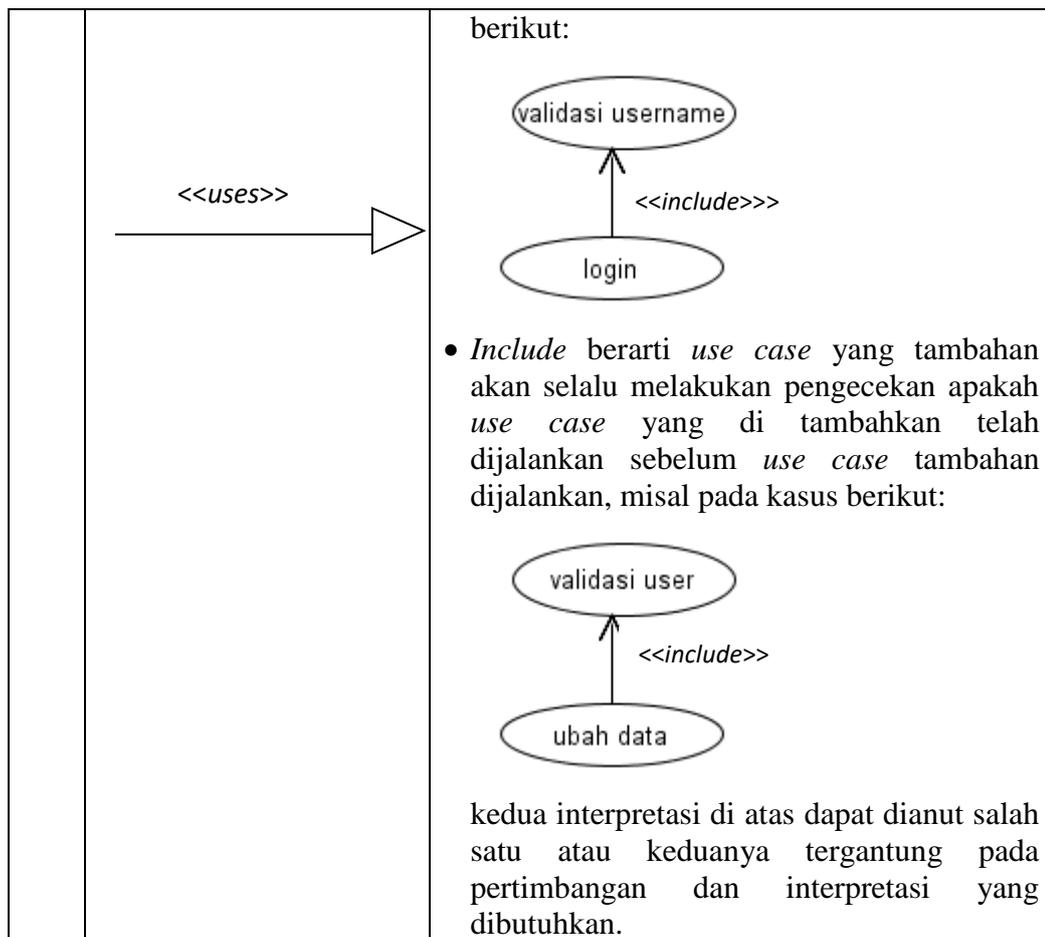
Menurut Sukamto dan Shalahuddin (2016:155), “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.2 Simbol-simbol *Use case Diagram*

No	Simbol	Deskripsi
1.		fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i> .
2.		orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.		komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan 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, misalnya



		 <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>
<p>5.</p>	<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,</p>  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p>6.</p>	<p>menggunakan / include / uses</p>  <p><<include>></p>	<p>relasi tambahan ke sebuah <i>use case</i> dimana <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 <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus



Sumber: Sukamto dan Shalahuddin (2016:156)

Ada dua hal utama pada *use case* yaitu:

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.2.4.2 Pengertian *Class Diagram*

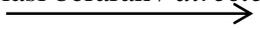
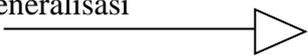
Menurut Sukamto dan Shalahuddin (2016:141), "*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program



atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.”

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

Tabel 2.3 Simbol-simbol *Class* Diagram

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

Sumber: Sukamto dan Shalahuddin

2.2.4.3 Pengertian *Activity Diagram*

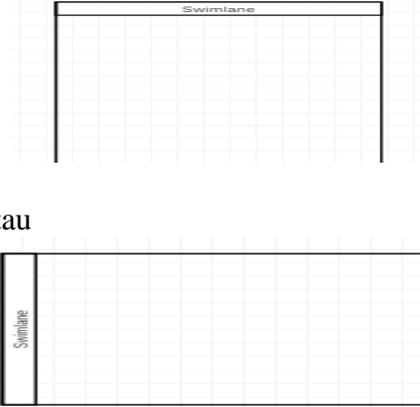
Menurut Sukamto dan Shalahuddin (2016:161), “*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa



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

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

Tabel 2.4 Simbol-simbol *Activity Diagram*

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

Sumber: Rosa dan Shalahuddin (2016:162)



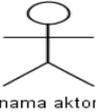
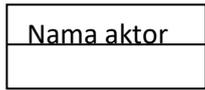
2.2.4.4 Pengertian *Sequence Diagram*

“Diagram sekuen 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 beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case” (Sukamto dan Shalahuddin, 2016:165). “Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu” (Safaat, 2015:33-34).

