

## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Pada penelitian yang peneliti lakukan dengan judul “Rancang Bangun Aplikasi *Booking* dan *Loyalty Program* Berbasis Android di Grafika Studio”. Peneliti mengambil beberapa referensi dari penelitian terdahulu sebagai acuan dalam melakukan penelitian agar dapat mengembangkannya menjadi sesuatu yang baru. Berikut ini adalah beberapa penelitian yang peneliti kaji sebelumnya.

Penelitian yang dilakukan oleh Aditya Dwi Saputra dan Rohmat Indra Borman dengan judul ‘Sistem Informasi Pelayanan Jasa Foto Berbasis Android (Studi Kasus: Ace Photography Way Kanan)’ bertujuan merancang sistem yang dapat membantu perusahaan untuk menunjang sistem pelayanannya. Permasalahan dalam penelitian ini yaitu sistem pengelolaan data pelayanan jasanya masih dilakukan secara manual yaitu dicatat kedalam nota sehingga sering terjadinya kerusakan dan kehilangan nota. Untuk mengatasi permasalahan tersebut dibuatkan sebuah sistem informasi pelayanan jasa foto dengan harapan dapat memudahkan pelanggan yang ingin menggunakan jasa foto perusahaan tersebut [9].

Selanjutnya, penelitian yang dilakukan Hans Fernaldi Wijaya dengan judul ‘Rancang Bangun Aplikasi *Customer Relationship Management* Berbasis Android pada PT Rembo Sukses Jaya’ bertujuan membantu perusahaan untuk menciptakan, meningkatkan dan mempertahankan loyalitas dari pelanggan. Permasalahan dalam penelitian ini adalah perusahaan kesulitan dalam meningkatkan kesetiaan pelanggan karena kurangnya interaksi perusahaan dengan pelanggannya. Untuk

mengatasi permasalahan tersebut dibuatkan sebuah sistem yang dapat mengimplementasikan sistem *loyalty program* dengan berupa *membership*, dan pada bagian *front office* akan diterapkan penggunaan *beacon* ketika dalam berinteraksi dengan pelanggan agar membantu dalam sistem operasional CRM-nya [10].

Kemudian penelitian yang dilakukan oleh Tatang Suhidayat, Azhar Affandi, dan Iwan Sidharta yang berjudul “*Service Quality on Customer Value; Perspective from Rural Banking Sectors in Bandung, Indonesia*” bertujuan mengetahui pengaruh kualitas pelayanan terhadap *customer value* di BPR. Penelitian ini dilakukan dengan survei terhadap 29 bank di kota Bandung. Berdasarkan hasil penelitian tersebut menunjukkan bahwa kualitas pelayanan berpengaruh signifikan terhadap *customer value*. Dapat diambil kesimpulan bahwa perlu adanya penerapan kualitas pelayanan yang baik oleh BPR [11].

Dari berbagai jurnal penelitian yang sudah peneliti kaji, beberapa penelitian tersebut menawarkan solusi yang cukup bagus tetapi belum ada solusi yang berfokus untuk memadukan sistem pelayanan *booking* jasa foto studio dengan strategi *marketing* perusahaan dalam mempertahankan *customer*.

Setelah melihat kelebihan dan kekurangan dari penelitian sebelumnya, maka peneliti berusaha memberikan solusi yang terbaik yang bersumber dari penelitian sebelumnya. Maka dari itu peneliti memberikan sebuah solusi yaitu membuat aplikasi *booking* dan *loyalty program* berbasis android untuk menciptakan inovasi dalam sistem pelayanan serta memberikan strategi *marketing* yang tepat dengan memprioritaskan *customer retention* yaitu mempertahankan *customer* lama dengan harapan *customer* lama tersebut dapat melakukan advokasi

atau rekomendasi kepada orang lain bahwa layanan yang dimiliki oleh Grafika Studio ini bonafit dan patut untuk dicoba sehingga dapat berdampak pada *marketing* secara organik untuk mendapatkan *customer* baru.

## 2.2 Teori-teori Dasar

Berikut ini adalah teori-teori dasar yang digunakan dalam penelitian ini untuk mendukung hasil penelitian menjadi lebih baik

### 2.2.1 Aplikasi

Aplikasi merupakan program siap guna yang ditujukan untuk menjalankan suatu fungsi yang dapat memudahkan pengguna untuk mencapai tujuannya. Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan [12]. Menurut Kamus Besar Bahasa Indonesia, “Aplikasi merupakan perangkat lunak atau program komputer yang dirancang dengan menggunakan bahasa pemrograman tertentu dan bertujuan untuk mengerjakan suatu tugas tertentu” [13].

Dalam penelitian ini peneliti bermaksud untuk membuat sebuah aplikasi *booking* dan *loyalty program* yang dibuat untuk membantu perusahaan dalam menyediakan sistem pelayanan yang *user-friendly* dan memudahkan calon *customer* ataupun *customer* yang ingin melakukan *booking* foto. Aplikasi ini juga didukung dengan fitur *loyalty program* berupa *loyalty point* yang dapat dikumpulkan oleh *customer* setiap kali selesai melakukan transaksi dan dapat ditukarkan dengan *reward*. Aplikasi ini dirancang dengan menggunakan bahasa pemrograman *java* untuk *android* nya dan *php* untuk halaman dashboard admin dalam mengelola konten dan transaksi yang terjadi pada aplikasi android.

