

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Verra Sofica, et. al., 2019 melakukan penelitian berjudul “Aplikasi Pendaftaran Pasien Rawat Jalan *Online* pada Klinik Mulya Medika menggunakan *Waterfall*”. Pada penelitian tersebut dihasilkan sebuah aplikasi pendaftaran pasien rawat jalan pada klinik Mulya Medika berbasis *website* yang dapat digunakan oleh pasien yang akan mendaftar secara *online*. Kekurangan dari penelitian tersebut adalah aplikasi yang dihasilkan ditargetkan untuk digunakan pada perangkat *desktop* sehingga tidak efisien untuk digunakan pada perangkat *mobile*. Yang menjadi perbedaan dengan penelitian saat ini adalah pada penelitian ini aplikasi yang dibangun berbasis android sehingga dapat digunakan secara *mobile* serta aplikasi pendaftaran pasien rawat jalan yang dibuat akan digunakan untuk pendaftaran pasien rawat jalan pada RSU Mitra Medika.

Edo Nado Candra, et. al., 2019 melakukan penelitian berjudul “Aplikasi Pendaftaran Pasien Rawat Jalan Rumah Sakit Meilia Cibubur Berbasis *Web*”. Pada penelitian tersebut dihasilkan sebuah aplikasi berbasis *web* untuk digunakan dalam proses pendaftaran pasien rawat jalan pada rumah sakit Meilia. Kekurangan dari penelitian tersebut adalah aplikasi yang dihasilkan ditargetkan untuk digunakan pada perangkat *desktop* sehingga tidak efisien untuk digunakan pada perangkat *mobile*. Yang menjadi perbedaan dengan penelitian saat ini adalah pada penelitian ini aplikasi yang dibangun merupakan berbasis android dan akan

digunakan untuk pendaftaran pasien rawat jalan pada RSU Mitra Medika.

Raden Budiarto Hadiprakoso, et. al., 2021 melakukan penelitian berjudul “Pengembangan Aplikasi Registrasi Rawat Jalan Pasien RSUD Menggunakan Perangkat Android”. Pada penelitian tersebut dihasilkan sebuah aplikasi berbasis android untuk melakukan pendaftaran pasien rawat jalan. Kekurangan dari penelitian tersebut adalah pada aplikasi yang dihasilkan tidak terdapat informasi ketersediaan kuota layanan yang tersedia. Yang menjadi perbedaan dengan penelitian saat ini adalah pada penelitian ini aplikasi yang dibangun merupakan berbasis android dan akan digunakan untuk pendaftaran pasien rawat jalan pada RSU Mitra Medika.

Nicodias Palasara, et. al., 2020 melakukan penelitian berjudul “Implementasi Aplikasi Klinik Pelayanan Rawat Inap dan Rawat Jalan”. Pada penelitian tersebut dihasilkan sebuah aplikasi berbasis *web* untuk digunakan dalam proses pendaftaran pasien rawat inap dan rawat jalan. Kekurangan penelitian tersebut adalah proses pendaftaran pasien rawat jalan hanya dapat dilakukan secara langsung di lokasi klinik. Yang menjadi perbedaan dengan penelitian saat ini adalah pada penelitian ini aplikasi yang dibangun merupakan berbasis android dan akan digunakan untuk pendaftaran pasien rawat jalan pada RSU Mitra Medika.

Regina Pricilia Robot, et. al., 2018 melakukan penelitian berjudul “Aplikasi Manajemen Rawat Inap dan Rawat Jalan di Rumah Sakit”. Pada penelitian tersebut dihasilkan sebuah aplikasi berbasis *web* untuk digunakan dalam proses pendaftaran pasien rawat jalan. Kekurangan dari penelitian tersebut

adalah proses pendaftaran pasien rawat jalan hanya dapat dilakukan secara langsung di lokasi rumah sakit yang menggunakan aplikasi yang dihasilkan. Yang menjadi perbedaan dengan penelitian saat ini adalah pada penelitian ini aplikasi yang dibangun merupakan berbasis android dan akan digunakan untuk pendaftaran pasien rawat jalan pada RSUD Mitra Medika.

