

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Untuk mendukung keberhasilan penelitian ini, penulis menggunakan referensi dari jurnal-jurnal dan penelitian terkait untuk membantu penelitian yang dilakukan agar sesuai dengan yang diharapkan.

Sebelumnya telah dilakukan penelitian yang dilakukan oleh Eka Sari Octaviani, Yunita Sari dan Erwin Suhandi, 2018 yang berjudul “**Rancang Bangun Aplikasi Konsultasi Online untuk Gangguan Jiwa Berbasis Web**”. Penelitian ini telah menghasilkan Aplikasi Konsultasi *Online* untuk Gangguan Jiwa berbasis web.

Penelitian yang dilakukan oleh Ilham Firman Maulana, 2020 yang berjudul “**Penerapan Firebase Realtime Database Pada Aplikasi E-Tilang Smartphone berbasis Mobile Android**”. Penelitian ini telah menghasilkan Aplikasi E-Tilang *Smartphone* berbasis *Mobile* Android.

Penelitian yang dilakukan oleh Hilton Haryono, 2020 yang berjudul “**Penerapan Fitur Chat Pada Aplikasi Teman Diabetes Berbasis Mobile**”. Penelitian ini telah menghasilkan Penerapan Fitur Chat Pada Aplikasi Teman Diabetes Berbasis *Mobile*.

Penelitian yang dilakukan oleh Ratih Ayu ninghemi, Atma Deharja, 2017 yang berjudul **“Pengembangan Layanan Aplikasi E- Konsul”**. Penelitian ini telah menghasilkan Pengembangan Layanan Aplikasi E- Konsul.

Penelitian yang dilakukan oleh Jauzaa Maylia Suhendro, Made Sudarma dkk, 2021 yang berjudul **“Rancang Bangun Aplikasi Seluler Penyedia Jasa Perawatan dan Kecantikan Menggunakan Framework Flutter ”**. Penelitian ini telah menghasilkan Aplikasi Seluler Penyedia Jasa Perawatan dan Kecantikan.

## **II.2. Landasan Teori**

Adapun landasan teori yang digunakan sebagai landasan dalam penulisan proposal skripsi ini adalah sebagai berikut :

### **II.2.1. Aplikasi**

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau pemrosesan data yang diharapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms.Word, Ms.Excel (Maiyana, 2018).

### **II.2.2. Konsultasi**

Konsultasi adalah suatu proses yang biasanya didasarkan pada karakteristik hubungan yang sama yang ditandai dengan saling mempercayai dan komunikasi yang terbuka, bekerja sama dalam mengidentifikasi masalah, menyatukan sumber-sumber pribadi untuk mengenal dan memilih strategi yang mempunyai kemungkinan dapat memecahkan masalah yang telah diidentifikasi, dan pembagian tanggung jawab dalam pelaksanaan dan evaluasi program atau strategi yang telah direncanakan. (Rahmatina Septiani, Sheilla, 2019)

### **II.2.3. Chatting**

*Chatting* adalah suatu feature pada suatu program yang ada di Internet untuk berkomunikasi atau berkonsultasi langsung sesama dengan pengguna internet lain yang dalam hal ini mereka diuntut *online* dan sama-sama menggunakan Internet. Yang berarti definisi *chatting* sendiri ialah suatu pesan cepat (instan) ataupun *instant messaging* pada teknologi baik *mobile* maupun *pc* yang saling terhubung sehingga bisa bertukar informasi. (Nur Riza Ziaraharja, Ari Kurniawan, 2020).

#### II.2.4. Android

Menurut Hermawan dalam Maiyana (2018), Android merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh Google pada tahun 2015. Dalam usaha mengembangkan Android, pada tahun 2017 dibentuklah *Open Handset Alliance* (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu *Texas Instruments*, *Broadcom Corporation*, Google, HTC, Intel, HGS, *Marvell Technology Group*, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-Mobile dengan tujuan untuk mengembangkan standard terbuka untuk perangkat *mobile*. Pada tanggal 19 Desember 2008, diumumkan bahwa 14 orang anggota baru akan bergabung dengan proyek Android, termasuk *Packet Video*, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.

#### II.2.5. Android Studio

Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan Google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java Populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda

dengan Eclipse yang menggunakan Ant, Android Studio menggunakan *Gradle* sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut :

1. Menggunakan *Gradle-based build system* yang fleksibel.
2. Bisa mem-*build multiple* APK.
3. *Template support* untuk Google Services dan berbagai macam tipe perangkat.
4. *Layout editor* yang lebih bagus, *Built-in support* untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
5. *Import library* langsung dari Maven Repository. ( A. Astuty Indriani et al., 2019 ).

#### II.2.6. *Flutter*

Flutter adalah sebuah SDK atau *framework open source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi Android dan iOS. *Flutter* menggunakan bahasa pemrograman *Dart* dalam pengkodean. Perbedaan *framework Flutter* dengan yang lainnya yaitu dalam *build* aplikasi, pada *framework* ini semua kodenya di *compile* dalam kode native-nya (Android NDK, LLVM, AOT-compiled) tanpa ada interpreter pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat. (Filian Enggar Krisnada.,2019)

### II.2.7. *Firebase*

*Firebase* merupakan *platform* untuk aplikasi *realtime*. Ketika data berubah, maka aplikasi yang terhubung dengan *Firebase* akan meng-*update* secara langsung melalui setiap *device* (perangkat) baik *website* ataupun *mobile*. *Firebase* mempunyai *library* (pustaka) yang lengkap untuk sebagian besar *platform web* dan *mobile* dan dapat digabungkan dengan berbagai *framework* lain seperti *flutter*, *java*, *javascript*, dan lain - lain. *Application Programming Interface* (API) untuk menyimpan dan sinkronisasi data akan disimpan sebagai bit dalam bentuk JSON (*Java Script Object Notation*) pada *Cloud* dan akan disinkronisasi secara *realtime*. (Edwin Adrin Wihelmus Sanadi, Andani Achmad, Dewian, 2018).