### **2.2.2 Booking**

*Booking* atau dalam bahasa indonesianya berarti reservasi merupakan proses perjanjian yang berupa pemesanan produk barang ataupun jasa namun belum ditutup dengan sebuah transaksi jual beli. Proses reservasi ditandai dengan adanya proses tukar menukar informasi antara produsen dan konsumen yang bertujuan untuk menciptakan adanya kesepahaman mengenai produk ataupun jasa yang ingin dipesan [12].

Dalam penelitian ini aplikasi yang akan dibangun terdapat fasilitas *booking* yang bertujuan memudahkan *customer* ketika ingin memesan layanan foto tanpa harus lama mengantri.

### **2.2.3 Customer Realtionship Management (CRM)**

CRM (*Customer Relationship Management*) merupakan sistem informasi skala perusahaan yang diperuntukan bukan hanya sebagai perangkat lunak melainkan sebagai strategi bisnis yang mengintegrasikan proses bisnis dengan para konsumen (penjualan, pelayanan dan pemasaran) dengan pelaku bisnis, proses bisnis dan teknologi pendukung guna memperoleh *customer retention* [14]. *Customer Relationship Model* merupakan perpaduan tiga aspek yang saling berkaitan. Apabila digambarkan membentuk segitiga dengan simpul *Customer – Relation – Management*. Berikut penjelasan ketiga aspek tersebut :

#### **1. Customer**

Merupakan pengguna layanan barang atau jasa dari suatu perusahaan yang menghasilkan *profit* atau keuntungan bagi perusahaan tersebut.

## **2. *Relationship***

Merupakan hubungan antara *customer* dengan perusahaan dapat berupa interaksi atau komunikasi baik terjadi dalam jangka waktu hanya sekali, jangka pendek, jangka panjang, atau bahkan terus menerus. Apabila *customer* telah memiliki rasa percaya atau puas terhadap layanan yang diberikan oleh perusahaan tersebut maka biasanya *customer* akan menggunakan kembali produk atau jasa yang ditawarkan oleh perusahaan tersebut

## **3. *Management***

Merupakan segala jenis informasi atau data yang didapatkan dari *customer* yang membuat perusahaan dapat memetakan bagaimana perilaku atau kebiasaan *customer*, sehingga dapat digunakan sebagai nilai tambah dari perusahaan untuk meningkatkan pelayanan agar dapat bersaing dengan kompetitornya.

Dalam penelitian ini aplikasi yang dibangun akan menerapkan konsep CRM sebagai acuan dalam membangun sistem yang dapat membantu meningkatkan sistem pelayanan perusahaan kepada *customer*-nya.

### **2.2.4 *Customer Retention***

*Customer retention* merupakan kemampuan perusahaan untuk mengubah *customer* menjadi *repeat customer* dan mencegah mereka untuk beralih ke kompetitor. Hal ini menunjukkan apakah produk/jasa yang dimiliki oleh perusahaan dapat menyenangkan *customer* yang sudah ada. Strategi *customer retention* ini merupakan proses dan inisiatif yang dilakukan bisnis untuk

membangun loyalitas *customer* dan meningkatkan nilai perusahaan bagi *customer* [15].

*Customer retention* berbeda dengan *customer acquisition*, yang mana *customer retention* berfokus kepada *customer* yang telah mendaftar layanan ataupun menggunakan produk dari suatu perusahaan. Mempertahankan kepuasan *customer* umumnya lebih hemat biaya daripada harus mendapatkan pelanggan pertama kali. Menurut *Harvard Business Review*, mendapatkan *customer* baru dapat menghabiskan 5 hingga 25 kali biaya lebih mahal daripada memertahankan *customer* yang sudah ada [15].

Adapun keuntungan yang bisa didapatkan dengan menerapkan *customer retention* pada perusahaan, diantaranya :

a) Hemat Biaya

*Customer retention* umumnya lebih hemat biaya daripada memperoleh *customer* baru

b) *Marketing* secara organik dari mulut ke mulut

*Customer loyal* lebih cenderung merekomendasikan kepada teman, keluarga ataupun orang terdekat mereka untuk menggunakan produk/jasa yang dimiliki oleh perusahaan tersebut.

c) Meningkatkan *retention rate*

Dengan meningkatkan *customer retention rate* sebesar 5% dapat meningkatkan pendapatan sebesar 25 – 95%.

Untuk menghitung *customer retention rate* dapat dilakukan dengan cara menghitung persentase *customer* lama yang tetap bertahan dalam periode waktu tertentu kemudian dihitung menggunakan formula seperti berikut :

$$((E - N / S) * 100 = X$$

- Jumlah *customer* pada akhir periode (E)
- Jumlah *customer* baru yang didapat selama periode tersebut (N)
- Jumlah *customer* pada awal periode (S)
- *Customer retention rate* (X)

Dalam penelitian ini peneliti memberikan inovasi dalam strategi *marketing* yaitu sebagai upaya untuk meningkatkan *customer retention rate* perusahaan dengan menerapkan sistem pelayanan *booking* berbasis aplikasi dan juga disertakan *reward* sebagai upaya membangun loyalitas *customer* agar terus menggunakan jasa yang dimiliki oleh perusahaan.

