



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Murhada dan Giap (2011:21), “Komputer adalah alat elektronik yang dapat menerima *input* data dan mengolahnya menjadi suatu informasi, dengan menggunakan suatu program yang tersimpan di memorinya, serta dapat menyimpan program dan hasil pengolahannya, dimana bekerja secara otomatis.”

Selain pengertian diatas, adapun pengertian komputer menurut Ariyus dan Pangera (2010:45) yaitu suatu perangkat yang bisa mengerjakan *input*, proses, *output* serta aktivitas penyimpanan oleh dirinya sendiri. Terdapat beberapa kategori komputer yang meliputi komputer pribadi (PC), komputer *mobile*, konsol *game*, *server*, *mainframe*, *super* komputer dan komputer tertanam.

2.1.2. Pengertian Perangkat Lunak

Pressman (2012:5), “Perangkat lunak adalah: (1) intruksi-intruksi (program komputer) yang ketika dijalankan menyediakan fitur-fitur, fungsi-fungsi, dan kinerja-kinerja yang dikehendaki; (2) struktur data yang memungkinkan program-program memanipulasi informasi, dan (3) informasi deskriptif pada salinan tercetak dan bentuk-bentuk maya yang menggambarkan pengoperasian dan penggunaan program-program”.

Sukamto dan Shalahuddin (2018:2), “Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (*software*). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.”

2.1.3. Pengertian Data

Murhada dan Giap (2011:160), “Data adalah representasi dari fakta dunia nyata yang mewakili suatu objek yang sedang ditinjau (manusia, barang,



peristiwa, hewan, konsep, keadaan dan sebagainya), dan direkam dalam bentuk huruf, kata, angka, simbol, gambar, bunyi, atau kombinasinya.”

Al-Fatta dalam Ghazali (2016:7), “Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun”.

2.1.4. Pengertian Sistem

Fathansyah dalam Ghazali (2016:8), “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu”.

Al Fatta dalam Ghazali (2016:8), “Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain”.

2.1.5. Karakteristik Sistem

Al Fatta dalam Ghazali (2016:8), karakteristik sistem terdiri dari :

a. Batasan (*boundary*)

Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.

b. Lingkungan (*environment*)

Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.

c. Masukan (*input*)

Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang di konsumsi dan di manipulasi oleh suatu sistem.

d. Keluaran (*output*)

Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.



e. Komponen (*output*)

Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan *input* menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

f. Penghubung (*interface*)

Tempat dimana komponen atau sistem dan lingkungannya bertemu dan berinteraksi.

g. Penyimpanan (*storage*)

Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut yang bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

2.1.6. Identifikasi Masalah

Menurut Al Fatta dalam Nilawati dan Karina (2016:30), “Pada tahapan analisis sistem, analisis mempunyai tugas mendefinisikan masalah sistem, melakukan studi kelayakan, dan menganalisis kebutuhan sistem yang akan dikembangkan. Masalah yang dipelajari analisis sistem adalah masalah yang dihadapi pengguna”.

2.1.7. Tahapan Analisis Sistem

Al Fatta dalam Nilawati dan Karina (2016:30), “Analisis sistem sebuah sistem istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecah masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka”.



2.1.8. Metodologi Pengembangan Sistem

Al Fatta dalam Ghazali (2016:9), siklus pengembangan sistem yaitu:

a. Analisis

Bertujuan untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru.

b. Desain

Bertujuan untuk mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil.

c. Implementasi

Bertujuan untuk mengimplementasikan sistem yang baru.

d. Pemeliharaan

Bertujuan agar sistem dapat berjalan secara optimal.

2.1.9. Studi Kelayakan

Al Fatta dalam Nilawati dan Karina (2016:31), “Dokumen yang dihasilkan dari tahapan-tahapan sebelumnya dikumpulkan menjadi suatu proposal pendahuluan proyek”. Untuk memastikan usulan tersebut bisa diteruskan menjadi proyek dan menguntungkan maka proyek harus dievaluasi kelayakannya dari berbagai segi kelayakan, diantaranya :

1. Kelayakan Teknis

Kelayakan teknis menyoroti kebutuhan sistem yang telah disusun dari aspek teknologi yang akan digunakan. Jika teknologi yang dikehendaki untuk pengembangan sistem merupakan teknologi yang mudah di dapat, murah, dan tingkat pemakaiannya mudah, maka secara teknis usulan kebutuhan sistem bisa dinyatakan layak.

