

## **BAB II TINJAUAN 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 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 dapat digunakan walaupun pengguna berpindah dari satu tempat ke tempat yang lain tanpa terjadi pemutusan atau terputusnya komunikasi.

### **2.2 Poliklinik Apotek Rosa**

Apotek Rosa merupakan salah satu apotek terbesar dan terlengkap yang terletak di Bandar Lampung. Apotek ini melayani farmasi, rekam medik, serta layanan poloklinik kesehatan. Terdapat 6 poliklinik praktek dokter pada Apotek Rosa, yaitu poli seperti poli penyakit dalam, poli gigi, poli kebidanan dan kandungan, poli anak, dan poli THT. Adapun jadwal dokter praktek tiap poli adalah seperti pada Tabel 2.1.

Tabel 2.1 Jadwal Praktek Dokter Apotek Rosa

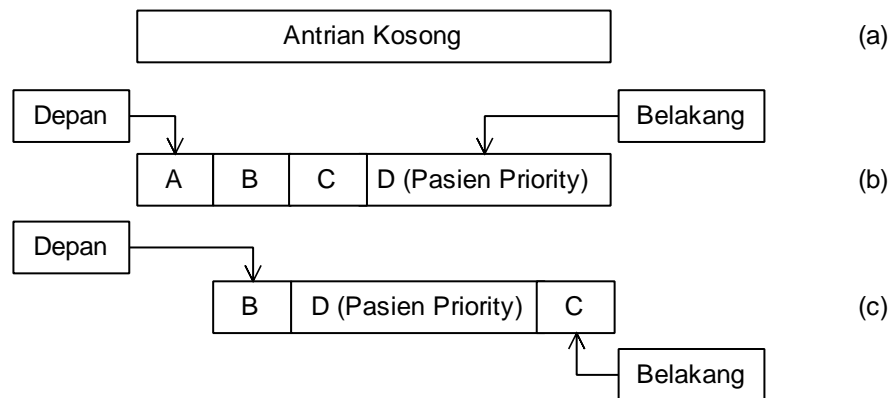
<b>Nama Poli</b>	<b>Nama Dokter</b>	<b>Hari</b>	<b>Jam</b>
Penyakit Dalam	Ali Imron	Senin dan Selasa	17:00 – 20:00
	Ahmad Yusron	Rabu, Kamis, Jumat	17:00 – 20:00

Tabel 2.1 Lanjutan

<b>Nama Poli</b>	<b>Nama Dokter</b>	<b>Hari</b>	<b>Jam</b>
Kebidanan dan Kandungan	Idris Hs	Senin, Rabu, Kamis, Jumat	17:00 – 20:00
		Selasa	13:00 – 15:00
		Sabtu	09:00 – 12:00
Anak	Eli Tua P	Senin - Sabtu	18:30 – 21:00
Gigi	Astrid T Hidayat	Senin - Sabtu	17:00 – 20:00
THT	Rina Hayati	Senin - Sabtu	17:00 – 20:00

### 2.3 Metode *Queue Priority Service*

Menurut Sindar (2019), antrian atau *queue* adalah salah satu struktur data yang memiliki sistem kerja pertama masuk dan pertama keluar seperti halnya antrian pada dunia nyata. *Queue* pada penambahan datanya hanya dapat dilakukan pada salah satu ujung dan penghapusan data hanya dapat dilakukan pada ujung yang lain. Elemen baru dumasukan pada bagian belakang dalam antrian dan penghapusan elemen hanya dilakukan pada bagian ujung depan elemen. *Priority Service* (PS) merupakan model antrian yang mengutamakan layanan kepada antrian yang memiliki prioritas lebih tinggi dibandingkan dengan prioritas yang lebih rendah. Meskipun datang paling akhir, jika prioritasnya paling tinggi maka akan didahulukan. Di dalam antrian dikenal istilah depan dan belakang. Contoh penggambaran depan belakang pemodelan *priority service* adalah ditunjukkan pada Gambar 2.1.



