

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu :

1. Berdasarkan penelitian yang dilakukan oleh Eka K, dkk: 2018 yang berjudul “aplikasi layanan jasa penyedia jasa garden pemilik gedung dan lahan serta penyedia jasa dapat menghemat waktu, tenaga dan juga biaya dalam proses pencarian informasi dan pemesanan jasa garden untuk rumah tangga dan kantor secara *online*, dikarenakan system pemasaran yang mudah dan *user friendly* dimana pelanggan dapat memesan jasa pengelola taman secara *online* lewat aplikasi jasa - *garden* yang akan dirancang berbasis *Mobile*. Metode yang digunakan dalam perancangan ini adalah menggunakan *Unified Modeling Language* (UML). Aplikasi ini juga dibuat dengan menggunakan bahasa pemrograman PHP menggunakan *MySQL* sebagai basis data. Dengan dirancangnya aplikasi jasa - *garden* tersebut dapat membantu para calon pelanggan dalam menemukan jasa tukang kebun secara *online*. Begitu juga dengan penyedia jasa dapat dengan mudah menemukan pelanggan dalam menggunakan jasa layanan yang disediakan. Dengan aplikasi ini juga dapat membantu setiap penyedia jasa dalam menawarkan dan mempublikasikan jasa nya”.
2. Berdasarkan penelitian yang dilakukan oleh Ferdi I, dkk: 2020 yang berjudul “Aplikasi Penyedia Layanan Jasa Fotografer Dan Videografer Berbasis

*Android* merupakan sebuah aplikasi yang digunakan untuk menyediakan layanan bagi para fotografer dan videografer untuk memasarkan dan mengelola jasanya. Penggunaan aplikasi ini diharapkan dapat mempermudah bagi penyedia jasa dan orang yang akan memesan untuk menentukan kapan pelayanan akan dilakukan. Aplikasi ini dirancang menggunakan UML (*Unified Modelling Language*) dan dibuat menggunakan bahasa pemrograman *java android* dan *MySQL* untuk mengolah *database*, yang mana aplikasi ini diharapkan dapat membantu dan mempermudah pihak yang bersangkutan.

3. Berdasarkan penelitian yang dilakukan oleh Ika I, dkk: 2020 yang berjudul “Efektifitas Penggunaan Aplikasi *Eclipse Crossword* pada peningkatan hasil belajar siswa kelas IV SD Negeri 56 Tolobali Kota Bima” adalah Penelitian ini bertujuan untuk mengetahui : peningkatan hasil belajar siswa kelas VI SD Negeri 56 Kota Bima yang diajar menggunakan media teka teki silang *Eclipse crossword* dan yang diajar menggunakan media konvensional, dan keefektifan penggunaan media teka teki silang *Eclipse crossword*. Penelitian ini merupakan quasi experimental, jenisnya *nonequivalent control group*. Populasi Penelitian ini adalah seluruh siswa kelas VI SD Negeri kecamatan Rasana’E Barat yang berjumlah 11 sekolah dengan jumlah siswa 600 orang.
4. Berdasarkan penelitian yang dilakukan oleh P.A. Ginanjar, dkk : 2017 yang berjudul “Aplikasi Pemesanan Bus Pariwisata Menggunakan *Payment Gateway* Berbasis *Android* (Studi Kasus: Bluestar, Salatiga)”, P.O Bluestar

adalah salah satu perusahaan penyedia bus pariwisata di Kota Salatiga, Jawa Tengah Indonesia. P.O Bluestar merupakan perusahaan jasa yang bergerak pada bidang transportasi bus pariwisata yang melayani jasa bus antar kota antar provinsi sehingga menjadi salah satu alat transportasi yang diminati oleh masyarakat. Banyaknya permintaan jasa persewaan bus pariwisata di P.O Bluestar membuat banyak pihak perantara seperti calo yang dapat merugikan calon pelanggan. Selain itu, proses pemesanan menggunakan telepon atau datang langsung ke kantor perusahaan menyebabkan kurang efisiennya waktu bagi para calon pelanggan. Sehingga dengan paparan masalah tersebut maka pemesanan secara *online* lebih efektif untuk menghindari calo serta lebih praktis dan tanpa harus mengeluarkan biaya untuk datang menuju P.O secara langsung.

