

## BAB II LANDASAN TEORI

### 1.1 Tinjauan Pustaka

Penelitian tentang Perancangan Sistem Informasi Inventaris Dan Peminjaman Barang Berbasis Web (Studi Kasus Di Universitas Teknokrat Indonesia), menurut penulis belum pernah dilakukan. Sebagai pendukung pernyataan maka penulis menguraikan hasil dari berbagai penelitian terkait sebagai berikut:

**Tabel 2.1 Tinjauan Pustaka**

| No | Nama Peneliti                 | Judul  | Permasalahan   | Metode                                      | Hasil   |
|----|-------------------------------|--|--|---|---|
| 1  | Rochmawati dan Saputra (2019) | Perancangan Sistem Informasi Inventaris Untuk Peminjaman Dan Pengembalian Barang Di Laboratorium Jurusan Teknik Informatika FT Unesa | Selama ini pencatatan yang dilakukan di laboratorium Jurusan Teknik Informatika dilakukan secara manual. | Metode Pengembangan Sistem <i>Prototype</i> | Aplikasi sistem informasi peminjaman alat laboratorium ini akan sangat bermanfaat dalam membantu hal-hal yang berhubungan dengan administrasi yang terkait dengan peminjaman alat laboratorium. |

**Tabael 2.1 Tinjauan Pustaka (Lanjutan)**

| No | Nama Peneliti      | Judul   | Permasalahan  | Metode                          | Hasil  |
|----|--------------------|---|---|---------------------------------|--|
| 2  | Samudra dkk (2021) | Perancangan Sistem Informasi Peminjaman dan Pengembalian Barang Laboratorium di SMK N 1 Solok Selatan | Labor SMK N 1 Solok Selatan proses transaksi peminjaman dan pengembalian barang masih melakukan pencatatan dengan kertas atau buku folio. | Metode Pengembangan Sistem SDLC | Dengan adanya sistem informasi peminjaman dan pengembalian barang laboratorium ini, transaksi peminjaman dan pengembalian barang labor dan pembuatan laporan tidak menggunakan sistem yang manual lagi, sehingga lebih menghemat waktu dalam melakukan transaksi |

**Tabael 2.1 Tinjauan Pustaka (Lanjutan)**

| No | Nama Peneliti              | Judul   | Permasalahan   | Metode                                      | Hasil   |
|----|----------------------------|---|--|---|---|
|    |                            |   |  |   | peminjaman dan pengembalian barang dan lebih efisien dalam pengolaan laporan.   |
| 3  | Sarwono dan Widjaja (2018) | Rancangan Sistem Informasi Pengadaan Dan Peminjaman Barang Berbasis Web Pada PT Jakarta <i>International Higher Education College (JIHEC)</i> | Pada PT. <i>JIHEC</i> proses penyampaian informasi yang berhubungan dengan pengadaan, peminjaman dan pengembalian barang masih menggunakan pencatatan biasa tanpa mesin. | Metode Pengembangan Sistem <i>Prototype</i> | Dengan adanya sistem informasi ini, tanda bukti pengembalian barang dapat dicetak oleh sistem sesuai dengan data yang dipinjam, sehingga pegawai memiliki tanda bukti bahwa barang sudah dikembalikan |

**Tabael 2.1 Tinjauan Pustaka (Lanjutan)**

| No | Nama Peneliti      | Judul   | Permasalahan  | Metode                                 | Hasil  |
|----|--------------------|---|---|--|--|
| 4  | Safitri dkk (2018) | Aplikasi Peminjaman dan Pengembalian <i>Tools</i> di Perusahaan PT Esco Bintan Indonesia Berbasis Web | Dalam operasional kegiatannya masih menggunakan sistem manual dan konvensional, belum di dukung oleh sebuah sistem yang terintegrasi. | Metode Pengembangan Sistem Prototyping | Dengan diterapkannya sistem ini, maka proses pembuatan form peminjaman, faktur pengembalian, data <i>tools</i> sudah terkomputerisasi sehingga dapat menghemat waktu, data lebih akurat, efisien dan data dapat terhubung dengan data pada proses yang lain. |

