

BAB II TINJUAN PUSTAKA

2.1 Pengertian Aplikasi

Peranan aplikasi yang sering dijumpai diberbagai bidang kehidupan, seperti pendidikan, bisnis, dan hiburan. Aplikasi yang dibangun pada penelitian ini merupakan aplikasi berbasis Android yang artinya aplikasi yang dibangun adalah aplikasi *mobile*. Menurut kutipan yang diambil pada penelitian yang dilakukan oleh Muslihudin dan Helmiyanto (2020), aplikasi *mobile* adalah sebuah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengkapan seperti *Personal Digital Assistants (PDA)*, telepon seluler atau *handphone*. Dengan menggunakan aplikasi *mobile* maka dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan kantor, *browsing* dan lain sebagainya. Secara keseluruhan aplikasi *mobile* merupakan aplikasi yang tetap dapat digunakan walaupun pengguna berpindah, dari satu tempat ke tempat yang lain tanpa terjadi pemutusan atau terputusnya komunikasi.

2.2 Pengertian Konseling

Menurut Lisna Zahrotun (2015), konseling adalah salah satu komponen yang penting dalam proses pendidikan sebagai suatu sistem. Konseling merupakan bantuan kepada individu dalam menghadapi persoalan-persoalan yang dapat timbul dalam hidupnya. Sedangkan konsep konseling pengertian konseling sebagai upaya membantu individu melalui proses interaksi yang bersifat pribadi antara konselor dan konseling agar konseli mampu memahami diri dan lingkungannya, mampu membuat keputusan dan menentukan tujuan berdasarkan nilai yang diyakininya, sehingga konseli merasa bahagia dan efektif perilakunya. Bimbingan membantu individu mencapai perkembangan diri secara optimal sebagai makhluk sosial.

2.3 Mental Illness

Menurut APA (*American Psychiatric Association*, 2018), Mental illness adalah kondisi kesehatan yang melibatkan perubahan emosi, pemikiran, atau perilaku (atau kombinasi dari ini). Mental illness dikaitkan dengan kesulitan dan/atau masalah yang mengganggu keberfungsian dalam kegiatan sosial, pekerjaan, atau aktivitas keluarga. Artinya, orang yang memiliki mental illness secara signifikan memengaruhi cara mereka merasa, berpikir, berperilaku, dan berinteraksi dengan orang lain. Mental illness bukanlah hal yang memalukan melainkan sebuah kondisi medis, sama seperti penyakit jantung atau diabetes.

Mental illness dapat disembuhkan dan para ahli di bidangnya terus memperluas pemahaman serta mengembangkan perawatan untuk membantu orang dengan kondisi kesehatan mental supaya dapat kembali berfungsi dalam kehidupan sehari-hari. Mental illness dapat berbentuk ringan yang hanya mengganggu keberfungsian hidup sehari-hari, namun ada pula kondisi mental illness yang sangat parah sehingga memerlukan perawatan di rumah sakit. Beberapa bentuk mental illness adalah *skizofrenia*, gangguan kecemasan, gangguan *mood*, dan gangguan kepribadian.

Gangguan kesehatan mental dibagi menjadi beberapa klasifikasi oleh para dokter dan ahli supaya menjadi sarana yang dapat dimengerti dalam menggambarkan kondisi gangguan tertentu. Adanya sistem klasifikasi gangguan ini dapat dijadikan pegangan diagnostik untuk para ahli supaya terdapat pemahaman yang sama mengenai pengertian, gejala dan dapat dipilih tindakan mana yang kiranya dapat efektif untuk menanganinya.

Klasifikasi yang paling populer digunakan adalah klasifikasi gangguan yang dikemukakan oleh APA (*American Psychiatric Association*) yang disebut DSM (*Diagnostic and Statistical Manual of Mental Disorder*). DSM telah mengalami revisi sejak tahun 1979 dan saat ini telah mencapai DSM-V. Selain itu, WHO juga membuat klasifikasi gangguan kejiwaan yang disebut ICD (*International Classification of Diseases of the World Health Organization*) yang saat ini telah mencapai ICD-10. Kedua sistem klasifikasi ini dioperasionalkan dengan munculnya kriteria dalam bentuk gejala-gejala tetap. Di Indonesia, sistem

klasifikasi gangguan yang digunakan oleh para ahli dan tentunya dibuat sesuai dengan gangguan khas yang ada di Indonesia disebut Pedoman Penggolongan Diagnostik Gangguan Jiwa (PPDGJ) dan saat ini telah mencapai PPDGJ-III.