5. Berdasarkan penelitian yang dilakukan oleh Fajar I, dkk : 2020 yang berjudul “Aplikasi pemesanan jasa cukur rambut berbasis *android*”, Peluang usaha dalam bidang jasa cukur rambut merupakan peluang bagi para pelaku bisnis untuk dapat memenuhi permintaan yang berkaitan dengan penampilan. Kepuasan dan kepercayaan pelanggan menjadi hal utama dalam bisnis jasa cukur rambut. Umumnya pelayanan yang diberikan oleh pihak penyedia jasa cukur rambut masih bersifat konvensional belum adanya sistem pemesanan dalam artian pelanggan terkadang mengantri terlalu lama di lokasi tempat cukur dan bisa berdampak buruk bagi pengusaha jasa cukur rambut, karena adanya kemungkinan pelanggan pergi ke tempat cukur lain. Pemanfaatan teknologi informasi terkini dapat digunakan untuk membangun sistem yang

dapat menunjang dalam pemesanan jasa cukur rambut agar lebih mudah. Perancangan sistem berbasis *android* dirasa perlu untuk mempermudah pemesanan jasa cukur rambut. Dalam penelitian ini dikembangkan sebuah aplikasi pemesanan jasa cukur rambut berbasis *android* untuk mengurangi antrian pada penyedia jasa cukur rambut. Berdasarkan hasil pengujian sistem menggunakan pengujian ISO 25010 mendapatkan nilai sangat baik atau sistem dapat diterima dan dijalankan dengan baik.

6. Berdasarkan penelitian yang dilakukan oleh Ferdi I, dkk : 2020 yang berjudul “Rancang Bangun Aplikasi Penyedia Layanan Jasa Fotografer Dan Videografer Berbasis *Android*”, Aplikasi Penyedia Layanan Jasa Fotografer Dan Videografer Berbasis *Android* merupakan sebuah aplikasi yang digunakan untuk menyediakan layanan bagi para fotografer dan videografer untuk memasarkan dan mengelola jasanya. Jasa fotografer dan videografer sering kali dibutuhkan untuk mendokumentasikan sebuah acara ataupun hal - hal yang dianggap sangat penting. Dengan aplikasi ini orang - orang akan lebih mudah menemukan jasa fotografer dan videografer. Penggunaan aplikasi ini diharapkan dapat mempermudah bagi penyedia jasa dan orang yang akan memesan untuk menentukan kapan pelayanan akan dilakukan. Aplikasi ini dirancang menggunakan UML (*Unified Modelling Language*) dan dibuat menggunakan bahasa pemrograman *java android* dan *MySQL* untuk mengolah *database*, yang mana aplikasi ini diharapkan dapat membantu dan mempermudah pihak yang bersangkutan.

## **II.2. Tinjauan Pustaka**

### **II.2.1. Sistem**

Sistem adalah suatu prosedur atau elemen yang saling berhubungan satu sama lain dimana dalam sebuah sistem terdapat suatu masukan, proses dan keluaran, untuk mencapai tujuan yang diharapkan. Menjadikan sebuah sistem terstruktur lalu menyusun dan mengolah data-data yang digunakan dalam perancangan aplikasi. (Iwan Laengge, dkk, 2016).

### **II.2.2 Android**

Android adalah program siap pakai yang dapat digunakan untuk menjalankan perintah - perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. (Agusdi S, 2018; 50).

### **II.2.3. Pelayanan Jasa**

Pelayanan yang sistematis merupakan layanan yang menentukan dalam menindaklanjuti komplain konsumen dari suatu kegagalan sehingga pada akhirnya mampu mengikat loyalitas konsumen, layanan sangat krusial dalam

mempertahankan pelanggan dalam waktu yang lama, Manusia pada dasarnya menginginkan pemuas kebutuhan. Dan salah satu hal untuk memuaskan kebutuhan manusia adalah pelayanan. Mendapatkan pelayanan dengan kualitas terbaik adalah hal yang diinginkan setiap manusia. (Heni R; 2018; 313 ).

