

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Aplikasi

Penelitian ini membangun aplikasi layanan *home care* yang diakses melalui *smartphone* Android. 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 Pengertian Pelayanan Kesehatan

Menurut Raodah (2019), pelayanan kesehatan adalah setiap upaya yang diselenggarakan sendiri atau secara bersama-sama dalam suatu organisasi untuk memelihara dan meningkatkan kesehatan, mencegah, dan menyembuhkan penyakit serta memulihkan kesehatan perorangan, keluarga, kelompok dan ataupun masyarakat. Kesehatan merupakan salah satu kebutuhan dasar manusia. Dalam hirarki Maslow dijelaskan bahwa kebutuhan dasar manusia menyangkut didalamnya pemenuhan kebutuhan fisik. Perlindungan fisik salah satunya berupa ancaman tubuh atau hidup yang dapat berupa penyakit. Seseorang yang mengalami sakit akan berupaya untuk mencari pelayanan kesehatan agar dirinya dapat kembali pada keadaan sehat. Pelayanan kesehatan diantaranya rumah sakit, puskesmas dan

tempat-tempat praktik kesehatan lainnya. Berbagai macam pelayanan ditawarkan kepada pasien yang ingin meningkatkan kesehatannya.

2.3 Pengertian *Home Care*

Menurut Raodah (2019), *home care* atau perawatan kesehatan merupakan suatu bentuk pelayanan kesehatan kepada masyarakat yang diterapkan di beberapa kota-kota besar di Indonesia. Pelayanan ini bertujuan untuk meningkatkan, mempertahankan atau memulihkan kesehatan dan memaksimalkan tingkat kemandirian serta meminimalkan komplikasi akibat dari penyakit serta memenuhi kebutuhan dasar pasien dan keluarga di rumah.

Menurut Meilianingsih (2016), *home care* adalah pelayanan kesehatan yang berkesinambungan dan komprehensif diberikan kepada individu, keluarga di tempat tinggal mereka yang bertujuan untuk meningkatkan, mempertahankan, memulihkan kesehatan, atau memaksimalkan kemandirian, dan meminimalkan kecacatan akibat penyakit. Hasil penelitian menunjukkan bahwa pelayanan *home care* dapat meningkatkan kemandirian pada keluarga dalam merawat pasien pasca-stroke. Dalam penerapan proses keperawatan di rumah terjadi proses alih peran dari perawat kepada keluarga secara bertahap dan berkelanjutan untuk mencapai kemandirian keluarga.

2.4 Android

Penelitian ini akan dibangun suatu aplikasi yang 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). 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.

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 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 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*.

2.6 Pemodelan Sistem UML (*Unified Modelling Language*)

Pemodelan sistem yang digunakan menggunakan 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, squence diagram, dan class diagram*.

2.6.1 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 2.1 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>
<p>Aktor</p> 	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
<p>Asosiasi</p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
<p>Ekstensi</p> 	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 itu; mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek.
<p>Generalisasi</p> 	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 2.1 Lanjutan

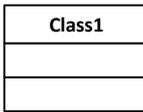
Simbol	Deskripsi
<p><i>Include</i></p>	Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :

	<p>a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan</p> <p>b. <i>Include</i> 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.</p>
---	--

2.6.2 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 2.2.

Tabel 2.2 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 2.2 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>).

2.7 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. Proyek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama “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*.

2.8 Metode *Earliest Due Date* (EDD)

Earliest Due date atau disingkat menjadi EDD adalah istilah bahasa Inggris yang berarti batas waktu yang lebih awal. *Earliest Due Date* (EDD) adalah metode penjadwalan produksi dengan menyelesaikan dahulu tugas yang mempunyai batasan waktu (*due date*) lebih rendah. Menurut Rudyanto (2010), Metode *Earliest Due Date* mengurutkan pekerjaan-pekerjaan berdasarkan tanggal jatuh tempo (*due date*) yang terdekat. Metode ini dapat digunakan, untuk penjadwalan pada satu mesin (*single machine*) maupun untuk penjadwalan pada beberapa mesin (*parallel machine*). Terdapat beberapa langkah dalam melakukan penjadwalan produksi dengan *Earliest Due Date* (EDD) sebagai berikut :

- a. Susun tugas kerja berdasarkan batasan waktu (*due date*) yang ada, nilai *due date* terendah berada paling awal. Begitu seterusnya hingga nilai *due date* paling besar.
- b. Hitung waktu kumulatif dari waktu pemrosesan dan jumlahkan nilainya secara keseluruhan.
- c. Hitung keterlambatan pekerjaan dengan mengurangi *flow time* dengan *job due date*. Kemudian jumlahkan keseluruhan nilai yang ada.
- d. Hitung Waktu penyelesaian keseluruhan dengan rumus sebagai berikut :

$$\text{average completion time} = \frac{\text{sum of total flow time}}{\text{number of job}}$$

- e. Hitung Utilisasi pekerjaan dengan rumus sebagai berikut :

$$\text{utilization} = \frac{\text{total job work time}}{\text{sum of total flow time}}$$

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 Sari, Kurniawan, dan Suhandono (2020), dalam penelitiannya berjudul “Rancang Bangun Aplikasi *Homecare* Ibu dan Anak Berbasis Android” menyimpulkan bahwa aplikasi ini memberikan kemudahan bagi para orang tua dalam memesan jasa *homecare* yang sesuai dengan perawatan / *treatment* yang dibutuhkan, memberikan kemudahan bagi para orang tua dalam hal waktu lebih efektif dan efisien, karena dapat memesan jasa *homecare* ke rumah maupun ke tempat perawatan sesuai perjanjian waktu yang telah ditentukan sebelumnya.
- b. Menurut Sitorus, Dwi, dan Pribadi (2018). Dalam penelitiannya berjudul “Pemanfaatan GIS Untuk Aplikasi Layanan *Homecare*” menyimpulkan bahwa aplikasi ini menggunakan *webview* sehingga dapat di rubah kebentuk apk dan dijalankan di Android. Aplikasi ini juga di lengkapi fitur mapping yang memberikan informasi jarak dan rute dari perawat menuju lokasi pasien guna membantu perawat dalam menemukan lokasi dari pasien. Selain itu dibuatnya aplikasi ini juga untuk mempermudah dalam pengelolaan program *homecare* dan pelaporan kegiatan perawatan.

- c. Menurut Dwi (2019) dalam penelitiannya yang berjudul “Penjadwalan Produksi untuk Meminimasi Keterlambatan Distribusi dengan Metode *Earliest Due Date*” menyimpulkan bahwa Metode EDD sebagai metode perbaikan dapat memperbaiki waktu penyelesaian sehingga keterlambatan pengiriman dapat terminimalisir.
- d. Menurut Yuni (2016) dalam penelitiannya yang berjudul “Membangun Kamus Obstetri Berbasis Android (Studi Kasus : Kampus Kebidanan Adila Bandar Lampung)” menyimpulkan bahwa aplikasi ini dapat menjadi alternatif mahasiswa dan masyarakat dalam menerjemahkan istilah kebidanan tanpa harus membawa buku panduan.
- e. Menurut Sulyono, Fitria, dan Lia (2018) dalam penelitiannya yang berjudul “Rancang Bangun Teknologi Informasi E-Complaint pada Perguruan Tinggi” menyimpulkan bahwa aplikasi *e-complaint* dapat dijadikan alternatif yang baik untuk melakukan proses komplain secara elektronik. Aplikasi ini juga dapat digunakan dimana saja selama terdapat akses internet dan dapat menyesuaikan ukuran *device* dengan kemampuannya yang *responsive*.