2. Kelayakan Operasional

Kelayakan operasional menyangkut beberapa aspek. Untuk disebut layak secara operasional, usulan kebutuhan sistem harus benar-benar bisa menyelesaikan masalah yang ada di sisi pemesan sistem informasi. Di samping itu, informasi yang dihasilkan oleh sistem harus merupakan informasi yang benar-benar dibutuhkan oleh pengguna tepat pada saat pengguna menginginkannya.



3. Kelayakan Ekonomi

Kelayakan ekonomi berhubungan dengan *return on investment* atau berapa lama biaya investasi dapat kembali. Analisis kelayakan ekonomi juga mempertimbangkan apakah bermanfaat melakukan investasi ke proyek ini atau kita harus melakukan sesuatu yang lain. Suatu proyek yang besar biasanya lebih menekankan kelayakan ekonomi karena umumnya berhubungan dengan biaya yang terbilang besar.

2.1.10. Tahapan Desain

Al Fatta (2007) dalam Nilawati dan Karina (2016:33), “Tahapan desain adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil. Tahapan desain sistem dapat dibagi menjadi 2 tahap, yaitu desain logis (*logical design*) dan tahapan desain fisik (*physical design*)”.

1. Desain Logis

Desain logis adalah bagian dari fase desain dalam SDLC (*System Development Life Cycle*) dimana semua fitur-fitur fungsional dari sistem dipilih dari tahapan analisis yang dideskripsikan terpisah dari platform komputer yang nanti digunakan. Tahapan desain logis biasanya menghasilkan beberapa dokumen, diantaranya dokumen model data, dokumen model proses, rancangan tabel, hierarki antar modul, sampai desain antar muka dari sistem yang akan dibuat.

2. Desain Fisik

Pada tahapan desain fisik, spesifikasi logis diubah ke dalam detail teknologi di mana pemrograman dan pengembangan sistem bisa diselesaikan. Pada tahapan inilah aktifitas *coding* dilakukan.



2.1.11. Implementasi

Menurut Mujilan (2017:22),” Tahap ini merupakan tahap meletakkan sistem agar siap digunakan. Pada tahap ini dapat dilakukan: rancangan implementasi, memilih dan melatih personil, mempersiapkan tempat dan lokasi sistem, menguji sistem, serta melakukan konversi sistem”.

2.1.12. Operasi dan perawatan

Menurut Mujilan (2017:22),” Kegiatan perawatan perlu dilakukan antara lain karena: 1). Karena memperbaiki kesalahan. 2). adanya perubahan karena permintaan pengguna sistem, 3). adanya perubahan lingkungan luar, 3). keperluan peningkatan sistem”.

2.2. Teori Khusus

2.2.1. Pengertian Pemrograman Berorientasi Objek

Menurut Sukanto dan Shalahuddin (2016:99), “Pemrograman berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.”

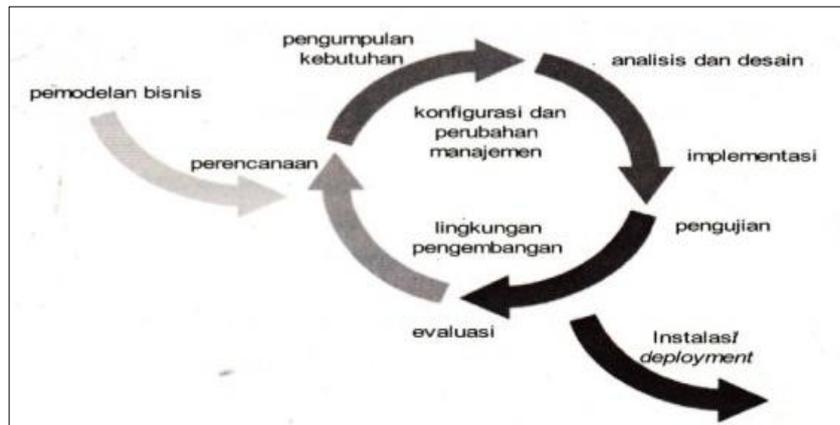
Dari definisi di atas dapat disimpulkan bahwa pemrograman berorientasi objek adalah program komputer dari berbagai objek yang melakukan suatu tindakan terhadap masing-masing objek.