### **2.2.5 Loyalty Program**

*Loyalty program* adalah alat pemasaran yang dirancang untuk meningkatkan loyalitas dengan memberikan insentif kepada konsumen berupa keuntungan yang bisa didapatkan [16]. Secara luas *loyalty program* dapat dibagi menjadi tiga kategori, diantaranya dapat berupa :

- a) *Gift cards* (kartu hadiah) dengan pembelian minimum
- b) Diskon member
- c) Undian atau kupon

*Loyalty program* dapat meningkatkan kepuasan *customer* sehingga ketika terjadi masalah hubungan antara *customer* dengan penyedia layanan dalam hal ini perusahaan maka *loyalty program* dapat mengurangi ketidakpuasan tersebut. Hal ini mengingatkan betapa pentingnya *loyalty program* dalam membangun hubungan dengan *customer*. *Loyalty program* yang baik tidak hanya dapat meningkatkan dari sisi penjualan saja tetapi juga dapat mendorong partisipasi *customer* dalam

peningkatan interaksi *loyalty program* antara bisnis dengan *customer*. Selain itu *customer* juga akan percaya bahwa mereka bisa mendapatkan nilai ekonomi yang lebih tinggi, *reward* yang beragam, dan juga kesempatan mendapatkan hadiah yang mereka inginkan dengan ikut turut terlibat dalam *loyalty program* tersebut.

Dalam penelitian ini peneliti membangun aplikasi yang didalamnya terdapat *loyalty program* berupa *loyalty point* yang didapatkan setiap kali selesai melakukan transaksi dan dapat ditukarkan dengan *reward*, hal ini bertujuan untuk memberikan inovasi dalam *strategi marketing* kepada perusahaan agar dapat mempertahankan *customer*-nya dan berdampak pada *marketing* secara organik.

#### **2.2.4 Android**

Android adalah kumpulan *software* untuk perangkat *mobile* yang didalamnya terdapat sistem operasi, *middleware*, dan *key application*. Android menjadi sebuah sistem operasi populer yang dikembangkan oleh perusahaan Google dengan basis kernel Linux dan juga berbagai perangkat lunak seperti *open source* dan lainnya [17]. Aplikasi yang dikembangkan untuk *platform* android umumnya menggunakan bahasa pemrograman Java. Aplikasi yang dibangun dengan menggunakan *platform* android dapat dikembangkan lebih variatif dan inovatif karena *platform* Android ini bersifat *open source* yang sama halnya dengan sistem operasi Linux dengan varian distro-nya.

Android memiliki banyak *library* yang variatif dan fiturnya dapat digunakan oleh pengembang untuk memodifikasi aplikasi agar dapat memenuhi kebutuhan pengguna. Berbagai fitur tersebut antara lain :

- a) *Open Graphic Library (Open GL)*, merupakan *Application Program Interface (API)* yang digunakan untuk membuat grafis baik 2D ataupun 3D.



- b) Android *run-time*, terdiri dari library *Dalvik Virtual Machine* dan Java
- c) WebKit, adalah *engine* dari *web browser* yang digunakan untuk menyederhanakan tampilan dari proses *loading* dan juga menampilkan isi *website*.
- d) SQLite, adalah *engine* dari *relational database* yang dapat diintegrasikan dengan aplikasi.

Dalam proses perkembangannya sistem operasi Android telah mengalami banyak perubahan, dan berikut adalah urutan versi Android dari sejak awal dibuat hingga yang terbaru :

- a) Android 1.0 & 1.1 : Astro (Alpha) & Bender (Beta)
- b) Android 1.5 : Cupcake
- c) Android 1.6 : Donut
- d) Android 2.0 & 2.1 : Éclair
- e) Android 2.2 : Froyo (Frozen Yoghurt)
- f) Android 2.3 : Gingerbread
- g) Android 3.0 & 3.2 : Honeycomb
- h) Android 4.0 : Ice Cream Sandwich
- i) Android 4.1 & 4.3 : Jelly Bean
- j) Android 4.4 : Kitkat
- k) Android 5.0 & 5.1 : Lollipop
- l) Android 6.0 : Marshmallow
- m) Android 7.0 & 7.1 : Nougat
- n) Android 8.0 & 8.1 : Oreo
- o) Android 9.0 : Pie

- p) Android 10
- q) Android 11
- r) Android 12

Google terus melakukan pengembangan terhadap sistem operasi Android karena mengikuti kebutuhan teknologi yang semakin tinggi dan juga membantu pengembang dalam membangun aplikasi yang lebih variatif dengan didukung banyaknya API atau *library* yang disediakan.

Dalam penelitian ini peneliti membuat aplikasi berbasis Android karena memiliki beberapa kelebihan, salah satunya yaitu sistem antarmuka yang mudah digunakan karena tidak akan membutuhkan waktu lama untuk beradaptasi dalam penggunaan Android. Selain itu Android memiliki sifat *open source* yang dapat memudahkan pengembang dalam membangun aplikasi *booking* dan *loyalty program* agar lebih baik dan dapat memenuhi kebutuhan pengguna.