**Tabael 2.1 Tinjauan Pustaka (Lanjutan)**

| No | Nama Peneliti           | Judul  | Permasalahan  | Metode                          | Hasil   |
|----|-------------------------|--|---|---------------------------------|---|
| 5  | Lase dan Setyadi (2020) | Perancangan Sistem Informasi Peminjaman Dan Pengembalian Inventaris Berbasis Web | Pencatatan peminjaman dan pengembalian inventaris barang di desa siofabanua masih menggunakan proses manual dimana data inventaris desa dicatat dan disimpan pada sebuah buku agenda, data tersebut dapat saja hilang atau pun terselip diakibatkan terlalu banyak data | Metode Pengembangan Sistem SDLC | Dengan menggunakan <i>system</i> pencatatan yang sudah terkomputerisasi diharapkan masalah atau hambatan yang dihadapi dalam <i>system</i> manual dapat teratasi atau mimimalisir kesalahan yang terjadi dalam <i>system</i> manula seperti dalam penyajian informasi yang kurang tepat dan cepat sehingga keakurantan data terjamin. |

## **1.2 Landasan Teori Perancangan**

### **1.2.1 Pengertian Perancangan**

Menuru Mulyanto (2019) Perancangan merupakan penentuan proses dan data yang diperlukan oleh sistem baru. Manfaat tahap perancangan sistem ini memberikan gambaran rancangan bangun yang lengkap sebagai pedoman bagi programmer dalam mengembangkan aplikasi. Sesuai dengan komponen sistem yang dikomputerisasikan, maka yang harus didesain dalam tahap ini mencakup *hardware* atau *software*, *database* dan aplikasi.

Menurut Sommerville dalam buku Agus Mulyanto (2019) proses perancangan bisa melibatkan pengembangan beberapa model sistem pada tingkat abstraksi yang berbeda-beda. Menurut Soetam Rizky (2018) perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya. Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa perancangan adalah tahapan setelah analisis sistem yang tujuannya untuk menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama tahap analisis.

### **1.2.2 Tahapan Perancangan**

Tahapan perancangan sistem adalah merancang sistem dengan terperinci berdasarkan hasil analisis sistem, sehingga menghasilkan model *system* baru (Mahdiana, 2021). Berikut tahapan-tahapan perancangan sistem menurut pendapat Mahdiana :

### 1 Perancangan *Output*

Perancangan *output* tidak dapat diabaikan, karena laporan yang dihasilkan harus memudahkan bagi setiap unsur manusia yang membutuhkan.

### 2 Perancangan *Input*

Tujuan dari perancangan *input* yaitu dapat mengefektifkan biaya pemasukan data, mencapai keakuratan yang tinggi, dan dapat menjamin pemasukan data yang akan diterima dan dimengerti oleh pemakai.

### 3 Perancangan Proses Sistem

Tujuan dari perancangan proses *system* adalah menjaga agar proses data lancar sehingga dapat menghasilkan informasi yang benar dan mengawasi proses dari sistem.

### 4 Perancangan *Database*

*Database* sistem adalah mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

### 5 Tahapan Perancangan Kontrol

Tujuan perancangan ini agar keberadaan sistem setelah diimplementasikan dapat memiliki kehandalan dalam mencegah kesalahan, kerusakan, serta kegagalan proses sistem.

## **1.2.3 Tujuan Perancangan**

Tujuan perancangan menurut Koniyo (2018) antara lain:

1. Memenuhi spesifikasi fungsional.
2. Memenuhi batasan-batasan media target implementasi, target sistem komputer.

3. Memenuhi kebutuhan-kebutuhan implisit dan eksplisit berdasarkan kinerja dan penggunaan sumber daya.
4. Memenuhi perancangan implisit dan eksplisit berdasarkan bentuk hasil rancangan yang dikehendaki.
5. Memenuhi keterbatasan-keterbatasan proses perancangan seperti lama atau biaya.
6. Untuk memberikan gambaran yang jelas dan rancangan bangun yang lengkap kepada pemrogram komputer dan teknik ahli lainnya yang terlibat.
7. Untuk tercapainya pemenuhan kebutuhan berkaitan dengan pemecahan masalah yang menjadi sasaran pengembangan sistem.
8. Untuk kemudahan dalam proses pembuatan *software* dan control dalam mengembangkan sistem yang dibangun.
9. Untuk kemaksimalan solusi yang diusulkan melalui pengembangan sistem.
10. Untuk dapat mengetahui berbagai elemen spesifik pendukung dalam pengembangan sistem baik berupa perangkat lunak maupun perangkat keras yang digunakan pada sistem yang didesain.

### **1.3 Pengertian Sistem**

Pengertian sistem menurut Romney dan Steinbart (2018) sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

Pengertian sistem menurut Diana dan Setiawati (2018), Sistem merupakan “serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu”.



Definisi sistem menurut Mulyadi (2018), Sistem adalah “suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

### **1.3.1 Tujuan Sistem**

Adapun tujuan sistem menurut Susanto (2019) Target atau sasaran akhir yang ingin dicapai oleh sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai sasaran tanpa mengetahui ciri-ciri atau kriteria dari sasaran tersebut kemungkinan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian.