II.2. Uraian Teoritis

II.2.1. Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari berbagai elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan merupakan tahapan yang berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras suatu sistem. (Kesumaningtyas dan Handayani ; 2020)

Perancangan merupakan penentuan proses dan data yang diperlukan oleh sistem baru. Manfaat tahap perancangan sistem ini memberikan gambaran rancangan bangun yang lengkap sebagai pedoman bagi programmer dalam mengembangkan aplikasi. Sesuai dengan komponen sistem yang dikomputerisasikan, maka yang harus didesain dalam tahap ini mencakup *hardware* atau *software*, *database* dan aplikasi. (Sitindaon dan Mulyono ; 2020)

Perancangan adalah tahapan yang dilakukan untuk memenuhi kebutuhan kepada pemakai sistem, dengan basis data yang bisa menyimpan data-data yang

diperlukan oleh pembuat keputusan dan memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada *programmer* dan ahli-ahli teknik lainnya yang terlibat. Apabila tujuan perancangan sistem kurang baik maka sistem yang telah dibuat akan berfungsi kurang baik atau gagal. Selain tujuan dari tahap rancangan sistem, perlu diketahui pula bahwa hasil akhir dari tahap rancangan sistem adalah suatu laporan spesifikasi teknis dari bentuk-bentuk output dan input serta spesifikasi teknik *software* yang akan berfungsi sebagai sarana pengolah data dan sekaligus penyaji informasi yang dibutuhkan. (Sitindaon dan Mulyono ; 2020)

II.2.2. Aplikasi

Aplikasi adalah sebuah perangkat lunak yang menjadi front end di sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang berguna orang-orang dan sistem yang bersangkutan. Istilah aplikasi berasal dari bahasa Inggris yaitu "*application*" yang berarti penerapan, lamaran atau pun penggunaan. Secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Firdausi dan Ramadhani ; 2020)

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan

untuk diterapkan menjadi sebuah bentuk yang baru. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. (Helmi Fauzi Siregar, et. al. ; 2018)

Aplikasi adalah salah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas. Aplikasi adalah program yang berisikan perintah-perintah untuk melakukan pengolahan data dengan membuat sistem atau program agar data diolah. misalnya *Microsoft Word* dan *Microsoft Excel*. Dari pengertian diatas Penulis menyimpulkan bahwa aplikasi adalah program siap pakai untuk melayani kebutuhan pengguna dalam berbagai aktifitas untuk pengolahan data. (Ana Naela Nurhayati, et. al. ; 2017)

II.2.3. Pengertian Pasien Rawat Jalan

Pasien adalah seseorang yang menerima perawatan medis, seringkali pasien menderita penyakit atau cedera dan memerlukan bantuan dokter untuk memulihkannya. Kata pasien dari bahasa Indonesia analog dengan kata *patient* dari bahasa inggris. *Patient* diturunkan dari bahasa latin yaitu *patiens* yang memiliki kesamaan arti dengan kata *pati* yang artinya menderita. (Agus Tugiaro, et. al. ; 2018)

Pelayanan rawat jalan adalah pencatatan yang diperlukan hanya sebatas catatan atau dokumen tentang identitas pasien, pemeriksaan fisik, diagnosis/masalah, tindakan/ pengobatan, dan pelayanan lain yang telah diberikan kepada pasien. (Regina Pricilia Robot, et. al. ; 2018)