Mental illness memiliki kriteria standar yang perlu dipenuhi apabila ingin menegakkan diagnosis. Dalam mendiagnosis, apa saja gejala yang dialami, berapa lama gejala tersebut dialami dan dampak apa yang muncul pada individu akibat dari gejala tersebut menjadi pertimbangannya. Perlu diketahui bahwa diagnosis untuk gangguan kesehatan mental tidak dapat dilakukan oleh sembarang orang melainkan hanya dapat dilakukan oleh profesional seperti psikiater dan psikolog.

2.3.1 Gangguan Depresi

Depresi adalah jenis gangguan mental yang paling sering terjadi. Jenis gangguan mental ini ditandai dengan kesedihan yang terlalu lama hingga penderitanya bisa merasa putus asa, bersalah, tidak berharga, tidak termotivasi, hingga berbagai keluhan fisik yang tidak jelas penyebabnya.

2.3.2 Gangguan Bipolar

Gangguan bipolar adalah penyakit mental yang ditandai dengan perubahan suasana hati yang tidak biasa. Perubahan ini bisa terjadi dari sangat bahagia menjadi sangat sedih dan putus asa.

2.3.3 Gangguan PTSD

Gangguan stres pascatrauma (PTSD) adalah gangguan kesehatan mental yang terjadi setelah seseorang mengalami atau melihat peristiwa traumatis. Peristiwa ini bisa berupa kejadian yang mengancam jiwa, seperti bencana alam, kecelakaan mobil, atau kekerasan seksual, atau pengalaman lain yang membuatnya trauma.

2.3.4 Gangguan Kecemasan

Gangguan kecemasan adalah perasan cemas yang sangat kuat, berlebihan, serta berlangsung lama dan bisa memburuk seiring waktu, hingga terasa sangat membebani. Jenis gangguan ini meliputi serangan panik, gangguan obsesif kompulsif (OCD), dan fobia.

2.3.5 Gangguan Skizofrenia

Skizofrenia adalah jenis mental illness yang terjadi saat penderitanya tidak mampu membedakan kenyataan dan pikirannya sendiri. Gangguan ini bisa menyebabkan penderitanya mengalami pemikiran yang tidak realistis, halusinasi, dan perubahan perilaku.

2.4 Pengertian Android

Penelitian ini membangun aplikasi berbasis Android. Menurut *Google Developer Training Team* (2016), Android adalah sistem operasi dan *platform* pemrograman yang dikembangkan oleh *Google* untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan *kit development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android.

Menurut kutipan yang diambil dari penelitian Suryadi, Sutedi, dan Melda (2019), Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. Android pada awalnya dikembangkan oleh Android, Inc., dengan dukungan keuangan dari Google, yang kemudian membelinya pada tahun 2005. Antarmuka pengguna Android umumnya merupakan manipulasi langsung, menggunakan gerakan sentuh yang mirip dengan tindakan nyata, seperti menggesek, mengetuk, dan mencubit untuk memanipulasi objek di layar, seperti serta keyboard virtual untuk menulis teks. Android adalah sistem operasi sumber terbuka, dan Google merilis kodenya di bawah lisensi Apache. Kode dengan open source dan lisensi lisensi di Android memungkinkan perangkat lunak untuk menjadi dimodifikasi dan didistribusikan secara bebas oleh pembuat perangkat, operator nirkabel, dan aplikasi pengembang. Selain itu, Android memiliki sejumlah besar Komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi yang disesuaikan dari bahasa pemrograman Java.

Menurut Karman, Mulyono, dan Taqwa (2019), Android adalah sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan kit *development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Android Juga menyediakan pasar untuk mendistribusikan aplikasi. secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler. Versi Android yang kompatibel dan digunakan dalam penelitian ini adalah dimulai dari versi 8.0 sampai versi Android saat ini.