2.2.2. Rational Unified Process (RUP)

Sukanto dan Shalahuddin (2016:125), RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use-case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak.



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



Sumber: Sukamto dan Shalahuddin (2016:125)

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

2.2.2.1. Fase *Rational Unified Process* (RUP)

Sukamto dan Salahuddin (2016:128-131), “RUP (*Rational Unified Process*) memiliki empat buah tahap atau fase yang dapat dilakukan secara iteratif.” Berikut ini penjelasan untuk setiap fase pada RUP (*Rational Unified Process*). :

- a. *Inception* (permulaan) Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).
- b. *Elaboration* (perluasan/perencanaan) Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah sistem yang diinginkan dapat dibuat atau tidak. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).
- c. *Construction* (konstruksi) Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih 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.

- d. *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. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada RUP (Rational Unified Process) dijalankan secara berurutan dan literatif dimana setiap iterasi dapat digunakan untuk memperbaiki literasi berikutnya.

2.2.3. Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin (2016:133), “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.”

Dari pengertian diatas dapat disimpulkan bahwa *UML (Unified Modelling Language)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasikan hingga mendokumentasikan sebuah sistem pengembangan perangkat lunak berorientasi objek.



2.2.3.1. Klasifikasi Diagram UML (*Unified Modelling Language*)

Sukanto dan Salahuddin (2016:140), “UML (*Unified Modelling Language*) terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. Structure Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. Behavior Diagram

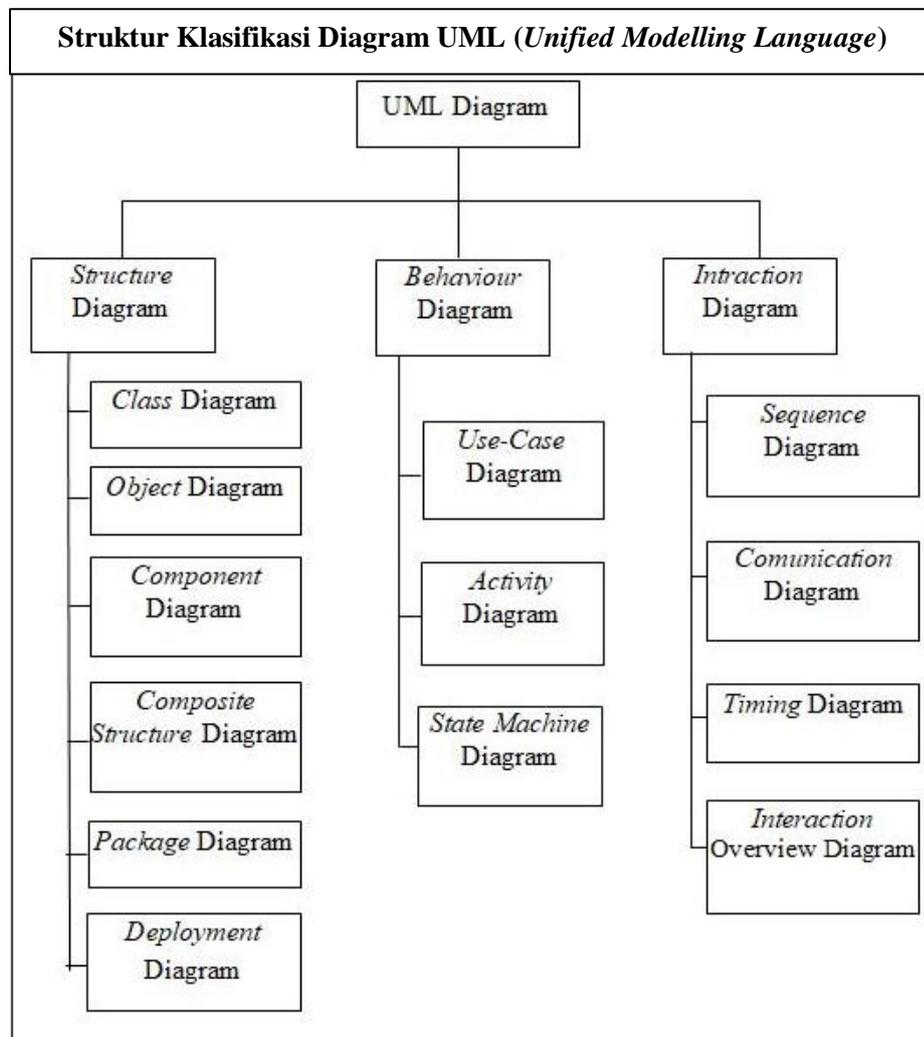
Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. Interaction Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar yang terklasifikasi seperti dibawah ini:



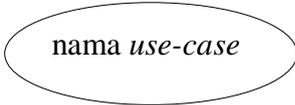
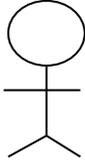
Sumber: Sukamto dan Shalahuddin (2016:140)

Gambar 2.2. Klasifikasi Diagram UML (*Unified Modelling Language*)

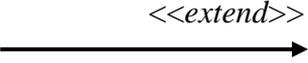
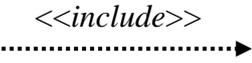
2.2.4. *Use-Case Diagram*

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 informasi yang akan dibuat”. Adapun simbol-simbol yang digunakan dalam *use-case* adalah sebagai berikut:

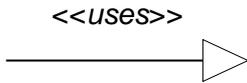
Tabel 2.1. Simbol-simbol *Use-Case Diagram*

No	Simbol	Deskripsi
1.	<p data-bbox="368 416 770 450"><i>Use-Case</i></p>  <p data-bbox="440 506 632 533">nama <i>use-case</i></p>	<p data-bbox="786 416 1367 667">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use-case</i>.</p>
2.	<p data-bbox="368 752 770 786">Aktor / <i>actor</i></p>  <p data-bbox="475 1003 679 1070">nama aktor nama_<i>interface</i></p>	<p data-bbox="786 752 1367 1160">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
3.	<p data-bbox="368 1245 770 1279">Asosiasi / <i>association</i></p> 	<p data-bbox="786 1245 1367 1384">Komunikasi antar aktor dan <i>use-case</i> yang berpartisipasi pada <i>use-case</i> atau <i>use-case</i> yang memiliki interaksi dengan actor.</p>

Lanjutan Tabel 2.1. Simbol-simbol *Use-Case Diagram*

No	Simbol	Deskripsi
4.	Ekstensi / <i>extend</i> 	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; ditambahkan, missal 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.
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.
6.	Menggunakan / <i>include</i> / <i>uses</i> 	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 ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :

Lanjutan Tabel 2.1. Simbol-simbol *Use-Case Diagram*

No	Simbol	Deskripsi
		<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 berikut: <p style="text-align: center;"><<include>></p> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut: <p style="text-align: center;"><<include>></p> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Sukamto dan Shalahuddin (2016:156-158)

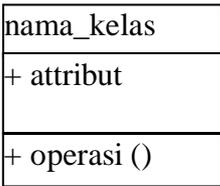


2.2.5. Class Diagram

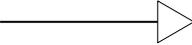
Sukanto dan Shalahuddin (2016:141), Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

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

Tabel 2.2. Simbol-simbol Class Diagram

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antar muka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman 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 kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

Lanjutan Tabel 2.2. Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
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: Sukamto dan Shalahuddin (2016:144-147)

2.2.6. Activity Diagram

Sukamto dan Shalahuddin (2016:161), *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.

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

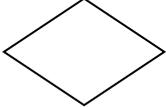
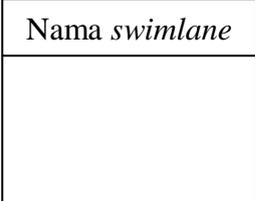
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface*



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 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 sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Sukamto dan Shalahuddin (2016:162-163)

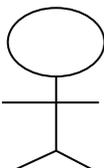


2.2.7. Sequence Diagram

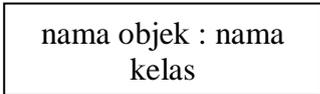
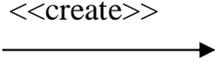
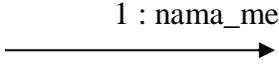
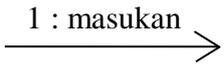
Menurut Sukamto dan Shalahuddin (2016:165), “Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi 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 pesesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.”