Pelayanan rawat jalan (*ambulatory services*) adalah salah satu bentuk dari pelayanan kedokteran. Pelayanan rawat jalan adalah pelayanan medis kepada seorang pasien untuk tujuan pengamatan, diagnosis, pengobatan, rehabilitasi, dan pelayanan kesehatan lainnya, tanpa mengharuskan pasien tersebut dirawat inap. Keuntungannya, pasien tidak perlu mengeluarkan biaya untuk menginap (*opname*). Pelayanan rawat jalan adalah salah satu bentuk dari pelayanan kedokteran. Secara sederhana yang dimaksud dengan perawatan rawat jalan adalah pelayanan kedokteran yang disediakan untuk pasien tidak dalam bentuk rawat inap (*hospitalization*). (Yuda dan Yuri ; 2019)

II.2.4. Pengertian Database

Database merupakan sekumpulan dari data yang terintegrasi. Sistem *Database* merupakan sistem komputerisasi yang bertujuan untuk menyimpan sejumlah data sehingga memudahkan pengguna untuk mendapatkan dan mengupdate informasi sesuai dengan kebutuhan Informasi yang disimpan dalam *Database* dapat berupa *text* maupun angka. Salah satu penggunaan *Database* dalam kehidupan sehari-hari adalah pada aplikasi *software* pembayaran oleh kasir supermarket. Ketika melakukan pembayaran dikasir, kasir tersebut melakukan pekerjaan dengan cara menempelkan *bar-code* di salah satu sensor *infrared*. Kemudian muncul harga dari barang tersebut dalam komputer. Sehingga kasir tersebut tidak perlu mengetik ulang barang yang dibeli oleh pembeli tersebut. (Intan Mardiono ; 2019)

Aplikasi pembayaran oleh kasir merupakan salah satu contoh dari penggunaan *Database* yang terintegrasi oleh komputer. Dalam aplikasi komputer tersebut telah terekam data-data barang beserta harga, sehingga kasir dengan mudah mendapatkan data barang serta saat melakukan input barang, maka *Database* tersebut secara otomatis terupdate. Salah satu keuntungan dari *Database* adalah dapat mempersingkat waktu pekerjaan. Sebuah *Database* dapat menampung data yang kompleks yang dapat digunakan secara internal maupun eksternal. *Software* yang digunakan oleh suatu organisasi untuk mengendalikan *Database* disebut dengan *Database Manegement System* atau dapat disingkat dengan DBMS. DBMS merupakan suatu kumpulan program yang memudahkan pengguna untuk membuat dan mengelola *Database*. Data yang ada pada DBMS dapat dimanipulasi atau dikelola dengan menggunakan *query*, dimana dengan *query* tersebut pengguna dapat mendapatkan data spesifik yang dibutuhkan, serta dapat mengupdate informasi dalam data-data tersebut. (Intan Mardiono ; 2019)

Database sebagai kumpulan terorganisasi dari data-data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. Terminologi hubungan berarti data mendeskripsikan domain (ranah) tertentu sehingga pengguna mudah untuk mendapatkan jawaban atas pertanyaan yang diajukan ke basis data tersebut. Sedangkan pengertian sistem basis data adalah sebagai koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus). (Muhammad Saed Novendri, et. al. ; 2019)

II.2.5. Bahasa Pemrograman Java

Java adalah suatu teknologi di dunia *software* komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu *platform*. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua *platform*. Dan juga dirancang untuk menghasilkan aplikasi-aplikasi dengan performansi yang terbaik, seperti aplikasi *database* Oracle 8i/9i yang *core*-nya dibangun menggunakan bahasa pemrograman Java. (Ahmad Ibnu Mubarak ; 2021)

Java adalah bahasa pemrograman yang populer, dikembangkan oleh *Sun Microsystems*. Salah satu penggunaan terbesar Java adalah dalam pembuatan aplikasi *native* untuk Android. Bahasa pemrograman ini bersifat *multi platform* yakni bahasa ini dapat digunakan di berbagai *platform*, seperti *desktop*, *android* dan bahkan untuk sistem operasi *Linux*. Sedangkan Java bersifat *neutral architecture*, karena *Java Compiler* yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai *Java Bytecode*. (Ahmad Ibnu Mubarak ; 2021)