### II.2.8. *Firebase Cloud Firestore*

*Cloud Firestore* merupakan *database* terbaru dari *Firebase* untuk pengembangan aplikasi seluler. *Database* ini melanjutkan keberhasilan *Realtime Database* dengan model data baru yang lebih intuitif. *Cloud Firestore* juga memiliki fitur kueri yang lebih lengkap dan lebih cepat, serta penskalaan yang lebih mendalam dibandingkan dengan *Realtime Database*. Seperti *Firebase Realtime Database*, *Cloud Firestore* membuat data tetap terhubung di aplikasi klien melalui *listener realtime* dan menawarkan dukungan secara *offline* untuk seluler dan *web*. (Ilham Firman Maulana, 2020)

### II.2.9. Unified Modelling Language (UML)

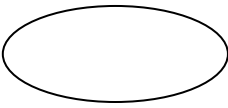
*Unified Modelling Language* (UML) yaitu sebuah standard untuk merancang dan mendokumentasikan sistem perangkat lunak terkhusus pada sistem yang berorientasi objek. Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya (Genta Primadana Edde et al., 2021)

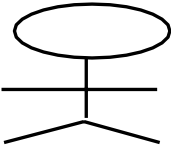
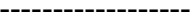
#### II.2.9.1. Diagram UML

##### 1. Use Case Diagram

Merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case diagram* terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem. (T. Bayu Kurniawan, 2020). Adapun Simbol *Use Case* yaitu :

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

| No | Simbol                                                                                                 | Deskripsi                                                                                         |
|----|--------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1. | <i>Use Case</i><br> | Fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |

|    |                                                                                                                    |                                                                                                                                                         |
|----|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2. | Aktor / <i>actor</i><br>          | Adalah orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. |
| 3. | Asosiasi / <i>association</i><br> | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .                                                                  |
| 4. | <<Include>><br>                   | Merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> lain, contohnya adalah pemanggilan sebuah program.         |
| 5. | <<Extends>><br>                 | Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat.                                                                                 |

**Sumber : Antonius Wahyu Sudrajat (2020)**

## 2. Class Diagram

*Class diagram* digunakan untuk melakukan visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe *diagram* yang paling banyak 36 digunakan. *Class diagram* juga dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap kelas di dalam model desain (*logical view*) dari suatu sistem. Selama proses desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. (T. Bayu Kurniawan, 2020). Adapun Simbol *Class Diagram* yaitu :

**Tabel II.2. Multiplicity Class Diagram**

| No | Multiplicity | Penjelasan                                                     |
|----|--------------|----------------------------------------------------------------|
| 1  | <b>1</b>     | Satu dan hanya satu                                            |
| 2  | <b>0..*</b>  | Boleh tidak ada atau 1 atau lebih                              |
| 3  | <b>1..*</b>  | 1 atau lebih                                                   |
| 4  | <b>0..1</b>  | Boleh tidak ada, maksimal 1                                    |
| 5  | <b>n..n</b>  | Batasan antar, contoh 2..4 mempunyai arti minimal 2 maksimal 4 |

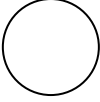
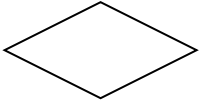
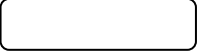
**Sumber : Antonius Wahyu Sudrajat (2020)**

### 3. Activity Diagram

*Activity diagram* menggambarkan berbagai aliran aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. (T. Bayu Kurniawan, 2020). Adapun Simbol *Activity Diagram* yaitu :

**Tabel II.3. Activity Diagram**

| No | Simbol | Deskripsi |
|----|--------|-----------|
|----|--------|-----------|

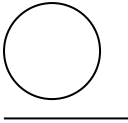
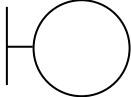
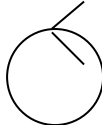
|    |                                                                                                                       |                                                                                           |
|----|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
|    | Status awal<br>                      | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.       |
| 2. | Aktivitas<br>                        | Aktivitas dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.                 |
| 3. | Percabangan/<br><i>decision</i><br> | <i>Asosiasi</i> percabangan dimana jika ada pilihan aktivitas lebih dari satu.            |
| 4. | Penggabungan<br>                   | <i>Asosiasi</i> , penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |

**Sumber : Antonius Wahyu Sudrajat (2020)**

#### 4. Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). (T. Bayu Kurniawan, 2020). Adapun Simbol *Sequence Diagram* yaitu :

**Tabel II.4. Sequence Diagram**

| No | Simbol                                                                                                      | Deskripsi                                                                                                                      |
|----|-------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 1. | <i>Entity Class</i><br>    | Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang menggambarkan hubungan yang akan dilakukan |
| 2. | <i>Boundary Class</i><br>  | Berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem.                |
| 3. | <i>Control Class</i><br> | Suatu objek digunakan untuk menghubungkan <i>Boundary</i> dengan tabel.                                                        |
| 4. | <i>Message</i><br>       | Simbol mengirim pesan antar <i>class</i> .                                                                                     |

**Sumber : Antonius Wahyu Sudrajat (2020)**