Pelayanan jasa merupakan upaya pemenuhan kebutuhan dan keinginan konsumen serta ketepatan penyampaianya mengimbangi harapan konsumen. Pelayanan yang baik mempengaruhi kepuasan konsumen yang berdampak terjadinya pembelian berulang - ulang yang berarti akan terjadinya peningkatan penjualan. Dengan pelayanan yang baik dapat menciptakan kepuasan dan loyalitas konsumen serta membantu jarak dengan pesaing. (Nurlina, 2019:2).

Pelayanan atau jasa adalah kinerja dari pelayanan yang diterima oleh konsumen itu sendiri dan konsumen akan menilai kualitas pelayanan atau jasa yang benar - benar mereka rasakan, ukuran yang berdasarkan kinerja (*service performance*) akan lebih merefleksikan kualitas pelayanan atau jasa karena pengukuran terhadap kualitas pelayanan dalam *service quality* yang diajukan telah membentuk paradigma yang lemah dimana harapan konsumen atau pasien terhadap kualitas jasa mengacu kepada harapan konsumen terhadap penyedia jasa secara umum, sedangkan persepsi kinerja pelayanan atau jasa mengarah kepada kepuasan pelayanan atau jasa yang dituju. Hal ini diperkuat dengan pendapat para ahli bahwa kinerja pelayanan akan menjadi prediktor yang baik bagi kualitas pelayanan jasa. (Mas Ayoe, 2018:125).

#### II.2.4. Eclipse

*Eclipse* adalah sebuah *IDE (Integrated Development Environment)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform (platform-independent)*. Berikut ini adalah sifat dari *Eclipse* :

- a. *Multi-platform* : Target sistem operasi *Eclipse* adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X*.
- b. *Mult-language* : *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *Java, C/C++, Cobol, Python, Perl, PHP*, dan lain sebagainya.
- c. *Multi-role* : Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan *web*, dan lain sebagainya. *Eclipse* pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *opensource* ( Ipan R, 2020 ; 3 ).
- d. Bahasa pemrograman *java* adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *Java* tidak hanya terfokus oada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

#### II.2.5. Database

Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lain. Basis data atau *database* merupakan

salah satu komponen yang penting dalam sistem informasi, karena berfungsi berfungsi sebagai basis penyedia informasi bagi pemakainya, Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan dengan yang lainnya dan untuk membuatnya tersedia beberapa aplikasi yang bermacam - macam dalam suatu sistem organisasi. Sistem basis data adalah suatu sistem menyusun dan mengelola *record - record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan(Priyo S; 2016 : 25).

#### **II.2.6. MySQL**

*MySQL* adalah salah satu jenis database server yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. *MySQL* termasuk jenis RDBMS (*Relational Database Management System*). Pada *MySQL*, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom(Priyo S; 2016 : 25).

#### **II.2.7. UML (Unified Modelling Language)**

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena



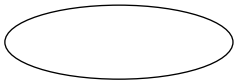
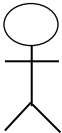
UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam membentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar ; 2018 : 49).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

#### 1. *Use case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan tipikal interaksi antara (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem yang dipakai (Munawar ; 2018 : 89).

**Tabel II.1. Simbol *Use Case Diagram***

| Gambar                                                                              | Keterangan                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit - unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>Use Case</i> .                                                                                                                       |
|  | Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas - tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor |

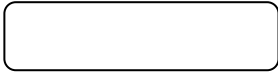
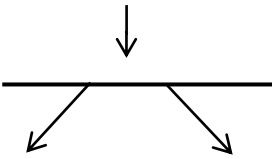
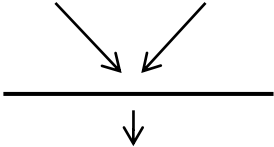
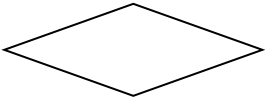
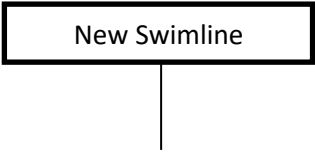
|                                                                                     |                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                     | berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki <i>control</i> terhadap <i>Use Case</i> .                                                                                                 |
|    | Asosiasi antara aktor dan <i>Use Case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data. |
|    | Generalisasi untuk menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .                                                                                                 |
|  | <i>Include</i> , merupakan di dalam <i>Use Case</i> lain ( <i>required</i> ) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.          |
|  | <i>Extend</i> , merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.                                                                                                     |

(Sumber : Munawar ; 2018 : 89)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram* (Munawar ; 2018 : 137).