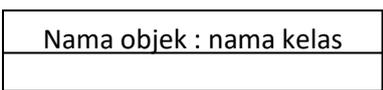
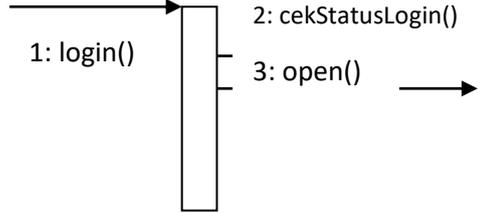
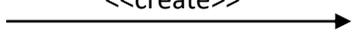
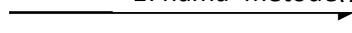
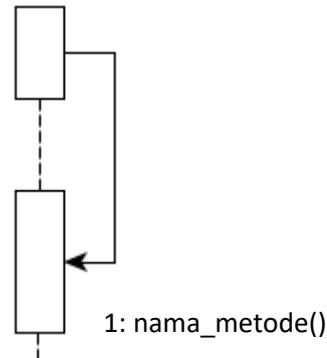
Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada Sequence Diagram :

Tabel 2.5 Simbol-simbol pada Sequence Diagram

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



2.	Garis hidup / <i>lifeline</i> 	menyatakan kehidupan suatu objek
3.	Objek 	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  maka cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif
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,  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 send 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe return 1: keluaran 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe destroy <<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 dan Shalahuddin (2013:165-167)6

2.3 Teori Judul

2.3.1 Pengertian Sistem Informasi

Menurut Pratama (2014: 7), “Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur dan sumber daya manusia (SDM) yang terlatih”.

Indrajani (2014:3), “Sistem Informasi merupakan kombinasi teratur apa pun dari orang-orang, hardware, software, jaringan komunikasi, dan sumber daya data, yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.”

Jadi dapat disimpulkan bahwa sistem informasi adalah suatu sistem gabungan dari empat bagian yaitu orang-orang, hardware, software, jaringan komunikasi dan sumber daya data yang menyediakan laporan dengan menyebarkan informasi dalam sebuah organisasi.



2.3.2 Pengertian Kode QR

Menurut Muharom (2016:114), “Quick Response Code sering di sebut QR Code atau kode QR adalah semacam simbol dua dimensi yang dikembangkan oleh Denso Wave yang merupakan anak perusahaan dari Toyota sebuah perusahaan Jepang pada tahun 1994. Tujuan dari QR Code ini adalah untuk menyampaikan informasi secara cepat dan juga mendapat tanggapan atau respon secara cepat. Pada awalnya QR Code digunakan untuk pelacakan bagian kendaraan untuk perusahaan manufacturing”.

2.3.3 Pengertian Presensi

Menurut Kamus Besar Bahasa Indonesia Presensi memiliki arti kehadiran.

2.3.4 Pengertian Sistem Informasi Pendaftaran Mahasiswa Baru di Politeknik Negeri Sriwijaya Studi Kasus Scan Kode QR Lokasi Ujian dan Presensi

Sistem Informasi Pendaftaran Mahasiswa Baru di Politeknik Negeri Sriwijaya Studi Kasus Scan Kode QR digunakan untuk pendaftaran mahasiswa baru untuk mengecek lokasi pelaksanaan ujian dan pengisian daftar hadir oleh pengawas ujian.

2.4 Teori Program

2.4.1 Pengertian Basis Data

Sukamto dan Shalahuddin (2013:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

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

Kesimpulannya, Basis data adalah sistem komputerisasi sebagai aktifitas dari satu atau lebih organisasi.



2.4.2 Pengertian MySQL

Menurut Prasetyo, Pattiasina dan Soetarmono (2015:13) menyatakan bahwa, "Mysql adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL bersifat free dengan lisensi GNU General Public License (GPL). Dengan adanya keadaan ini maka anda dapat menggunakan software ini dengan bebas tanpa perlu harus takut dengan lisensi yang ada.

Sedangkan menurut Hendry (2015:7), "MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional".

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

2.4.3 Pengertian Android

Menurut Safaat (2015:1), "Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi".

Menurut Firly (2018:2), "Android yaitu dalam Bahasa Inggris istilah Android berarti "Robot yang menyerupai manusia", hal tersebut dapat terlihat jelas pada icon android yang menggambarkan sebuah robot berwarna hijau yang memiliki sepasang tangan dan kaki".

Jadi kesimpulannya yaitu Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi yang menggambarkan sebuah robot menyerupai manusia.



3.4.4 Pengertian Xampp

Pratama (2014:440), “XAMPP adalah aplikasi web server bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi linux maupun sistem operasi windows”.

Santosa & Radna Nurmalina (2016:86-87) menyatakan bahwa, “Dengan menggunakan XAMPP tidak dibingungkan dengan penginstalan program-program lain, karena semua kebutuhan telah tersedia oleh XAMPP. Yang terdapat pada XAMPP diantaranya: Apache, MySQL, PHP, FilZilla FTP Server, PhpmyAdmin dll”.

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa XAMPP yaitu aplikasi web server dengan kebutuhan akan program-program telat terinstal.

3.4.5 Pengertian Java

Enterprise (2016:1), “Java merupakan Bahasa pemrograman yang sangat populer karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangatlah luas, mulai dari komputer hingga smartphone”.

Sedangkan menurut Hariyanto (2014:3), “ Java merupakan bahasa orientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat internet/jaringan komunikasi”.

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