#### **2.2.4 Metode Pengembangan *Prototype***

Metode pengembangan *prototype* merupakan metode pengembangan perangkat lunak berupa gambaran kerja sistem dan berfungsi sebagai versi awal dari sistem yang akan dibangun [18]. Dengan menerapkan metode ini hasil yang diharapkan berupa *prototype* sistem sebagai perantara pengguna dengan pengembang untuk dapat saling berinteraksi dalam proses pengembangan sistem. Metode *prototype* bertujuan mengumpulkan informasi dari pengguna sehingga dapat berinteraksi dengan model *prototype* yang dibangun.

Adapun langkah-langkah dalam melakukan metode pengembangan *prototype* sebagai berikut :

1. Mengumpulkan Kebutuhan Pengguna

Dalam mengumpulkan kebutuhan pengguna akan melibatkan pertemuan antara pengembang dan pengguna dalam menentukan tujuan dan kebutuhan dibuatnya aplikasi atau sistem ini.

## 2. Membuat Desain

Setelah mendapatkan kebutuhan sistem dari pengguna kemudian dibuatkan desain yang mengarah ke pembangunan *prototype* yang akan dihasilkan mencakup seperti *input*, proses dan *output*-nya.

## 3. Membangun *Prototype*

Pada tahap ini desain yang sudah jadi kemudian dibuatkan *prototype*-nya agar dapat berjalan sebagaimana aplikasi atau sistem yang sudah jadi namun masih versi awal untuk selanjutnya dilakukan pengujian.

## 4. Melakukan Evaluasi dan Perbaikan

*Prototype* yang sudah dibangun tersebut kemudian di ujicoba kepada pengguna apakah sudah berjalan dengan baik dan memenuhi kebutuhan pengguna atau belum, setelah itu barulah dapat dilakukan perbaikan sebelum dikembangkan menjadi versi aplikasi yang sudah jadi.

## 5. Mengembangkan Aplikasi

Pada tahapan ini *prototype* yang sudah jadi dan telah dilakukan perbaikan sebelumnya, maka mulai masuk kedalam tahap pengembangan menjadi aplikasi agar dapat menjadi produk nyata yang dapat digunakan oleh pengguna.

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

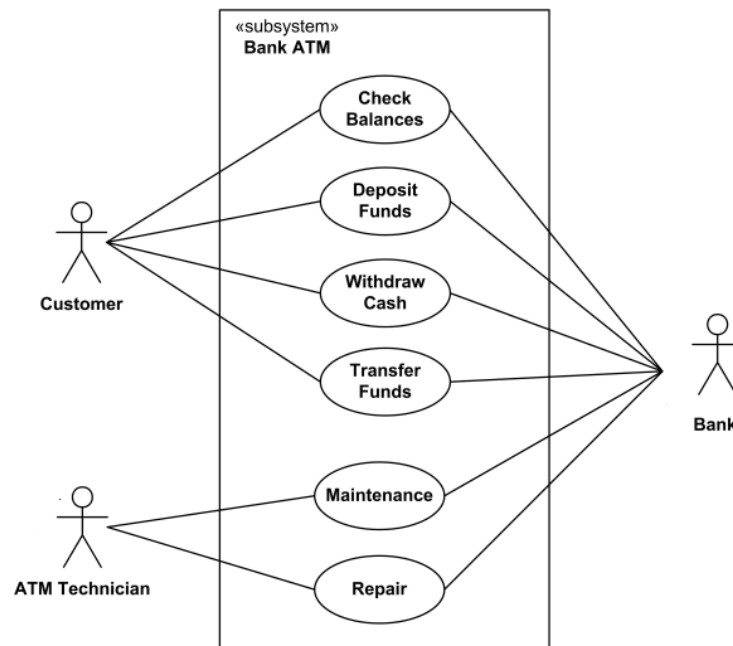
UML merupakan salah satu alat bantu dalam pengembangan sistem berorientasi objek atau biasa disebut juga bahasa pemodelan sistem yang digunakan

untuk menggambarkan sistem dalam bentuk grafis yang bertujuan untuk merinci, menggambarkan dan membangun sebuah sistem agar dapat dengan mudah dipahami oleh manusia atau mesin. UML memiliki konsep berorientasi objek yang bertujuan membantu manusia atau pengembang memahami sistem yang akan dibuat berupa visualisasi grafis [19].

UML memiliki beberapa diagram untuk melakukan pemodelan data atau sistem. Berikut adalah macam-macam diagram UML:

1. *Use Case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem serta bagaimana sistem tersebut dapat berinteraksi dengan pengguna dan menjabarkan fungsionalitas sistem yang dapat terlihat oleh pengguna [20]. Pada *use case diagram* yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *Use case diagram* hanya menjabarkan pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.