## **1.4 Pengertian Informasi**

Pengertian Informasi menurut Mulyanto (2019), informasi adalah “data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata”.

Hal serupa disampaikan oleh Romney dan Steinbart (2018) Informasi (information) adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peningkatan informasi.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian informasi adalah data yang diolah agar bermanfaat dalam pengambilan keputusan bagi penggunanya.

### **1.5 Pengertian Sistem Informasi**

Pengertian menurut Kadir (2018), Sistem informasi adalah “sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai”.

Definisi menurut Diana dan Setiawati (2018) Sistem informasi, yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Hal serupa juga disampaikan oleh Laudon (2019) yang mendefinisikan sistem informasi : Secara teknis sebagai sesuatu rangkaian yang komponen-komponennya saling terkait yang mengumpulkan (dan mengambil kembali), memproses, menyimpan dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan mengendalikan perusahaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

### **1.5.1 Ciri-Ciri Sistem Informasi**

Berikut ciri-ciri sistem informasi menurut Ardana dan Lukman (2018):

1. Satu Kesatuan: Satu-Kesatuan organisasi
2. Bagian-Bagian: ada manajemen, karyawan, pemangku kepentingan (*stakeholder*) lainnya, gedung kantor, sub-sistem komputer (perangkat keras, perangkat lunak, perangkat jaringan, sumber dayam manusia, basis data dan informasi)
3. Terjalin erat: tercermin dalam bentuk hubungan, interaksi, prosedur kerja antar manajemen
4. Mencapai tujuan: menghasilkan informasi yang berkualitas bagi manajemen dan pemangku kepentingan lainnya.

### **1.5.2 Pengertian Sistem Informasi Berbasis Komputer**

Pengertian menurut Mardi (2018), Sistem komputerisasi akuntansi merupakan “aplikasi dari *accounting system* yang berbasis sistem *database* dengan menggunakan teknologi komputer”.

Pengertian menurut Weygant dkk (2018) dalam sistem akuntansi terkomputerisasi., ada program-program yang digunakan dalam menjalankan siklus akuntansi, seperti penjurnalan, posting (pembukuan) dan penyusunan neraca saldo. Dalam sistem yang terkomputerisasi, jurnal dan buku besar dapat dicatat dalam basis data (*database*) komputer. Lebih jauh lagi, telah ada *software* untuk menjalankan sistem bisnis seperti fungsi penagihan, fungsi penyiapan penggajian dan fungsi penganggaran.

## **1.6 Peminjaman**

Menurut Oktaviani (2019), peminjaman atau sewa yaitu sebagai persetujuan untuk pemakaian sementara suatu benda, baik bergerak maupun tidak bergerak, dengan perjanjian tertentu yang telah disepakati.

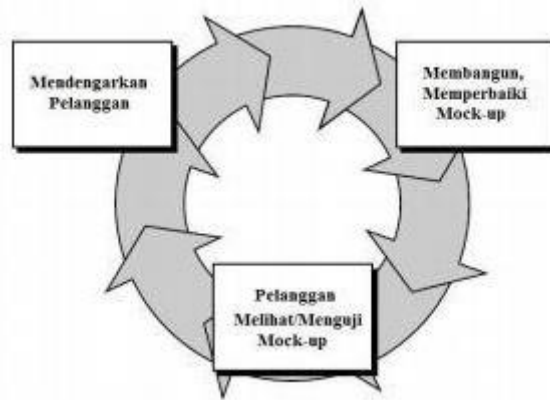
## **1.7 Inventaris**

Menurut Oktaviani (2019), Inventaris merupakan simpanan barang-barang mentah, material atau barang jadi yang disimpan untuk digunakan dalam masa mendatang atau dalam kurun waktu tertentu. Inventaris merupakan proses mengelola pengadaan atau persediaan barang yang dimiliki oleh suatu kantor atau Perusahaan dalam melakukan kegiatan operasionalnya.

Dengan demikian penulis dapat menyimpulkan bahwa inventaris adalah proses pengadaan maupun persediaan barang yang dimiliki oleh suatu kantor atau perusahaan dalam melakukan operasional baik digunakan dalam masa mendatang maupun kurun waktu tertentu.