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut *class*. *Class* terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah

melakukan tugasnya. Para pemrogram Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). *Java Application Programming Interface* telah menyediakan fungsionalitas yang memadai untuk menciptakan applet dan aplikasi canggih. Ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana mempergunakan kelas pada *Java Application Programming Interface* (API). (Hidayat dan Sabrina ; 2018)

II.2.6. Android

Android merupakan sistem operasi berbasis Linux yang digunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA). Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk membantu kegiatan dalam berbagai bidang, sehingga bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer *tablet*. Android awalnya dikembangkan oleh Android, Inc dengan dukungan finansial *Google*, yang kemudian membelinya pada tahun 2005. (Suheri, et. al. ; 2021)

Pada tanggal 23 september 2008, sistem operasi Android versi 1.0 resmi diluncurkan. Sekitar sebulan berikutnya, pada tanggal 22 Oktober 2008, *smartphone* pertama yang menjalankan Android 1.0 itu, yaitu HTC *Dream*, diluncurkan ke pasar. Pada tanggal 9 februari, Android versi 1.1 diluncurkan untuk memperbaiki *bug* dari versi sebelumnya dan menambah fitur yang tersedia. Setelah versi 1.1, rilis Android berikutnya menggunakan nama makanan manis dengan urutan alfabetis, dimulai dengan 1.5 *Cupcake* yang diluncurkan pada tanggal 30 April 2009. Rilis-rilis Android selanjutnya, yaitu *Donut*, *Eclair*, *Froyo*, dan *Gingerbread* semua dibuat untuk *smartphone*. Namun, Apple meluncurkan iPad pada tahun 2010 dan meningkatkan ketertarikan masyarakat luas kepada *computer tablet*. Beberapa pengembang Android mencoba mengembangkan *tablet* Android untuk menyaingi iPad, seperti Samsung *Galaxy Tab* yang menggunakan *Gingerbread* yang dikustomisasi. Google dan OHA pun bergerak dengan melakukan pengembangan Android versi baru yang lebih optimal untuk *tablet*. (Yusfrizal ; 2019)

Pada tanggal 22 februari 2011, android *Honeycomb* diluncurkan ke pasar dan pada tanggal 24 februari 2011, *tablet* pertama yang menggunakan *honeycomb*, yaitu Motorola Xoom, diluncurkan ke pasar. Pada tanggal 19 Oktober 2011, Android meluncurkan *Ice Cream Sandwich* versi ini dapat bekerja secara optimal baik di *smartphone* maupun di *tablet*. Rilis android berikutnya, yaitu *Jelly Bean*, bertujuan untuk semakin meningkatkan apa yang sudah tersedia di *Ice Cream Sandwich*, dengan memperbaiki *bug-bug* dn menambahkan fitur-fitur. Pada tanggal 3 september 2013, diumumkan versi Android selanjutnya adalah Android

4.4 Kit Kat. Android sudah mendapatkan izin dari Nestle dan Hershey selaku pemilik dagang Kit Kat. Sebelum pengumuman ini, banyak yang berspekulasi bahwa versi Android berikutnya akan diberi nomor 5.0 dengan nama *Key Lime Pie* berikut adalah tabel untuk semua sistem operasi Android yang sudah diluncurkan sampai sekarang. Saat buku ini ditulis, sistem operasi Android yang terbaru adalah Android 6.0 Marshmallow. (Yusfrizal ; 2019)

Adapun penjelasan dari versi masing-masing Android adalah sebagai berikut : (Muhamad Zaelani ; 2017)

1. Android 1.0 : Astro (*Alpha*)

Kedua versi awal Android ini mungkin agak asing kamu dengarkan. Pasalnya versi Android 1.0 Astro (*Alpha*) dan Android 1.1 Walaupun belum menggunakan nama makanan manis, kedua sistem operasi Android ini tentu menjadi pionir. Pasalnya di sinilah Android bermula lewat *smartphone* pertama, HTC *Dream*.