Tabel II.2. Simbol Diagram Aktivitas

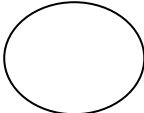
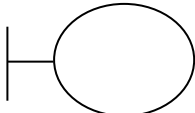
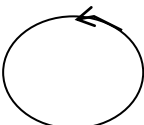
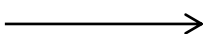
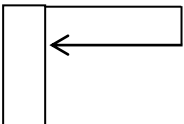
| Gambar                                                                              | Keterangan                                                                                                                                                |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.                                                                        |
|    | <i>End point</i> , akhir aktifitas.                                                                                                                       |
|    | <i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.                                                                                            |
|   | <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | <i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.                                                                     |
|  | <i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .                                                  |
|  | <i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.                                                                |

(Sumber : Munawar ; 2018 : 137)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* adalah salah satu *interaction diagram*. Karena *sequence diagram* mengacu kepada obyek, maka sebelum membuat diagram ini class diagram sudah harus teridentifikasi (Munawar ; 2018 : 186).

**Tabel II.3. Simbol Diagram Urutan**

| Gambar                                                                              | Keterangan                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | <i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas - entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.      |
|  | <i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak. |
|  | <i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.  |
|  | <i>Message</i> , simbol mengirim pesan antar <i>class</i> .                                                                                                                                     |
|  | <i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.                                                                                                           |
|  | <i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi                                                                               |

|                                                                                   |                                                                                                                           |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
|                                                                                   | aktivitas sebuah operasi.                                                                                                 |
|  | <i>Lifeline</i> , garis titik - titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |

(Sumber : Munawar ; 2018 : 186)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap - tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan - aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* merupakan diagram statis dari suatu aplikasi. *Class Diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak (Munawar ; 2018 : 101).

**Tabel II.4. Simbol *Class Diagram***

| Multiplicity | Penjelasan                                                      |
|--------------|-----------------------------------------------------------------|
| 1            | Satu dan hanya satu                                             |
| 0..*         | Boleh tidak ada atau 1 atau lebih                               |
| 1..*         | 1 atau lebih                                                    |
| 0..1         | Boleh tidak ada, maksimal 1                                     |
| n..n         | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4 |

(Sumber : Munawar ; 2018 : 101)

### II. 3. Bahasa Pemrograman (*java & Kotlin*)

*Java* adalah bahasa pemrograman yang populer, dikembangkan oleh *Sun Microsystems*. Salah satu penggunaan terbesar *Java* adalah dalam pembuatan aplikasi *native* untuk *android*. Bahasa pemrograman ini bersifat *multi platform* yakni bahasa ini dapat digunakan di berbagai platform, seperti desktop, *android* dan bahkan untuk sistem operasi Linux. Beberapa ciri dari bahasa pemrograman ini adalah sebagai berikut :

1. *Object oriented language*
2. *Multithreading*
3. *Garbage collector support*
4. *Statically Typed*
5. *Multiplatform*. (Niko S,2018:320)

Sedangkan *Kotlin* adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh *JetBrains*. *Kotlin* merupakan bahasa pemrograman yang pragmatis untuk *android* yang mengkombinasikan *object oriented* (OO) dan pemrograman fungsional. *Kotlin* juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman *Java*. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis *desktop*, *web* dan bahkan untuk *backend*. Beberapa keuntungan yang mungkin akan didapatkan

jika pengembangan aplikasi beralih menggunakan *Kotlin* untuk mengembangkan aplikasi diatas platform JVM adalah sebagai berikut:

1. Dapat mengatasi *NullPointerException* yang umumnya terdapat pada *Java*.
2. Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa *Java*.
3. Mudah dipelajari.
4. Dukungan IDE untuk mempermudah dalam pemrograman (Niko S,2018:320).