Gambar 2.1 Penjabaran Alur Kerja Antrian *Priority Service* (PS)

Gambar 2.1 (a) menyatakan keadaan ketika antrian dalam keadaan kosong. Gambar 2.1 (b) memperlihatkan ketika terdapat empat item yang sedang antri. Depan menunjuk pada A yang menyatakan data yang akan dilayani duluan dan belakang menunjuk ke D (merupakan prioritas) yang menyatakan item yang terakhir kali masuk antrian. Apabila kemudian A dilayani, maka A tidak lagi berada pada antrian. Hal ini digambarkan pada Gambar 2.1 (c). Dalam hal ini, depan sekarang menunjuk ke D yang merupakan pasien prioritas dan C yang terakhir dilayani.

## 2.4 Android

Penelitian ini membangun aplikasi berbasis Android. Pembangunan aplikasi android ini dibuat dengan menggunakan aplikasi Android Studio. Versi Android minimal yang digunakan dalam pembuatan aplikasi ini adalah versi Android 8.0 Oreo. 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 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 *Kotlin*. Sedangkan *database* yang digunakan dalam penyimpanan data adalah *MySQL*.

### **2.6.1 Bahasa Pemrograman *Kotlin***

Bahasa pemrograman yang digunakan dalam pembuatan aplikasi berbasis Android ini adalah *Kotlin*. Menurut artikel yang tertera pada *website* resmi Android yaitu

developer.android.com, Kotlin adalah bahasa pemrograman modern melalui pengetikan statis yang digunakan lebih dari 60% developer Android profesional untuk membantu meningkatkan produktivitas, kepuasan developer, dan keamanan kode. Sejak diluncurkan pada pagelaran Google I/O pada bulan Mei 2017, Kotlin menjadi bahasa kelas utama (*first class language*) dalam pembuatan aplikasi Android.

Menurut Azriel., dkk (2020), Kotlin adalah sebuah bahasa pemrograman dengan Statically typed (tipe statis) yang berjalan pada *platform Java Virtual Machine* (JVM). Kotlin menggunakan *compiler LLVM* yang artinya, dapat dikompilasi ke dalam kode JavaScript. Pengembang utamanya berasal dari tim programmer JetBrains yang bermarkas di Rusia. Bahasa pemrograman yang satu ini banyak diminati oleh para developer. Kotlin merupakan bahasa yang *powerfull* tentu cocok bagi developer dalam membuat aplikasi Android. Karena Kotlin adalah bahasa nomor satu untuk pengembangan aplikasi Android. Menurut Oktaviyan (2017), kelebihan Kotlin adalah :

- a. *Concise* : Kotlin mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa Java.
- b. *Safe* : Kotlin mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya **NullPointerException** ketika kita *coding* menggunakan bahasa Java.
- c. *Versatile* : Kotlin sejatinya sama seperti Java, karena memang kotlin itu sendiri di turunkan dari bahasa induknya, yaitu Java. Sehingga kotlin juga dapat di pakai dalam pengembangan aplikasi di *Web* maupun *Mobile*.
- d. *Interoperable* : Kotlin tidak sama seperti bahasa Java turunan lainnya (misal; *Scala* ataupun *Clojure*) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan Java. Kotlin mampu membaca kode lama atau library yang kita gunakan atau kita tulis dengan bahasa Java dan begitupun sebaliknya.

### 2.6.2 MySQL

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.

Menurut Solichin (2016), *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL (Database Management System)* atau *DBMS* yang *multithread, multi-user*. Beberapa kelebihan *MySQL* antara lain: *free* (bebas di *download*), stabil dan tangguh, fleksibel dengan berbagai pemrograman, *security* yang baik, dukungan dari banyak komunitas, dan kemudahan.