2. Android 1.1 : Bender (*Beta*)

Android 1.1 *Bender (Beta)* ini belum diluncurkan secara publik untuk kebutuhan komersil. Platform Android sendiri pertama kali diluncurkan pada September 2008 dengan andil Andy Rubin yang saat ini dikenal sebagai Bapak Android. Walaupun belum menggunakan nama makanan manis, kedua sistem operasi Android ini tentu menjadi pionir. Pasalnya di sinilah Android bermula lewat *smartphone* pertama, HTC *Dream*. (Muhamad Zaelani ; 2017: 5)

3. *Cupcake*

Mulai dari versi ini, Android menggunakan nama makanan manis untuk setiap versi yang diluncurkan. Android 1.5 *Cupcake* sendiri dirilis pada tanggal 30 April 2009 dengan berbagai fitur di sebuah perangkat *smartphone* untuk menggantikan *featured phone* kala itu.

4. *Donut*

Tentu pada awal perilisannya, sistem operasi Android tetap memiliki banyak *bug* yang pengembangnya perlu mengadakan perbaikan. Hal ini dilakukan pada Android 1.6 *Donut* yang dirilis pada 15 September 2009. Yang artinya belum genap setahun semenjak perilisan Android 1.5 *Cupcake* atau hanya berselang 5 bulan saja. Android pun menambahkan beberapa pembaruan, terutama dukungan pada layar *smartphone* yang lebih besar.

5. *Eclair*

Sama seperti versi sebelumnya, Android 2.0 & 2.1 *Eclair* masih berfungsi untuk menutupi *bug* yang masih ditemukan pada sistem operasi *mobile* ini. Di samping itu, Android juga menambah berbagai fitur di dalamnya. Mulai dari dukungan *Bluetooth* hingga fitur kamera yang mulai menjadi nilai jual *smartphone* kala itu. Android 2.0 & 2.1 *Eclair* digunakan pada perangkat seperti HTC Nexus One.

6. Android 2.2 : *Froyo* (*Frozen Yoghurt*)

Mulai versi ini Android tampaknya sudah mulai dikenal luas oleh berbagai brand *smartphone*. Android 2.2 *Frozen Yoghurt* alias *Froyo* ini dirilis pertama kali pada tanggal 20 Mei 2010. Walaupun sudah mulai

dipergunakan pada beberapa brand, namun tetap saja Android masih kalah bersaing dengan *Symbian* yang mendominasi pasar *featured phone*. Android 2.2 *Froyo* memberikan peningkatan pada kecepatan kerja, fitur USB *tethering*, *WiFi hotspot*, serta fitur keamanan.

7. *Gingerbread*

Belum setahun berselang lagi, Android 2.3 *Gingerbread* kembali diluncurkan pada Desember 2010 dengan berbagai peningkatan yang cukup signifikan. Hal ini terutama pada tampilan tatap muka alias *user interface* yang digunakan. Mulai versi ini, banyak brand *smartphone* mulai melirik menggunakan sistem operasi Android. Salah satunya Samsung *Galaxy Series* yang populer hingga saat ini.

8. *Honeycomb*

Untuk para pengguna *smartphone* mungkin akan agak asing dengan versi Android yang satu ini. Android 3.0 & 3.2 *Honeycomb* yang menggunakan ikon lebah ini memang diperuntukkan penggunaannya khusus untuk perangkat *tablet*. Tentu perilis Android 3.0 & 3.2 *Honeycomb* pada 10 Mei 2011 ini untuk mendukung Samsung yang mulai merilis perangkat *tablet*, Samsung *Galaxy Tab Series* untuk menyaingi Apple iPad.

9. *Ice Cream Sandwich* (ICS)

Selain itu, Android pun juga merilis versi Android 4.0 *Ice Cream Sandwich* yang kembali diperuntukkan untuk perangkat *smartphone*. Android 4.0 *Ice Cream Sandwich* sendiri dirilis pada 19 Oktober 2011 silam. Punya versi nama paling panjang hingga saat ini, Android 4.0 *Ice Cream Sandwich*

memberikan banyak pembaruan. Mulai dari animasi yang semakin halus, sederhana, dan mudah digunakan.