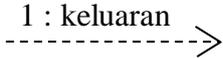
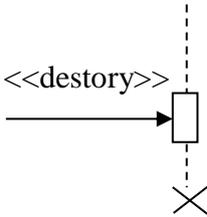
Tabel 2.4. Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Aktor Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama_aktor</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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>

Lanjutan Tabel 2.4. Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
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.
5.	pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

Lanjutan Tabel 2.4. Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
8.	Pesan tipe <i>return</i> 	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>destrory</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: Sukamto dan Shalahuddin (2016:165-167)

2.2.8. Metode Pengujian Perangkat Lunak

2.2.8.1. *Back-Box Testing* (Pengujian Kotak Hitam)

Sukamto dan Shalahuddin (2016:275), “*Black-box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.”

Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:

- Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.



b. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

2.3. Teori Judul

2.3.1. Pengertian Aplikasi

Budiharto dalam Ghazali (2016:18), “Aplikasi merupakan program yang dapat berjalan di komputer tersendiri (*stand alone computer*), dari mulai program yang *simple* sampai dengan program besar dan rumit”.

2.3.2. Pengertian Evaluasi

Hakam dalam Niantyasari (2018:3), “Evaluasi adalah tindakan mengamati dan melakukan investigasi efektifitas program, serta menilai kontribusi program terhadap perubahan dan menilai kebutuhan untuk perbaikan, kelanjutan, atau perluasan program”.

2.3.3. Pengertian Kinerja

Jelantik dalam Ghazali (2016:18), “Kinerja atau *performance* adalah usaha yang dilakukan dari hasil kerja yang dapat dicapai seseorang atau sekelompok orang dalam suatu organisasi sesuai dengan tugas dan tanggung jawab”.

2.3.4. Pengertian Pegawai

Widjaja dalam Alim (2013:38), “Pegawai adalah orang-orang yang dikerjakan dalam suatu badan tertentu, baik di lembaga-lembaga pemerintah maupun di badan-badan usaha”.

Dari definisi di atas dapat diketahui bahwa pegawai merupakan modal pokok dalam suatu organisasi. Baik itu organisasi pemerintah maupun organisasi swasta. Dikatakan bahwa pegawai merupakan modal pokok dalam suatu organisasi karena berhasil tidaknya suatu organisasi dalam mencapai tujuannya tergantung pada pegawai yang memimpin dan melaksanakan tugas-tugas yang ada dalam organisasi tersebut. Pegawai yang telah memberikan tenaga dan pikirannya dalam melaksanakan tugas ataupun pekerjaan, baik itu organisasi pemerintah



maupun organisasi swasta akan mendapat imbalan sebagai balas jasa atas pekerjaan yang telah dikerjakan.

2.3.5. Pengertian *Website*

Hariyanto dalam Destiningrum dan Adrian (2017:32), “*Web* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya. Baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*)”.

Berdasarkan definisi tersebut, penulis menyimpulkan bahwa web adalah sebuah software yang berfungsi untuk menampilkan dokumen-dokumen pada suatu web yang membuat pengguna dapat mengakses internet melalui *software* yang terkoneksi dengan internet.

2.3.6. Pengertian *CodeIgniter*

Sidik dalam Destiningrum dan Adrian (2017:32), “*CodeIgniter* adalah sebuah *framework php* yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuatnya dari awal”.

2.4. Teori Program

2.4.1. Sekilas Tentang *PHP*

2.4.1.1. Pengertian *PHP*

Wahidin (2010:54), “*PHP (PHP Hypertext Preprocessor)* adalah bahasa pemrograman berbasis *server side*. Artinya untuk dapat menjalankan *script PHP* anda membutuhkan *web server* yang dalam hal ini *Apache*. *PHP* berlisensi *free* atau gratis untuk digunakan.”