## **1.8 Metode Pengembangan Sistem *Prototype***

Menurut Pressman dan Bruce (2015), Model *Prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembangan perangkat lunak. Model *protipe* (*prototyping* model) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. *Prototype* biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Berikut adalah gambar dari model *prototype*:



**Gambar 2.1 Model Prototype**

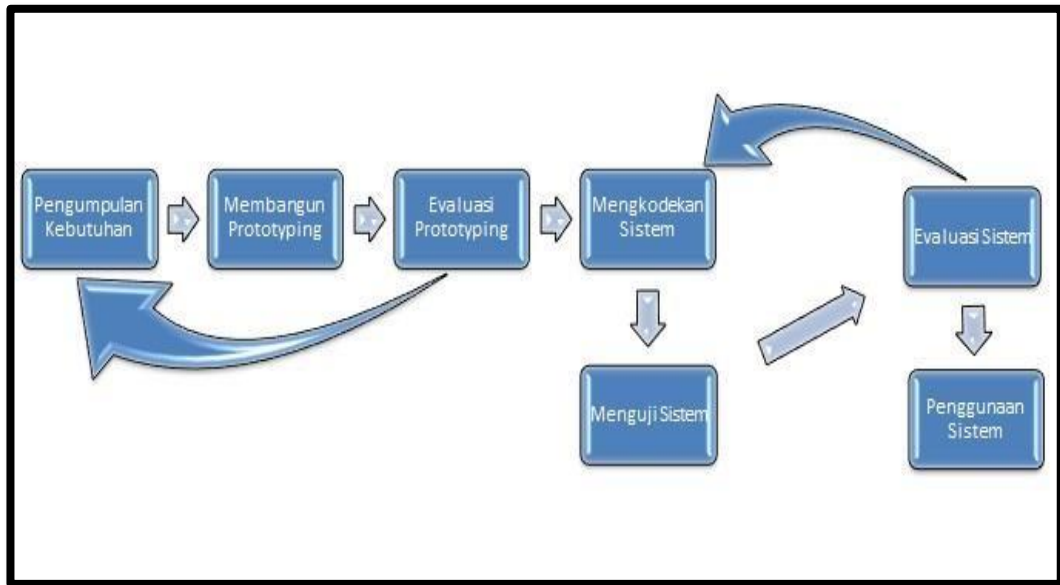
Sumber : Pressman dan Bruce (2015)

Dari gambar diatas dapat dijelaskan bawahwa :

1. Tentukan kebutuhan dengan mendengarkan *user* atau pelanggan. Analisis *system* mewawancarai *user* untuk mendapatkan ide tentang apa yang diinginkan atau yang didengarkan oleh *user* dari *system* yang akan dikembangkan.
2. Buat *prototype* atau *Mockup*. Analisis *system* bekerja sama dengan ahli komputer yang lain, dengan memanfaatkan satu atau beberapa alat bantu untuk pembuatan *prototype*, mengembangkan *prototype*.
3. Evaluasi setelah *user* atau pelanggan melihat atau menguji *Mockup*. Analisis *system* memperkenalkan *prototype* kepada *user*, menuntun *user* untuk mengenali karakteristik dari *prototype*.

### **2.8.2 Tahapan Dalam Prototyping**

Menurut Pressman dan Bruce (2015) Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut :



**Gambar 2.2 Proses *Throw-away Prototyping***

Sumber : Pressman dan Bruce (2015)

Dari gambar diatas dapat dijelaskan bawahwa :

1. Pengumpulan kebutuhan, pelanggan dan *developer* bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun *Prototyping*, membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan format *output*).
3. Evaluasi *Prototyping*, evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulangi langkah 1, 2, dan 3.
4. Mengkodekan sistem, dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji sistem, setelah sistem sudah menjadi suatu perangkat lunak yang

siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.

6. Evaluasi sistem, pelanggan mengevaluasi apakah sistem yang sudah jadi telah sesuai dengan yang diharapkan.
7. Menggunakan sistem, perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

## **1.9 Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak). UML merupakan notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, namun secara normal digunakan untuk pemodelan sistem komputer. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Rosa, 2019).

### **1.9.1 Diagram UML**

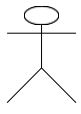


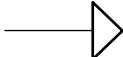
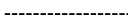
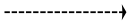
Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

#### **1. Use Case Diagram**

*Use case* atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*, sedangkan *use*

*case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client* (Rosa, 2019).