10. *Jelly Bean*

Peningkatan signifikan terasa saat menggunakan Android 4.1 & 4.3 *Jelly Bean*. Sistem operasi ini sendiri pertama kali dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis. Dengan begini, tentu Android 4.1 & 4.3 *Jelly Bean* bisa memberikan peningkatan fungsi pada *user interface* dan teknologi *Vsync* yang digunakannya.

11. KitKat

Menggunakan nama *brand* cemilan terkenal, Android 4.4 KitKat pertama kali dirilis pada Oktober 2013. Versi Android terbaik ini pun bisa dikatakan menjadi favorit karena mendukung hampir seluruh perangkat *smartphone* di dunia. Hal ini dikarenakan Android 4.4 KitKat dapat memberikan optimalisasi yang baik termasuk pada perangkat *smartphone* yang memiliki spesifikasi kurang mumpuni alias cukup rendah saat itu.

12. *Lollipop*

Mulai beberapa versi ke belakang, Android dan Google pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun. Termasuk Android 5.0 & 5.1 *Lollipop* yang dirilis dan diresmikan pada Juni 2014. Bisa dibilang Android 5.0 & 5.1 *Lollipop* menjadi pionir dibuatnya *smartphone flagship* dengan spesifikasi cukup mumpuni. Versi Android ini sudah mendukung arsitektur 64-bit yang sudah memungkinkan

penggunaan RAM di atas 3GB. Salah satunya ASUS Zenfone 2 yang sudah mengusung RAM 4GB saat itu.

13. *Marshmallow*

Android 6.0 *Marshmallow* menjadi suksesor dari versi Android sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2015 dan mulai dirilis pada Oktober 2015 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya *fingerprint* sensor sebagai sistem keamanan *biometrik* yang digunakan. Selain digunakan untuk mengunci layar, *fingerprint* sensor ini dapat digunakan untuk otentikasi *Google Play Store* dan pembelian dengan menggunakan *Android Pay*.

14. *Nougat*

Untuk saat ini, sistem operasi Android ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini. Android 7.0 & 7.1 *Nougat* pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot Android dengan batangan Nougat. Sistem operasi Android 7.0 & 7.1 *Nougat* mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga fitur *splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus.

15. Android Oreo 8.0

Android Oreo 8.0 hadir pada 21 Agustus 2017 dengan fitur-fitur yang *multitasking* serta penampilan UI yang lebih rapi.

16. Android Pie 9.0

Pie 9.0 dirilis pada 6 Agustus 2018 dengan membawa *smart reply*, *digital wellbeing*, *adaptive battery* dan lain-lain.

17. Android 10

Salah satu alasan versi ke 10 dinamakan Android 10 dikarenakan tidak adanya nama makanan yang berawalan Q. (Muhamad Zaelani ; 2017)

II.2.7. Android Studio

Android studio merupakan sebuah *Integrated Development Environment* (IDE) yang digunakan untuk mengembangkan sebuah aplikasi android. Android studio digunakan sebagai alat pengembangan aplikasi android. Android studio ialah IDE resmi untuk keperluan pengembangan aplikasi android. Android studio juga merupakan pengembangan dari eclipse, namun memakan RAM lebih besar daripada eclipse. (Fauzan Fadli Shan, et. al. ; 2020)

Android studio merupakan lingkungan pengembangan perangkat lunak terpadu *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio juga menawarkan banyak fitur untuk meningkatkan produktivitas saat membuat aplikasi Android. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip

dengan Eclipse disertai dengan ADT *plugin* (*Android Development Tools*).