### 2.7 Aplikasi Android Studio

Android studio merupakan suatu aplikasi yang digunakan dalam pembuatan (*coding*) Android. 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

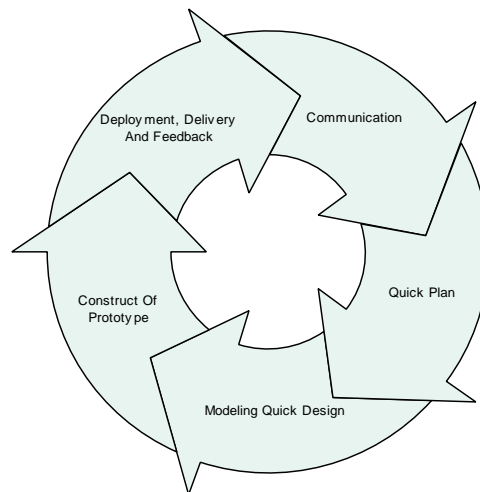
Secara default, Android Studio akan menampilkan *file* proyek anda dalam tampilan proyek Android. Tampilan diatur oleh modul untuk menyediakan akses cepat ke *file* sumber utama proyek anda. Semua versi *file* muncul di bagian atas di bawah *Gradle Scripts* dan setiap modul aplikasi berisi folder berikut:

- 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*.

## 2.8 Metode Pengembangan Sistem Prototipe

Pengembangan perangkat lunak dalam aplikasi ini menggunakan metode prototipe. Menurut kutipan pada penelitian Puspita (2019), *prototype* merupakan metode yang efektif dalam merancang perangkat lunak. Prototype dimulai dengan mengumpulkan kebutuhanyang akan di rancang. Pengembang mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala aktifitas yang diketahui dan kemudian melakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan *input* dan format *output*).

Menurut kutipan pada penelitian Gunawan dan Puspita (2017), *prototype model* adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode prototype, pengembangan dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Prototipe memiliki lima tahapan siklus. Siklus atau ilustrasi dari metode prototipe dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Model Prototipe (Sumber : Pressman, 2012)

## 2.9 Pemodelan Sistem yang Digunakan



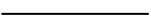
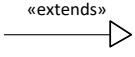
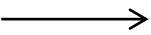
Sistem dibangun dirancang menggunakan pemodelan UML (*Unified Modelling Language*), diantaranya adalah *use case diagram*, *activity diagram*, dan *class diagram*.

### 2.9.1 Use Case Diagram

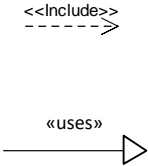
Pemodelan perancangan sistem yang digunakan dalam pembuatan aplikasi ini adalah *use case diagram*. Menurut Rosa dan Shalahuddin (2018), *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui 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*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem

informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.2 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.
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		Relasi use case 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.


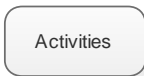

Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)

Keterangan	Simbol	Deskripsi
Menggunakan <i>/include/uses</i>		<p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ol style="list-style-type: none"> <li>Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan</li> <li>Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan.</li> </ol>


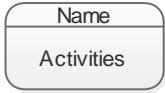

### 2.9.2 Activity Diagram

Menurut Rosa (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 2.3.

Tabel 2.3 Simbol Diagram Aktivitas

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

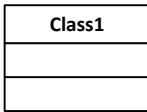
Tabel 2.3 Lanjutan

Simbol	Deskripsi
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

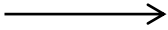
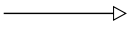
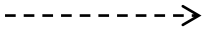

### 2.9.3 Class Diagram

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 2.4.

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Natarmuka/ <i>interface</i> Interface2 ○—	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi —————	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

Tabel 2.4 Lanjutan

Simbol	Deskripsi
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

### 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 Hadi dan Nugrahaeni (2016) dalam penelitiannya yang berjudul “Aplikasi Penjadwalan Terapi dengan Metode FCFS pada Sixo Reflexology” menyimpulkan bahwa aplikasi penjadwalan terapi dapat membantu pihak Sixo Reflexology dalam menjadwalkan terapi bagi pasiennya.
- b. Menurut Fratiwi dan Mariana (2020) dalam penelitiannya yang berjudul “Metode FCFS Dalam Menunjang Sistem Layanan Antrian Pembagian Dana Pensiun Studi Kasus Kantor Pos Bongsari” menyimpulkan bahwa dengan sistem ini, para pensiun bisa mendaftar dimana saja sebelum mendatangi kantor yang dituju sehingga tidak terjadi nya kesalahan menginput data dan juga untuk mendisiplinkan antrian agar tidak terjadi antrian yang panjang dan membuat kericuhan pada kantor tujuan