Sidik dalam Nilawati (2016:38), “*PHP* merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen *HTML* secara *on the fly*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan



dokumen *HTML* yang dibuat dengan menggunakan *editor teks* atau *editor HTML*’.

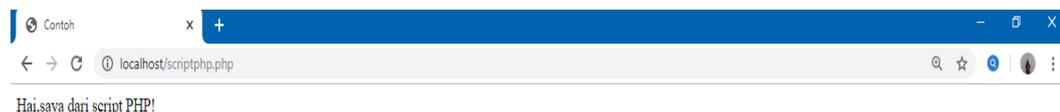
Dari definisi tersebut maka *PHP* dapat diartikan sebagai bahasa pemrograman *web* berbasis server side yang berupa *script-script* penerjemah menjadi bahasa yang dapat dimengerti oleh komputer.

2.4.1.2. Script *PHP*

Berikut contoh kode *PHP* yang berada dalam kode *HTML* (disimpan dengan ekstensi.php) :

```
<html>
<head>
<title>Contoh</title>
<body>
<?php echo "Hai,saya dari script PHP!" ; ?>
</body>
```

Bila dijalankan melalui *browser*, maka kode tersebut akan membentuk tampilan sebagai berikut :



Gambar 2.3 Contoh Script *PHP*

2.4.2. Sekilas Tentang Basis Data (*Database*)

2.4.2.1. Pengertian Basis Data

Fathansyah dalam Marhada dan Giap (2011:160), “Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

Kristanto dalam Marhada dan Giap (2011:160), “Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut ditunjukkan dengan kunci dari tiap file yang ada untuk digunakan dalam satu lingkup perusahaan, instansi”.



Berdasarkan definisi diatas, dapat disimpulkan bahwa basis data adalah kumpulan file data yang dipadu dan saling berkaitan untuk membentuk suatu fondasi pengelolaan informasi pada satu organisasi.

2.4.2.2. Operasi Dasar Basis Data

Fathansyah dalam Ghazali (2016:20), Operasi-operasi dasar basis data meliputi :

1. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru
2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus dengan isinya, jika ada)
3. Pembuatan tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada
4. Penghapusan tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip
5. Penambahan/pengisian data baru ke sebuah tabel di sebuah basis data (*insert*), yang identik dengan penambahan lembaran arsip ke sebuah map arsip
6. Pengambilan data dari sebuah tabel (*query*), yang identik dengan pencarian lembaran arsip dari sebuah map arsip
7. Pengubahan data dari sebuah tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip
8. Penghapusan data dari sebuah tabel (*delete*), yang identik dengan penghapusan sebuah lembaran yang ada di sebuah map arsip.

2.4.3. Pengertian HTML

Fauziah (2014:2), “HTML merupakan jenis bahasa yang digunakan untuk membuat halaman *website* yaitu dengan menggunakan *tag-tag* yang telah dideklarasikan pada halaman *notepad* yang dapat saling berhubungan dengan dokumen HTML yang lainnya yang sering kita kenal dengan istilah *link*”.

Menurut Winarno dan Utomo dalam Prayitno dan Safitri (2015:2) “HTML singkatan dari Hypertext Markup Language dan berguna untuk menampilkan halaman web”.



2.4.4. Pengertian Cascading Style Sheet (CSS)

Suryana dan Koesheryatin dalam Ghazali (2016:21), “CSS (*Cascading Style Sheet*) adalah suatu bahasa *style sheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna dan semua yang berhubungan dengan tampilan”.

2.4.5. Pengertian MySQL

Sukamto dan Shalahuddin (2018:46), “SQL (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus”.

Raharjo dalam Ghazali (2016:21), “*MySQL* merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*)”.

2.4.6. Pengertian JavaScript

Suryana dan Koesheryatin dalam Ghazali (2016:21), “*JavaScript* adalah bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML, di mana objek tersebut dapat berupa suatu *window*, *frame*, URL, dokumen, *form*, *button*, atau item yang lain”.

2.4.7. Pengertian XAMPP

Wahana Komputer dalam Ghazali (2016:22), “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP*, dan *Perl*. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket”.



Menurut Binarso dkk (2012:76), “XAMPP adalah sebuah software web server apache yang didalamnya sudah tersedia database server MYSQL dan mendukung PHP programming”.