Android Studio memiliki fitur :

1. Projek berbasis pada *Gradle Build*
2. *Refactory* dan pembenahan *bug* yang cepat
3. *Tools* baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibilitas aplikasi dengan cepat.
4. Mendukung *Proguard And App-signing* untuk keamanan.
5. Memiliki GUI aplikasi android lebih mudah
6. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan. (Suheri, et. al. ; 2021)

11.2.8. Android SDK (*Software Development Kit*)

Android SDK mencakup perangkat *tools* pengembangan yang komprehensif. android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, dengan menggunakan *plugin Android Development Tools* (ADT), dengan ini pengembang dapat menggunakan ide untuk mengedit dokumen java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi android dan pengendalian perangkat android. (Dicky Taruna, et. al. ; 2021)

Android SDK adalah *kit tools* dari *platform* android khususnya bagi para *programmer* yang mengembangkan aplikasi berbasis android. Di dalam android SDK terdapat beberapa *tools* terdiri dari *debugger*, *software libraries*,

emulator, dokumentasi, contoh kode, dan tutorial. (Muhammad Yusuf Rambe, et. al. ; 2020)

Android SDK adalah *tool* untuk mengakses *library* Android dan menggunakannya untuk mengembangkan aplikasi Android. Android SDK terdapat *file-file* dan *utilities* (alat bantu) lainnya yang berfungsi untuk mempermudah pembuatan aplikasi Android secara cepat. (Febrina Cokro, et. al. ; 2019)

II.2.9. Pengertian UML

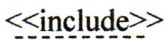
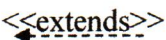
UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Julianto dan Setiawan ; 2019). Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

II.2.9.1. Use Case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Julianto dan

Setiawan ; 2019). Berikut ini adalah simbol-simbol diagram *use case*, seperti yang terlihat pada tabel II.1 dibawah ini :

Tabel II.1. Use Case Diagram

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktir, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i>
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat

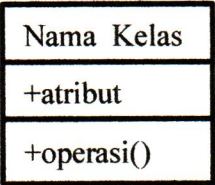
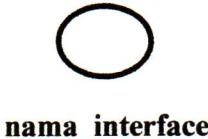
(Sumber : Julianto dan Setiawan ; 2019)

II.2.9.2. Class Diagram

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.

(Julianto dan Setiawan ; 2019). Berikut ini adalah simbol-simbol diagram *class diagram*, seperti yang terlihat pada tabel II.2 dibawah ini :

Tabel II.2. Class Diagram

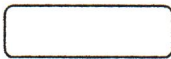
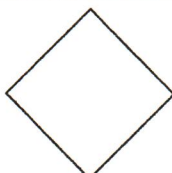
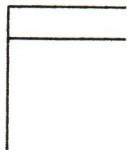
Gambar	Keterangan
	<i>Kelas</i> pada struktur sistem
	<i>Antar Muka atau Interface</i> sama dengan konsep interface dalam pemograman berorientasi objek
	<i>Asosiasi atau Association</i> Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
	<i>Generalisasi</i> Relasi antar kelas dengan makna generalisasi-generaliasi spesialisasi

(Sumber : Julianto dan Setiawan ; 2019)

II.2.9.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan aktor (Julianto dan Setiawan ; 2019). Berikut adalah simbol-simbol *activity diagram*, seperti terlihat pada tabel II.3 berikut ini :

Tabel II.3. *Activity Diagram*

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa

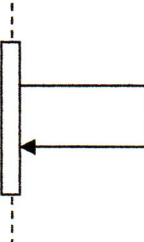
(Sumber : Julianto dan Setiawan ; 2019)

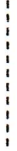
II.2.9.3. *Sequence Diagram*

Diagram sekuen “menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode

yang dimiliki kelas yang diinstansiasi menjadi objek itu”. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Julianto dan Setiawan ; 2019). Berikut adalah simbol-simbol *sequence* diagram, seperti terlihat pada tabel II.4 berikut ini:

Tabel II.4. *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i> , simbol mengirim pesan antar <i>class</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi

	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>
---	---

(Sumber : Julianto dan Setiawan ; 2019)