**Tabel 2.2 Simbol-simbol Use Case Diagram**

| No. | Notasi                | Keterangan  | Simbol  |
|-----|-----------------------|---|---|
| 1.  | <i>Actor</i>          | Pengguna sistem atau yang berinteraksi langsung dengan sistem, bisa manusia, aplikasi, ataupun objek lain   |    |
| 2.  | <i>Use Case</i>       | Digambarkan dengan lingkaran elips dengan nama <i>use case</i> nya tertulis ditengah lingkaran  |    |
| 3.  | <i>Association</i>    | Digambarkan dengan sebuah garis yang berfungsi menghubungkan <i>actor</i> dengan <i>use case</i> .  |    |
| 4   | <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.  |  |
| 5   | <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.   |  |
| 6   | <i>Include</i>        | Relasi <i>use case</i> 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 dijalankannya <i>use case</i> ini. |  |

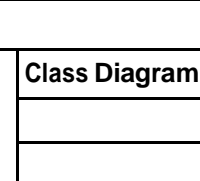
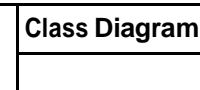
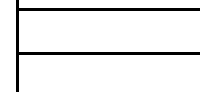

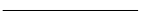
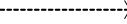

**Sumber : (Rosa : 2019)**



## 2. *Class Diagram*

*Class* adalah deskripsi kelompok obyek-obyek dengan *property*. *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem (Rosa, 2019).

**Tabel 2.3 Simbol *Class Diagram***

| No. | Notasi            | Keterangan   | Simbol  |
|-----|-------------------|--|---|
| 1.  | <i>Object</i>     | Blok-blok pembangun program. Bagian atas <i>class</i> menunjukkan namadari <i>class</i> , bagian tengah mengindikasikan atribut dari <i>class</i> , dan bagian bawah mendefinisikan <i>method</i> dari sebuah <i>class</i> |    |
|     |                   |  |    |
|     |                   |  |    |
|     |                   |  |    |
| 2.  | <i>Activation</i> | Menunjukkan <i>relationship</i> atau hubungan antar <i>class</i>   |  |
| 3.  | <i>Lifeline</i>   | Menunjukkan ketergantungan antara satu <i>class</i> dengan <i>class</i> yang lain  |  |
| 4.  | <i>Message</i>    | Menunjukkan <i>inheritance</i> dari satu <i>class</i> ke beberapa <i>class</i>   |  |






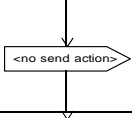
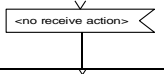



Sumber : (Rosa : 2019)


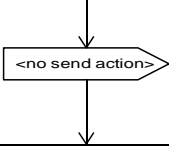
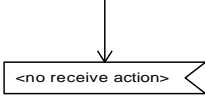


## 3. *Activity Diagram*

*Activity Diagram* ini menggambarkan rangkaian aliran dari aktivitas dari sebuah sistem atau proses bisnis. *Activity Diagram* adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran penting seperti halnya *flowchart*, akan

tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung paralel sedangkan *flowchart* tidak bisa (Rosa, 2019).

**Tabel 2.4 Simbol Activity Diagram**

| No. | Notasi               | Keterangan  | Simbol  |
|-----|----------------------|---|---|
| 1.  | <i>Initial State</i> | Titik awal untuk memulai suatu aktivitas  |    |
| 2.  | <i>Final State</i>   | Titik akhir untuk mengakhiri suatu aktivitas  |    |
| 3.  | <i>Activity</i>      | Menandakan sebuah aktivitas   |    |
| 4.  | <i>DeciSistemon</i>  | Pilihan untuk pengambilan keputusan   |    |
| 5.  | <i>Fork/Join</i>     | Menunjukkan kegiatan menggabungkan dua panel activity menjadi satu atau satu panel activity menjadi dua |  |
| 6.  | <i>Send</i>          | Tanda pengiriman  |  |
| 7.  | <i>Receive</i>       | Tanda penerimaan  |  |
| 8.  | <i>ControlFlow</i>   | Arus aktivitas  |  |
| 9.  | <i>Note</i>          | Catatan khusus untuk sebuah aktivitas   |  |
| 10. | <i>DeciSistemon</i>  | Pilihan untuk pengambilan keputusan   |  |

| No.  | Notasi             | Keterangan  | Simbol  |
|------|--------------------|---|---|
| 11.  | <i>Fork/Join</i>   | Menunjukkan kegiatan menggabungkan dua panel <i>activity</i> menjadi satu atau satu panel <i>activity</i> menjadi dua |    |
| 12.  | <i>Send</i>        | Tanda pengiriman  |    |
| 13.  | <i>Receive</i>     | Tanda penerimaan  |    |
| 14.  | <i>ControlFlow</i> | Arus aktivitas  |    |
| 15.. | <i>Note</i>        | Catatan khusus untuk sebuah aktivitas   |  |

Sumber : (Rosa : 2019)