**Gambar 2. 1 Contoh Use Case Diagram**

**(Sumber : Apa itu UML? Beserta Pengertian dan Contohnya [21])**

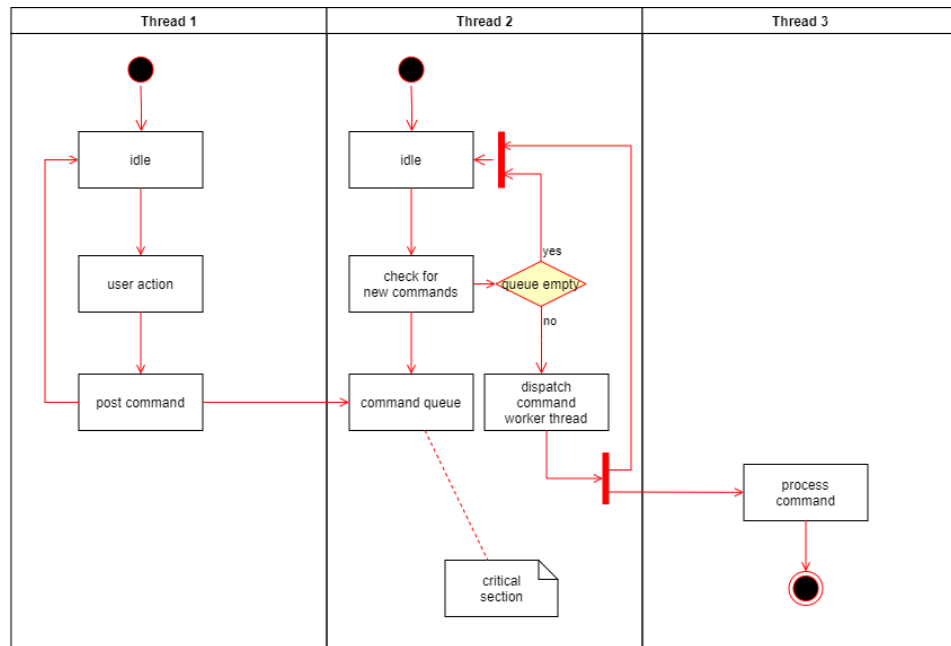
## 2. Use Case Scenario

*Use case scenario* adalah suatu gambaran model yang berbentuk skenario yang menjelaskan berupa tahapan-tahapan yang dilakukan oleh aktor dalam melakukan suatu aktivitas [20]. Dalam menentukan kemungkinan-kemungkinan yang dapat terjadi ketika menjalankan sebuah sistem tersebut nantinya akan dipecah menjadi beberapa skenario. Di dalam proses sebuah sistem nantinya akan ada sekumpulan proses, proses inilah yang akan menjadi beberapa kemungkinan yang kemudian bisa disebut dengan *use case scenario*.

## 3. Activity Diagram

*Activity diagram* adalah diagram yang menggambarkan alur atau jalan suatu aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir [20].

*Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

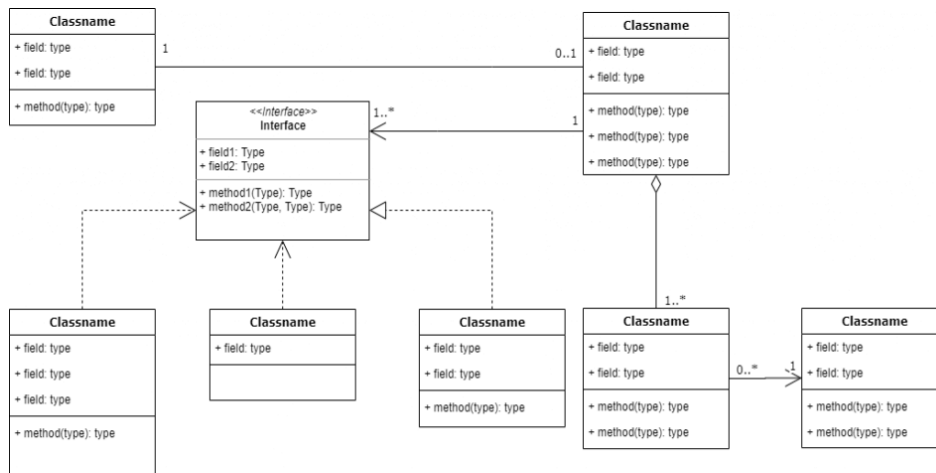


**Gambar 2. 2 Contoh Activity Diagram**

(Sumber : Apa itu UML? Beserta Pengertian dan Contohnya [21])

#### 4. Class Diagram

*Class diagram* merupakan spesifikasi yang apabila diturunkan dapat menghasilkan sebuah objek atau *class* yang merupakan inti dari pengembangan sistem yang berorientasi pada objek. Di dalam sebuah *class* terdapat atribut atau properti dari sebuah sistem dan juga terdapat layanan untuk manipulasi keadaan tersebut berupa metode atau fungsi [20].