Menurut Yudhanto dan Wijayanto (2017), Perangkat Android memiliki banyak fitur perangkat keras di dalamnya. Inilah yang dapat dimanfaatkan *developer* dalam membangun aplikasi, diantaranya adalah :

- a. *Touchscreen*. Perangkat Android memiliki fitur layar sentuh yang memberikan beberapa kemungkinan bagi pengguna untuk berinteraksi dengan aplikasi menggunakan jari.
- b. *GPS*. Sistem operasi Android mendukung *GPS* yang memungkinkan *developer* untuk mengakses lokasi pengguna.
- c. *Accelerometer*. Android mendukung *accelerometer*, yaitu perangkat yang digunakan untuk mengatur percepatan.
- d. *SD Card*. Android memiliki fitur yang memungkinkan pengguna atau aplikasi untuk mengakses (menyimpan atau membuka) *file* pada *SD Card*.

Android memiliki banyak fitur perangkat lunak yang dapat digunakan oleh *developer* dalam mengembangkan aplikasi, yakni :

- a. *Internet*. Kemampuan akses internet pada Android memiliki banyak keunggulan. Berbagai informasi secara real-time dapat diperoleh dengan mudah dengan internet.
- b. *Audio dan Video Support*. Sistem operasi Android memungkinkan *developer* menyertakan audio dan video dalam aplikasi dengan mudah.

- c. *Contact*. Android memungkinkan akses ke kontak yang tersimpan pada perangkat.
- d. *Security*. Android memungkinkan aplikasi untuk melakukan banyak hal. Android juga menyiapkan keamanan berupa permission berkaitan dengan beberapa tugas.
- e. *Google APIs*. Sistem operasi Android memungkinkan dengan tidak terbatas membuat panggilan telepon, mengorganisasi kontak dan menginstal aplikasi. *Developer* juga dapat mengintegrasikan peta ke dalam suatu aplikasi dengan menggunakan *Maps API*.

2.5 DBMS

DBMS yang digunakan dalam pengelolaan data pada aplikasi yang dibangun adalah MySQL. Menurut Suryadi, Sutedi, dan Melda (2019), basis data adalah kumpulan informasi yang disimpan dalam komputer sistematis sehingga dapat diperiksa menggunakan program komputer untuk memperoleh informasi dari basis data itu. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri basis data disebut sistem manajemen basis data (DBMS). Sistem *database* dapat dipelajari dalam informasi *sains*. Istilah "*database*" berasal dari ilmu komputer.

Konsep dasar *database* adalah kumpulan catatan, atau potongan pengetahuan. Sebuah *database* memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan yang dimaksud adalah disebut dengan skema. Skema menggambarkan objek yang diwakili oleh *database* dan hubungan antara objek-objek tersebut. Ada banyak cara untuk mengatur skema, atau untuk memodelkan struktur *database* yang dikenal sebagai model *database* atau model data. Model umumnya digunakan saat ini adalah model relasional, yang dalam istilah awam mewakili semua informasi dalam bentuk tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Dalam model ini, hubungan antar tabel adalah direpresentasikan menggunakan nilai yang sama antar tabel. Model lain seperti hierarki model dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antara tabel. Istilah *database* mengacu pada kumpulan data yang saling berhubungan, dan perangkat lunak harus menyebutnya sebagai

sistem manajemen basis data (DBMS). Komponen-komponen yang menyusun lingkungan DBMS terdiri atas perangkat keras, perangkat lunak, data, prosedur, dan orang. Terdapat beberapa elemen basis data, yaitu *database*, tabel, entitas, atribut, nilai data, *file*, dan *record*.

2.6 Aplikasi Android Studio

Menurut Karman, Mulyono, dan Taqwa (2019), Android Studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android Studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android Studio memiliki fitur :

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat
- d. Mendukung proguard dan app-signing untuk keamanan
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platform untuk setiap aplikasi yang dikembangkan

Menurut Suryadi, Sutedi, dan Melda (2019), Android Studio adalah lingkungan pengembangan terintegrasi atau Pengembangan Terpadu *Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain menjadi editor kode *IntelliJ* dan alat pengembang yang kuat, Android Studio menawarkan lebih banyak fitur untuk meningkatkan produktivitas anda saat membuat aplikasi Android, misalnya:

- a. Sistem versi berbasis *Gradle* yang fleksibel
- b. Emulator yang cepat dan kaya fitur
- c. Lingkungan terpadu untuk pengembangan semua perangkat Android

- d. *Instant Run* untuk mendorong perubahan pada aplikasi yang sedang berjalan tanpa membuat APK baru
- e. *Template* kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor sampel kode
- f. Alat dan kerangka kerja pengujian yang luas
- g. Alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya
- h. Dukungan C++ dan NDK
- i. Dukungan bawaan untuk *Google Cloud Platform*, memudahkan integrasi *Google Cloud*

Messaging dan *App Engine* setiap proyek di Android Studio berisi satu atau beberapa modul dengan *file* kode sumber dan *file* sumber daya. Jenis-jenis modul tersebut antara lain:

- a. Modul aplikasi Android
- b. Modul Perpustakaan
- c. Modul Mesin Aplikasi Google

Secara default, Android Studio akan menampilkan *file* proyek anda dalam tampilan proyek Android. Tampilan diatur oleh modul untuk menyediakan akses cepat ke *file* sumber utama proyek anda. Semua versi *file* muncul di bagian atas di bawah *Gradle Scripts* dan setiap modul aplikasi berisi folder berikut:

- a. *manifest*: berisi *file AndroidManifest.xml*.
- b. *java*: berisi *file* kode sumber *Java*, termasuk kode pengujian *JUnit*.
- c. *res*: berisi semua sumber daya non-kode, seperti *layout XML*, *string UI*, dan gambar *bitmap*.

2.7 Bahasa Pemrograman dan Database yang Digunakan

Aplikasi yang dibangun berbasis Android. Aplikasi dibuat dengan menggunakan aplikasi *Android Studio*. Bahasa pemrograman yang digunakan dalam pembuatan aplikasi ini adalah *Java*. Sedangkan *database* yang digunakan dalam penyimpanan data adalah *MySQL*.

2.7.1 Bahasa Pemograman *Java*

Bahasa pemograman yang digunakan dalam membuat aplikasi berbasis Android adalah Java. Menurut Yuniansyah (2020), bahasa pemograman adalah teknik komunikasi standar untuk mengekspresikan instruksi kepada komputer. Layaknya bahasa manusia, setiap bahasa memiliki tata tulis dan aturan tertentu. Bahasa pemograman memfasilitasi seorang programmer untuk secara spesifik membuat perintah apa yang akan dilakukan oleh komputer selanjutnya, bagaimana data tersebut disimpan dan dikirim, dan apa yang akan dilakukan apabila terjadi kondisi yang variatif.

Bahasa pemograman Java merupakan bahasa pemograman berorientasi objek dan sampai saat ini terus berkembang. Banyak paket standar yang kemudian dikembangkan untuk semua kebutuhan, seperti Java Lang untuk elemen dasar, Java.net untuk kelas yang terkoneksi ke jaringan, Java.awt untuk antarmuka pengguna, dan Java.applet untuk aplikasi berbasis web. Karakteristik dari pemograman Java adalah :

- a. Sederhana (simple) bahasa Java menggunakan sintak atau penulisan kode yang sederhana sehingga mudah dimengerti oleh programmer.
- b. Berorientasi objek (object oriented) Java menggunakan pemograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali.
- c. Terdistribusi (distributed) Java dikembangkan untuk dapat membuat aplikasi terdistribusi secara mudah karena terdapat pustaka networking yang telah terintegrasi pada Java.
- d. Interpreted Program Java dijalankan menggunakan interpreter yaitu Java Virtual Machine (JVM). Untuk itu, setelah kompilasi program Java dapat digunakan pada platform yang berbeda-beda.
- e. Multithreaded Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.
- f. Dynamic Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu class dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan class tersebut.

2.7.2 MySQL

Pengelolaan DBMS yang digunakan dalam pembuatan aplikasi sistem informasi penjadwalan pemadaman listrik berbasis Android ini adalah *MySQL*. Menurut Fitri (2020), *MySQL* adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia). *MySQL* adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *free software* (perangkat lunak bebas) dan *shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi, *MySQL* adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada.