### 1.10 Pengertian Website

Pengertian *website* menurut Sebok dkk (2018) adalah kumpulan halaman yang saling terhubung yang di dalamnya terdapat beberapa item seperti dokumen dan gambar yang tersimpan di dalam web *server*. Web app adalah sebuah aplikasi yang berada dalam web *server* yang bisa *user* akses melalui browser. *Web app* biasanya menampilkan data *user* dan informasi dari *server*. sejak awal 1990, *world wide web* atau *website* merevolusi kehidupan pribadi maupun professional. Web menjadi situs yang terus berkembang dan sebagai perpustakaan informasi yang ada di mana-mana yang dapat diakses melalui mesin pencari dan portal. Web menjadi tempat penyimpanan media yang memfasilitasi hosting dan berbagi sumber daya

yang sering kali gratis dan sebagai pendukung layanan *do-it-yourself*. Web juga menjadi platform perdagangan tempat orang dan perusahaan semakin menjalankan bisnisnya.

Beberapa tipe *website* menurut Sebok dkk (2018), yaitu :

1. *Search Engines*

*Search engine* adalah perangkat lunak yang menemukan situs web, halaman web, gambar, video, berita, peta, dan informasi lain yang berkaitan dengan topik tertentu.

2. *News, Weather, Sports, and Other Mass Media*

Situs *website* ini berisi materi yang layak diberitakan termasuk cerita dan artikel yang berkaitan dengan kejadian terkini, kehidupan, uang, politik, cuaca, dan olahraga.

3. *Educational*

*Website educational* menawarkan jalan yang menarik dan menantang untuk pengajaran dan pembelajaran formal dan informal. Instruktur sering menggunakan web untuk menyempurnakan pengajaran di kelas dengan menerbitkan materi, nilai, dan informasi kelas terkait lainnya.

4. *Business, Governmental, and Organizational*

Merupakan *website* yang berisi konten yang meningkatkan kesadaran merek, memberikan latar belakang perusahaan, dan mempromosikan produk atau layanan. Hampir setiap perusahaan memiliki situs web bisnis.

5. *Banking and Finance*

*Online banking* dan *online trading* memungkinkan *user* untuk mengakses catatan keuangan mereka dari mana saja selama memiliki koneksi internet.

Dengan menggunakan *online banking*, *user* dapat mengakses akun, membayar tagihan, mentransfer dana, dan mengelola aktivitas keuangan lainnya. Dengan *online trading*, *user* dapat berinvestasi di saham atau pasar uang tanpa menggunakan broker.

#### 6. *Travel and Tourism*

*Travel and tourism website* memungkinkan *user* untuk mencari opsi perjalanan dan membuat pengaturan perjalanan. *User* dapat membaca *review* perjalanan, mencari dan membandingkan harga penerbangan, memesan maskapai penerbangan, kamar, atau mobil sewaan.

#### 7. *E-Commerce*

*E-Commerce* merupakan transaksi bisnis yang terjadi melalui jaringan elektronik. Beberapa orang menggunakan istilah *M-Commerce* atau *mobile commerce* untuk mengidentifikasi *E-Commerce* yang terjadi menggunakan perangkat *mobile*. Penggunaan *E-Commerce* yang sering dijumpai yaitu belanja dan lelang, keuangan, perjalanan, hiburan, dan kesehatan.

Sebagian besar halaman web menyertakan multimedia, yang mengacu pada aplikasi yang menggabungkan teks dengan media. Media ini meliputi :

##### 1. Grafik

Grafik adalah representasi visual dari informasi nonteks, seperti gambar, bagan, atau foto. Sebuah *website* sering menggunakan infografis untuk menyajikan konsep, produk, dan berita. Infografis adalah representasi visual dari data atau informasi dengan menggunakan grafik dan diagram. Format grafik yang sering digunakan untuk menampilkan gambar dalam sebuah *website* adalah format *JPEG* dan *PNG*.

## 2. Animasi

Animasi adalah kemunculan gerak yang dibuat dengan menampilkan rangkaian gambar diam secara berurutan. Contohnya teks yang dianimasikan dengan menggulir melintasi layar dapat berfungsi sebagai *sticker* untuk menampilkan informasi.

## 3. Audio

*Audio* dalam sebuah *website* mencakup musik, ucapan, atau suara lainnya. Suatu file *audio* dikompresi untuk mengurangi ukuran filenya. *Format audio* yang paling umum adalah *MP3* karena format ini mengurangi *file audio* menjadi sekitar sepersepuluh dari ukuran aslinya dan tetap mempertahankan sebagian besar kualitas suara aslinya.