**Gambar 2. 3 Contoh Class Diagram**

**(Sumber : Apa itu UML? Beserta Pengertian dan Contohnya [21])**

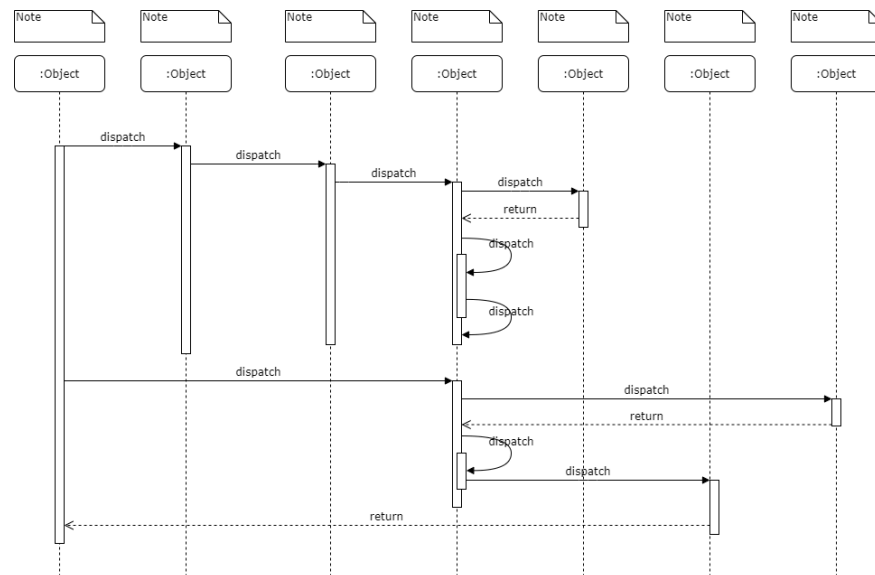
### 5. Use Case Scenario

*Use case scenario* adalah suatu gambaran model yang berbentuk skenario yang menjelaskan berupa tahapan-tahapan yang dilakukan oleh aktor dalam melakukan suatu aktivitas [20]. Dalam menentukan kemungkinan-kemungkinan yang dapat terjadi ketika menjalankan sebuah sistem tersebut nantinya akan dipecah menjadi beberapa skenario. Di dalam proses sebuah sistem nantinya akan ada sekumpulan proses, proses inilah yang akan menjadi beberapa kemungkinan yang kemudian bisa disebut dengan *use case scenario*.

### 6. Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan interaksi antar objek di dalam sistem seperti pengguna, *view*, dan sebagainya berupa *class* atau objek yang digambarkan terhadap waktu. Dalam menentukan kemungkinan-kemungkinan yang dapat terjadi ketika menjalankan sebuah sistem tersebut nantinya akan dipecah menjadi beberapa skenario. *Sequence*

*diagram* dapat menggambarkan tahapan demi tahapan yang harus dilakukan untuk dapat menghasilkan *output* tertentu [20].

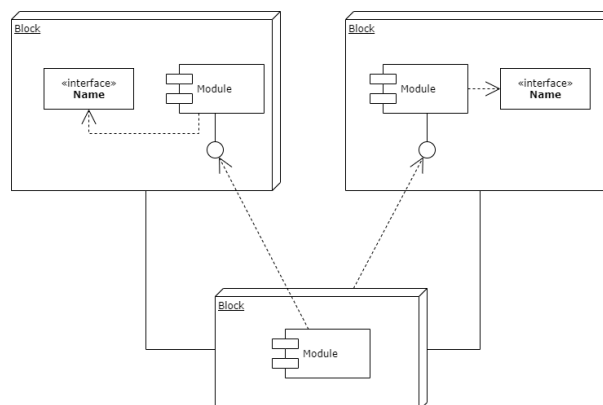


**Gambar 2. 4 Contoh Sequence Diagram**

(Sumber : Apa itu UML? Beserta Pengertian dan Contohnya [21])

#### 7. Deployment Diagram

*Deployment diagram* merupakan *diagram* yang berfungsi untuk menggambarkan detail komponen dari suatu infrastruktur sistem yang dibangun [20]. Komponen-komponen tersebut biasanya bersifat fisikan seperti letak server, mesin, atau perangkat keras, dan lain sebagainya.



**Gambar 2. 5 Contoh Deployment Diagram**

(Sumber : Apa itu UML? Beserta Pengertian dan Contohnya [21])



### 2.2.6 Basis Data

Basis data atau yang biasa diebut dengan *database* merupakan kumpulan data yang disimpan di dalam komputer yang disusun secara sistematis sehingga nantinya dapat digunakan ataupun dicari oleh program komputer untuk mendapatkan informasi dari *database* tersebut. Dalam mengelola basis data dibutuhkan sebuah DBMS atau *Database Management System* untuk memanggil *query* dari basis data tersebut. Singkatnya *database* merupakan kumpulan data yang saling berelasi dan diorganisasikan berdasarkan struktur tertentu [22].

### 2.2.7 Testing Perangkat Lunak

Dalam proses membangun suatu sistem atau aplikasi tentu harus melewati suatu tahapan yaitu *testing* atau pengujian *software*. Hal ini penting dilakukan untuk mengetahui *bug* maupun *error* dari setiap fungsi yang dijalankan oleh aplikasi tersebut. Tahapan pengujian ini penting dilakukan agar *error* dan *bug* yang ditemukan dapat segera diperbaiki sebelum digunakan oleh pengguna. Dalam melakukan *testing* perangkat lunak terdapat dua metode yaitu :

1. *Black Box Testing*

*Black box testing* merupakan salah satu metode *testing* perangkat lunak yang berfokus pada *interface* dan menguji bagaimana cara aplikasi beroperasi dengan memasukan berupa *input* data dan melihat bagaimana *output* yang dihasilkan apakah sesuai dengan yang diharapkan atau tidak [23]. *Black box testing* dapat digunakan untuk menemukan kesalahan seperti fungsi yang tidak benar atau hilang, kesalahan dari antarmuka aplikasi, kesalahan dari struktur data atau akses basis data dan lain-lain.