Pengelolaan DBMS yang digunakan dalam pembuatan aplikasi ini adalah *MySQL*. Menurut kutipan yang diambil pada penelitian yang dilakukan oleh Muslihudin dan Helmiyanto (2020), *MySQL* (*My Structure Query Language*) adalah suatu perangkat lunak database relasi atau *Relational Database Management System* (RDBMS) yang didistribusikan gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang dijadikan *closed source* atau komersial.

Seperti yang sudah disebutkan sebelumnya, *MySQL* masuk ke dalam jenis RDBMS (*Relational Database Management Sistem*). Maka dari itu, istilah semacam baris, kolom, tabel dipakai pada *MySQL*. Menurut Enterprise (2018), *MySQL* merupakan *server* yang melayani *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL. *MySQL* merupakan *database engine* atau *server database* yang mendukung bahasa SQL sebagai bahasa interaktif dalam mengelola data. *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*.

2.8 Pemodelan Sistem UML (*Unified Modelling Language*)

Pemodelan sistem yang digunakan untuk perancangan adalah pemodelan UML (*Unified Modelling Language*). UML (*Unified Modelling Language*) adalah suatu

metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software*. Jenis pemodelan UML yang digunakan dalam pengembangan sistem pada penelitian ini adalah *use case diagram*, *activity diagram*, dan *class diagram*.

2.8.1 Use Case Diagram

Menurut Rosa dan Shalahuddin (2018), *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang dibuat aktor dan *use case*.

Tabel 2.1 Simbol *Use Case Diagram*

Keterangan	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.

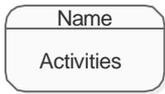
Tabel 2.1 Lanjutan

Keterangan	Simbol	Deskripsi
Asosiasi	—————	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi	«extends» —————▷	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.
Generalisasi	—————→	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan <i>/include/uses</i>	«<<include>>» -----▷ «uses» —————▷	Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : a. Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan b. Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

2.8.2 Activity Diagram

Menurut Rosa dan Shalahuddin (2018), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang terdapat pada *activity diagram* adalah seperti pada Tabel 2.2.

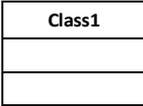
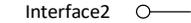
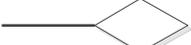
Tabel 2.2 Simbol Diagram Aktivitas

Simbol	Deskripsi
<p>Status awal</p> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
<p>Aktivitas</p> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
<p>Percabangan</p> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
<p>Penggabungan</p> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<p>Swimlane</p> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
<p>Status akhir</p> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.8.3 Class Diagram

Menurut Rosa (2018), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Simbol-simbol yang ada pada diagram kelas adalah seperti pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Natarmuka/<i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
<p>Asosiasi</p> 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasi</p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.9 Metode Pengembangan Sistem *Scrum*

Menurut Mei Prabowo (2020), scrum adalah suatu metodologi atau kerangka kerja yang terstruktur untuk mendukung pengembangan produk yang kompleks. *Scrum* terdiri dari sebuah tim yang memiliki peran dan tugas masing-masing. Setiap komponen dalam kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan *scrum*.

Proses pengembangan menggunakan metode scrum terdapat empat tahapan pengembangan yaitu : (1) *product backlog*, (2) *sprint planning meeting*, (3) *sprint*

backlog, (4) *sprint*, dan (5) *finished work*. Adapun kerangka kerja scrum terlihat pada Gambar 2.1



Gambar 2.1 Kerangka Kerja Scrum

Adapun penjelasan dari tiap tahapan pada kerangka kerja *scrum* menurut Ikhwansyah dan Ramadhan (2019) adalah sebagai berikut :

a. *Product Backlog*

Pada tahapan ini pekerjaan-pekerjaan yang akan dilakukan ditulis dan dijelaskan oleh *Product Owner* pada *Product Backlog*. Pekerjaan yang akan dilakukan memiliki prioritas dalam pengerjaan. Kebutuhan *customer* dan kebutuhan secara teknikal sebagai acuan untuk menentukan prioritas pengerjaan. Selain itu prioritas juga dapat ditentukan berdasarkan hal yang dinilai krusial atau penting dengan sistem yang akan dibangun. Untuk menentukan prioritas *Product Owner* dapat meminta pendapat dari tim *scrum* atau *stakeholders*.