## 4. Video

Video terdiri dari gambar-gambar yang diputar dalam gerakan. *User* dapat mengupload, berbagi, atau melihat klip video di situs *website*. File video sering kali dikompresi karena ukurannya yang cukup besar. Video yang diposting ke sebuah *website* biasanya berdurasi pendek sekitar kurang dari sepuluh menit.

### 1.11 Pengertian *PHP*

Menurut Susanti (2018), *PHP* merupakan singkatan dari *Personal Home Page Hypertext Processor*. Atau biasa disebut *PHP* merupakan bahasa *script* yang ditempatkan dalam *server* dan diproses di *server* hasilnya dikirimkan ke klien, tempat pemakainya menggunakan browser. Seperti bahasa pemrograman yang lain, *PHP* memiliki kelebihan dan juga kelemahan. Adapun kelebihan dari *PHP* antara lain :

1. *PHP* merupakan suatu bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *PHP* dapat berjalan pada web *server* yang dirilis oleh *Microsoft*, juga pada *Apache* yang bersifat open source.
3. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreted pada *PHP* lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembangannya.
4. *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.

*PHP* dapat berjalan pada 3 operating *system*, yaitu linux, Unix dan Windows, dan juga dapat dijalankan secara runtime pada suatu console.

### **1.12 Kualitas Perangkat Lunak ISO/IEC 25010**

Menurut Wattiheluw (2019), ISO/IEC merupakan standar yang digunakan oleh dunia internasional untuk melakukan evaluasi atau pengukur-an kualitas dari perangkat lunak. ISO/IEC yang digunakan dalam penelitian ini adalah versi 25010 yang merupakan versi lanjutan dari ISO/IEC 25010 dengan penambahan beberapa struktur dan bagian dari standar model kualitas. Secara keseluruhan ISO/IEC 25010 memiliki 8 karakteristik untuk mengukur kualitas perangkat lunak secara menyeluruh, antara lain portability, performance efficiency, reliability, security usability, maintainability, compatibil-ity, dan functional suitability. Adapun beberapa definisi karakteristik ISO/IEC 25010 adalah sebagai berikut :

1. Functional suitability adalah produk aplikasi yang memberikan fungsional untuk memenuhi kebutuhan saat menggunakan produk dalam keadaan tertentu.

2. Reliability adalah tingkat dimana produk aplikasi dapat mempertahankan kinerja pada level tertentu ketika digunakan dalam keadaan tertentu.
3. Performance efficiency adalah tingkat dimana produk aplikasi menyediakan performa yang baik dengan jumlah resource yang digunakan.
4. Usability adalah dimana produk aplikasi mudah dimengerti, dipakai dan menarik untuk digunakan.
5. Security adalah tingkat produk aplikasi menyediakan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, atau pengungkapan yang berbahaya.
6. Compatibility adalah kemampuan dari suatu komponen aplikasi atau lebih untuk bertukar informasi.
7. Maintainability adalah tingkat dimana produk aplikasi dapat dimodifikasi. Modifikasi yang dilakukan dapat meliputi perbaikan, pengembangan atau adaptasi perangkat lunak untuk menyesuaikan dengan lingkungan, serta modifikasi pada kriteria dan spesifikasi fungsi.
8. Portability adalah tingkat dimana produk aplikasi dapat dipindahkan dari satu ruang ke ruang lain.

### **1.13 Black Box Testing**

Menurut Rosa dan Shalahuddin (2019), *black box testing* adalah 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. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai



dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black box testing* harus dibuat dengan kasus benar dan kasus salah.

Menurut Nurajizah dan Aziz (2019), *black box testing* juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan inisialisasi dan akhir program.

#### 1. *Equivalence Partitioning*

*Equivalence Partitioning* merupakan metode *black box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga test case dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *boolean*.

Kesetaraan kelas dapat didefinisikan menurut panduan berikut:

1. Jika masukan kondisi menentukan kisaran, satu sah dan dua diartikan tidak valid kesetaraan kelas.
2. Jika masukan membutuhkan nilai, kondisi tertentu satu sah dan dua tidak valid kesetaraan kelas diartikan.