2. *White Box Testing*

*White box testing* merupakan metode testing perangkat lunak yang berfokus pada bagaimana cara kerja suatu perangkat lunak yang dilihat dari logika program yang berjalan dengan menyediakan *test case* secara detail [23]. *White box testing* ini mengacu pada bagaimana program yang ada dalam perangkat lunak tersebut harus betul-betul benar secara 100%.

### **2.3 Software yang Digunakan**

Dalam melakukan penelitian ini peneliti menggunakan beberapa *software* untuk membangun sistem yang dirancang guna terselesaikannya penelitian ini dengan baik. Berikut ini adalah *software* yang peneliti gunakan untuk membangun aplikasi *booking* dan *loyalty program* berbasis android di Grafika Studio .

#### **2.3.1 Android Studio**

Android studio merupakan *Integrated Development Environment* atau yang biasa disebut IDE untuk pengembangan aplikasi android. Android studio dirilis pertama kali pada tanggal 16 mei 2013 oleh Google pada saat event Google I/O *Conferenece*. Android studio resmi menggantikan Eclipse sebagai IDE resmi dalam pengembangan aplikasi android dan bersifat *open source* [24].

Android studio dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan IDE sebelumnya yaitu Eclipse dan dilengkapi dengan *Android Development Tools*. Android studio dikembangkan dengan didukung beberapa fitur diantaranya :

- a) *Refactory* untuk pencarian *bug*
- b) GUI (*Graphical User Interface*) aplikasi android yang mudah
- c) *Support* terhadap *Google Cloud Platfrom* pada setiap aplikasi yang dibangun
- d) Pengembangan projek berbasis Gradle Build

- e) Terdapat *Proguard and App-signing* untuk *security*

Pada penelitian ini peneliti menggunakan Android Studio sebagai *tools* utama dalam membangun aplikasi atau sistem ini. Alasan peneliti memilih Android Studio karena merupakan *software* yang *open source* dan juga terdapat banyak fitur serta *library* yang menunjang peneliti untuk membangun aplikasi yang dapat memenuhi kebutuhan pengguna.

### 2.3.2 Java

Java adalah salah satu bahasa pemrograman yang sangat populer dan menggunakan metode berbasis objek atau biasa dikenal dengan *Object Oriented Programming* (OOP). Bahasa pemrograman Java menjadi sangat populer karena rentang aplikasi yang dapat dikembangkan dengan bahasa pemrograman ini sangat luas baik itu untuk aplikasi komputer maupun *smartphone*. Java merupakan bahasa pemrograman yang dapat ditulis satu kali dan dapat berjalan pada *multi platform* atau biasa disebut dengan “*Write Once, Run Anywhere*” [25].

Adapun beberapa kelebihan yang dimiliki oleh bahasa pemrograman Java ini yaitu diantaranya :

- a) *Object Oriented Programming* (OOP)
- b) Sederhana agar mudah dipelajari
- c) Dinamis
- d) Memiliki performa tinggi karena menggunakan *compiler* langsung
- e) *Multithreaded* atau mampu membuat program yang bisa mengerjakan beberapa tugas sekaligus

Dengan melihat beberapa kelebihan tersebut peneliti mempertimbangkan untuk menggunakan bahasa pemrograman Java sebagai *tools* untuk membangun sistem yang dirancang.

### **2.3.3 Java Development Kit (JDK)**

*Java Development Kit* atau yang biasa dikenal dengan JDK merupakan kumpulan *software* yang dapat digunakan untuk mengembangkan aplikasi berbasis Java. Selain itu JRE merupakan implementasi dari Java Virtual Machine yang digunakan untuk menjalankan program java. Setiap JDK akan berisi satu atau lebih JRE dan berbagai alat pengembangan untuk aplikasi lainnya seperti *development libraries, compiler java, debuggers*, dan lain sebagainya [24].

### **2.3.4 XML**

XML atau yang merupakan kepanjangan dari *Extensible Markup Language* bukan merupakan bahasa pemrograman melainkan kumpulan aturan yang digunakan dalam membuat format teks, sehingga format teks yang dibuat lebih terstruktur dan mudah dibaca oleh komputer. Fungsi XML yaitu sebagai media pembawa data atau informasi [26].

Dalam melakukan penelitian ini peneliti menggunakan XML sebagai media untuk menampilkan data dan informasi dari sistem yang dibangun. Selain itu pada *software* yang peneliti gunakan yaitu Android Studio mendukung XML sebagai *tools* dalam membuat *frontend*-nya.

### **2.3.5 XAMPP**

XAMPP merupakan perangkat lunak *open source* yang *membundle* beberapa aplikasi lain yang dibutuhkan dalam pembangunan *website* yang biasanya digunakan untuk membuat server sendiri pada komputer pengembang atau

istilahnya *localhost*. XAMPP merupakan singkatan dari X (*cross-platform*) karena tersedia di banyak sistem operasi, A (Apache), M (MySQL), P (PHP) dan P (Perl). Perangkat lunak ini dapat berjalan di semua sistem operasi [27].