b. *Sprint Planning Meeting*

Sprint planning merupakan rapat yang membahas tentang pekerjaan-pekerjaan yang akan dilakukan selama *sprint*. Seberapa lama dan sampai mana produk yang dikembangkan selama *sprint* itu berlangsung dibahas di dalam *sprint planning*. Pada *meeting* ini, *scrum master* bertugas untuk memastikan seluruh anggota tim hadir, dan memahami tujuannya. Tujuan diadakannya *sprint planning*, untuk bisa menjawab pertanyaan-pertanyaan berikut:

1. Apa *goal* dari *sprint*?
2. Apa yang dapat disajikan dari hasil *sprint* yang sudah berjalan?
3. Apa saja yang harus dilakukan untuk mencapai *goal* dari *sprint*?

c. *Sprint Backlog*

Pekerjaan-pekerjaan dari *product backlog* didiskusikan dan disepakati oleh tim untuk dimasukkan pada *sprint*. Proses ini merupakan hasil perpindahan dari prioritas tertinggi dari *product owner* yang kemudian akan dikembangkan melalui *sprint*. Proses ini juga dapat disebut sebagai inputan awal sebelum melakukan proses *sprint* pada pengembangan produk.

d. *Sprint*

Sprint memiliki durasi waktu yang konsisten sepanjang masa pengembangan produk. Setelah sprint selesai maka langsung mengerjakan *sprint* berikutnya. *Sprint* merupakan proses dimana dilakukannya pemaparan hasil produk dalam bentuk prototipe kepada pihak terkait. Selain memaparkan terkait dengan produk prototipe juga dilakukan memaparkan berkaitan dengan hal teknis terkait peralatan yang dibutuhkan untuk menjalankan *test engine*.

e. *Finished Work*

Kegiatan pada tahap ini adalah komunikasi antara tim *scrum* dengan pihak klien. Percakapan difokuskan pada peninjauan fitur yang baru saja selesai dalam konteks upaya pengembangan secara keseluruhan. Pada tahap ini pihak klien dapat mengajukan perbaikan jika dimungkinkan. Komunikasi yang berlangsung disertai dengan demo produk atau sistem yang sudah dibuat. Tahap demo aplikasi yang dimaksud yaitu tim menyerahkan perangkat lunak atau produk *increment* kepada klien dengan cara melakukan demonstrasi dan klien melakukan evaluasi terhadap produk.

2.10 Metode *Black Box Testing*

Metode pengujian yang dipakai sebagai uji coba aplikasi yang dibangun adalah *black box testing*. Menurut Rosa dan Shalahuddin (2018), *black box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan pengeluaran 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 dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus *login* maka kasus uji yang dibuat adalah:

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

2.11 Penelitian Terkait

Adapun penelitian terdahulu yang mendukung dan terkait dengan penelitian yang dilakukan dalam pembuatan aplikasi saat ini adalah :

- a. Menurut penelitian yang dilakukan oleh Adrinura dan Hidayat (2021) dengan judul “Perancangan Aplikasi *Mobile Apps* Konsultasi *Online* Pusat Inovasi Psikologi UNPAD” disimpulkan bahwa aplikasi yang dibuat bisa menjadi salah satu solusi untuk mengurangi tingkat depresi yang ada di Indonesia, memberikan *awareness* kepada masyarakat terhadap pentingnya kesehatan mental karena dalam aplikasi ini juga terdapat fitur untuk mencari informasi terhadap kesehatan mental dan sekaligus mempromosikan Pusat Inovasi Psikologi itu sendiri.
- b. Menurut penelitian yang dilakukan oleh Andriyani dan Jati (2017) dengan judul “Model Layanan *Screening* Dalam Konseling Kesehatan Mental Berbasis Android (*E-Couns*) Untuk Mendukung Gerakan Revolusi Mental” disimpulkan model layanan *screening* berbasis Android dalam konseling kesehatan mental untuk mendukung gerakan revolusi mental yang berguna untuk mendapatkan balasan langsung mengenai gangguan kesehatan mental yang kita alami, dan dapat mendapatkan proses *referral* atau pengalih tanganan dengan konselor kesehatan mental yang profesional. Hal tersebut dilakukan untuk menanggulangi gangguan kesehatan mental sudah banyak dilakukan.