3. Jika masukan kondisi menentukan anggota dari set, satu sah dan satu tidak valid kesetaraan kelas diartikan.
4. Jika kondisi yang *input, boolean* satu sah dan satu tidak valid kelas diartikan. Sebagai contoh, pemeliharaan data untuk aplikasi bank yang sudah diotomatisasikan. Pemakai dapat memutar nomor telepon bank dengan menggunakan mikro komputer yang terhubung dengan *password* yang telah ditentukan dan diikuti dengan perintah-perintah. Data yang diterima adalah :

1. Kode Area : kosong atau 3 digit
2. *Prefix* : 3 digit atau tidak diawali 0 atau 1
3. *Suffix* : 4 digit
4. *Password* : 6 digit alfanumerik
5. Perintah : *check, deposit*, dll

Selanjutnya kondisi masukan digabungkan dengan masing-masing data elemen, dapat ditentukan sebagai berikut :

1. Kode Area : kondisi masukan, *Boolean*-kode area mungkin ada atau tidak. Kondisi masukan, kisaran nilai ditentukan antara 200-999.
2. *Prefix* : kondisi masukan kisaran lebih besar 200 atau tidak diawali 0 atau 1.
3. *Suffix* : kondisi masukan nilai 4 digit.
4. *Password* : kondisi masukan *Boolean*-pw mungkin diperlukan atau tidak. Kondisi masukan nilai dengan 6 karakter string.
5. Perintah : kondisi masukan diatur dengan berisi perintah-perintah yang telah didefinisikan.

Menerapkan pedoman untuk derivasi kelas kesetaraan, uji kasus untuk setiap masukan domain item data dapat dikembangkan dan dilaksanakan. Uji kasus dipilih

sehingga jumlah terbesar dari atribut dari kelas kesetaraan tersebut dilakukan sekaligus.

Beberapa kata kunci dalam pengujian perangkat lunak yang dapat diperhatikan, yaitu:

1. Dinamis

Pengujian perangkat lunak dilakukan pada masukan yang bervariasi. Masukan ini ditentukan sebelum pengujian dilakukan dengan batasan yang disesuaikan dengan kemampuan perangkat lunak. Masukan tidak harus sesuatu yang dimungkinkan terjadi pada penggunaan program lebih lanjut, melainkan meliputi keseluruhan batasan yang dapat dijangkau perangkat lunak dan dilakukan pemercontohan (*sampling*) secara acak untuk proses pengujian.

2. Terbatas

Meskipun pengujian dilakukan pada perangkat lunak sederhana sehingga rumit sekalipun, pengujian dilakukan dengan memenuhi batasan-batasan tertentu sesuai dengan kemampuan program. Batasan ini juga diberlakukan pada masukan-masukan yang dipilih untuk pengujian. Tidak semua kemungkinan masukan diujika pada perangkat lunak karena akan memakan waktu yang cukup panjang mengingat begitu banyaknya kemungkinan yang bisa terjadi. Untuk mengatasi hal ini, pemilihan masukan-masukan pada proses pengujian secara acak yang diperkirakan mampu memenuhi kebutuhan pengujian perangkat lunak akan dilakukan.

3. Tertentu

Pengujian dilakukan dengan batasan tertentu disesuaikan dengan harapan pada fungsi, respon, dan karakteristik perangkat lunak tersebut. Batasan tersebut akan disesuaikan dengan teknik-teknik pengujian yang ada. Pemilihan kriteria pengujian yang paling tepat merupakan hal yang kompleks. Dalam praktiknya, analisis risiko pengujian dan pengalaman terhadap pengujian-pengujian sejenis akan diperlukan.

4. Harapan

Kata kunci ini memiliki keadaan-keadaan yang diharapkan, baik berupa respon sistem terhadap masukan maupun karakteristik responnya. Dalam hal ini, batasan-batasan hasil pengujian yang diharapkan harus ditentukan. Dengan demikian, dapat diketahui apakah perangkat lunak tersebut telah memenuhi hasil pengujian yang diharapkan atau memerlukan pembenahan kembali, baik berupa perbaikan maupun pengembangan perangkat lunak.

Menurut Achyani dan Saumi (2019), tujuan dari pengujian adalah untuk menemukan dan memperbaiki sebanyak mungkin kesalahan dalam program sebelum menyerahkan program kepada *customer*. Salah satu pengujian yang baik adalah pengujian yang memiliki probabilitas tinggi dalam menemukan kesalahan.

Format tabel pengujian *black box* dapat dilihat pada Tabel 2.5.

**Tabel 2.5 Format Pengujian *Black Box***

| <b>Kasus dan Hasil Pengujian</b> |                        |                   |                   |
|----------------------------------|------------------------|-------------------|-------------------|
| <b>Data Masukan</b>              | <b>Yang Diharapkan</b> | <b>Pengamatan</b> | <b>Kesimpulan</b> |
|                                  |                        |                   | [ ] Diterima      |
|                                  |                        |                   | [ ] Ditolak       |