Dalam melakukan penelitian ini peneliti menggunakan XAMPP dengan memanfaatkan Apache dan PHP dalam membangun sistem yang telah peneliti rancang sebelumnya dalam hal ini untuk membuat *website dashboard* admin.

### 2.3.6 PHP

PHP atau singkatan dari *Hypertext Preprocessor* merupakan bahasa pemrograman yang perintahnya dilaksanakan server dan *output*-nya akan di tampilkan pada komputer *client*. PHP merupakan bahasa pemrograman *open source* atau dapat digunakan secara gratis [27].

Peneliti memilih bahasa pemrograman PHP untuk membuat halaman webiste dashboard admin pada sistem yang dibangun karena PHP memiliki beberapa kelebihan diantaranya :

- a) Reliabilitas yaitu PHP didukung oleh dokumentasi yang lengkap, dan memiliki banyak komunitas helpdesk
- b) Integritas dengan *database* yang kebanyakan berjenis *relational* seperti MySQL, SQLite, Oracle, dan lain-lain
- c) Keamanan karena PHP menyediakan tiga jenis autentikasi yaitu HTTP, Cookies, dan Session.

### 2.3.7 CSS

CSS atau singkatan dari *Cascading Style Sheet* merupakan sebuah dokumen yang berupa aturan-aturan yang berfungsi untuk memisahkan isi dengan tata letak pada setiap halaman *website* yang dikembangkan. Dalam pembuatan CSS dikenal

yang namanya *template* yaitu *style* yang dibuat untuk mengizinkan pembuatan kode agar lebih mudah dari setiap halaman *website* yang dirancang [28].

Dalam implemenntasinya CSS akan memisahkan konten utama dengan tampilan seperti *font*, warna dan juga *layout*. Fungsi utama dari CSS ini yaitu untuk mendesain sebuah halaman *website* dengan merubah bentuk serta tampilan secara keseluruhan dengan menempelkannya pada setiap *tag html*. Cara kerja CSS yaitu dengan menempelkan setiap kelas yang telah dibuat pada file terpisah yang sudah memiliki format *css* kemudian hanya tinggal menuliskan pada *selector* di dalam HTML yaitu *class* dan *id*.

### 2.3.8 Bootstrap

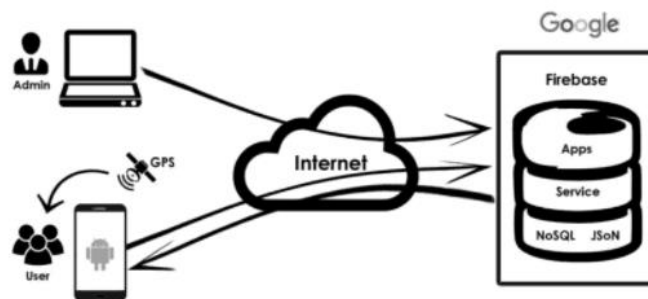
Bootstrap merupakan *front-end framework* yang dibangun dengan menggunakan HTML, CSS, dan Javascript yang digunakan untuk mempercepat dan mempermudah dalam pengembangan *website*. *Bootstrap* mengedepankan tampilan responsif untuk berbagai macam ukuran layar mulai dari komputer, tablet hingga *mobile* [29].

Adapun beberapa kelebihan Bootstrap yang menjadi pertimbangan peneliti untuk menggunakannya dalam penelitian ini yaitu :

- a) *Responsive Layout* untuk berbagai ukuran layar
- b) *Grid System* dengan sistem 12 *column* agar lebih tertata dengan rapi.
- c) Tersedia banyak komponen seperti *form*, *button*, dan lain sebagainya yang sudah siap pakai
- d) Tidak memerlukan *hardisk*, karena dapat digantikan dengan peran *ethernet card* dan *boot lan*

### 2.3.9 Firebase

Firebase merupakan sebuah layanan yang dikembangkan oleh Google yang menyediakan *realtime database* dan *backend as a service*. Pengembang aplikasi nantinya tidak perlu membuat *backend* sendiri karena telah disediakan API oleh Firebase sehingga dapat mempercepat proses pengembangan aplikasi. Firebase dibangun dengan menggunakan database MongoDB sehingga tipe *database*-nya adalah NoSQL, tipe database ini sangat cocok digunakan untuk aplikasi berbasis *mobile* karena sifatnya yang fleksibel [30].



**Gambar 2. 6 Arsitektur Sistem Firebase dengan Android**

(Sumber : Firebase Membangun Aplikasi Berbasis Android [30, p 33])

Dalam melakukan penelitian ini peneliti memilih Firebase karena memiliki beberapa fitur unggulan diantaranya :

- a) Firebase Authentication yang berfungsi mengelola pengguna dengan aman, mudah dan cepat serta layanan ini menerapkan kode *client-side*.
- b) Firebase Realtime Database yang disimpan di dalam *cloud* selain itu layanan ini juga menerapkan API dan datanya disimpan dengan format JSON sehingga dapat disinkronkan secara *realtime* pada setiap pengguna. Apabila terdapat perubahan pada data yang disimpan, maka setiap pengguna yang terhubung akan mendapatkan pembaruan secara *realtime* / otomatis.