BAB II
LANDASAN TEORI

2.1 Android Sdk

Merupakan tools API (*Application Programming Interface*), diperlukan guna memulai pengembangan aplikasi pada Android menggunakan bahasa pemrograman Java. Android adalah perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh *Google*. Saat ini disediakan Android SDK (*Software Development Kit*), untuk alat bantu dan API untuk memulai pengembangan aplikasi pada platform *Android* menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, *Android* memberikan anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone atau Smartphone (SAGITA, 2016).

2.2 Android Studio

Menurut Palupi et all (2018) *android studio* adalah sebuah IDE (*Integrated Development Environment*) yang digunakan untuk pengembangan aplikasi *Android*. *Android Studio* memiliki fitur editor kode cerdas (*Intelligent Code Editor*) yang memiliki kemampuan penyelesaian kode, optimalisasi, dan analisis kode yang canggih. Selain itu fitur *New Project Wizards* membuat proses memulai proyek baru menjadi jauh lebih mudah bahkan dapat mengimpor contoh kode *Google* dari *GitHub*. Berbagai modul baru digunakan dalam *Android Studio* ini, salah satunya adalah pengembangan aplikasi multilayar yang memudahkan pengembangan untuk membangun sebuah aplikasi untuk ponsel dan tablet *Android*, *Android Gear*, *Android TV*, *Android Auto*, *Android Google Glass*.

2.3 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer, mengembangkan bagian back-end dari software, website, aplikasi android

termasuk telepon genggam. Dikenal memiliki moto *"Write once, Run Anywhere"*. Artinya *Java* mampu dijalankan diberbagai platform tanpa harus menyusun ulang penyesuaian platformnya. Semisal berjalan di *Android*, *Windows*, *Linux* dan lain sebagainya. James Gosling adalah sosok yang berperan sebagai pencipta pertama bahasa ini saat masih bergabung di Sun Microsystems kemudian menjadi bagian dari Oracle dan dirilis pada tahun 1995. Bahasa ini banya mengandung sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi yang berbasis java umumnya dikompilasi ke dalam p-code dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin.

2.4 Firebase

Firebase adalah layanan dari Google yang digunakan agar mempermudah para pengembang aplikasi untuk mengembangkan aplikasi. Firebase atau BaaS (Backend as a Service) merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan seorang developer.

2.4.1 Fitur Firebase

1. Firebase Analytics

   Adalah salah satu fitur pada Firebase yang digunakan sebagai koleksi data dan reporting untuk aplikasi Android maupun iOS Fitur ini memiliki kelebihan yang memungkinkan kita untuk bisa membuat segmentasi user berdasarkan user attribute. User attribute adalah suatu parameter yang bisa digunakan sebagai filter yang bertujuan untuk reporting dan notifikasi.

2. Firebase Cloud Messaging and Notifications

   Dengan fitur ini kita dapat mengirim dan menerima pesan serta notifikasi di Android, iOS dan web tanpa perlu biaya dan tentunya hemat baterai antar server maupun antar device.

3 Firebase Authentication

Firebase authentication adalah salah satu fitur *back-end*, fitur Android dan iOS, SDK yang mudah digunakan dan tampilan *interfaces* yang siap pakai untuk mengautentikasi pengguna ke aplikasi yang dibuat. Firebase Authentication mendukung autentikasi menggunakan nomor telepon, sandi, penyedia identitas gabungan populer seperti Facebook, Google Mail, Twitter, dan sebagainya.

4 Firebase Cloud Firestore

Adalah database yang bersifat fleksibel dan terukur untuk pengembangan perangkat seperti seluler, website, dan server di Firebase dan Google Cloud Platform. Sama seperti Firebase Realtime Database, Cloud Firestore membuat data user tetap terkoneksi di aplikasi user melalui listener realtime dan menawarkan layanan secara offline untuk aplikasi seluler dan web.

5 Firebase Realtime Database

Firebase Realtime Database adalah database yang dihost melalui cloud. Data disimpan dan dieksekusi dalam bentuk *JSON* dan disinkronkan secara *realtime* ke setiap *user* yang terkoneksi. Hal ini dilakukan supaya memudahkan *developer* dalam mengelola suatu database dalam skala yang cukup besar.

6 Firebase Hosting

Firebase Hosting adalah suatu layanan hosting konten web. Hanya dengan satu instruksi, maka kita dapat mengimplementasikan aplikasi web serta menyajikan konten statis maupun dinamis ke CDN (jaringan penayangan konten) global dengan cepat. Kegunaannya sendiri yakni bisa menayangkan konten melalui koneksi yang begitu aman, mengirimkan konten secara cepat, dan mendukung semua jenis konten untuk di hosting.

2.5 UML (*Unified Modeling Language*)

Menurut Sri Mulyani (2019) uml adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML juga dapat didefinisikan sebagai yakni bahasa standar visualisasi, pendokumentasian sistem, perancangan atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software. Dengan UML, maka diharapkan bisa mempermudah pengembangan perangkat lunak serta memenuhi semua kebutuhan pengguna dengan tepat, efektif dan lengkap. UML memiliki diagram-diagram yang digunakan pada saat pembuatan aplikasi berorientasi objek, contohnya sebagai berikut :

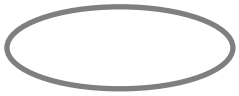
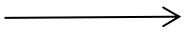
1) Use Case Diagram

Adalah jenis dari diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan pengguna. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan apa yang akan dilakukan oleh si pengguna dalam sistem itu sendiri seperti yang terdapat pada *Use Case. Use case* harus bisa menggambarkan bagaimana urutan pengguna dalam menggunakan sistem atau didalam sistem. Model *use case* bisa dijabarkan dalam diagram *use case*, namun diagram tidaklah identik dengan model karena model lebih luas dari diagram.

Berikut ini adalah simbol-simbol yang biasanya digunakan didalam *use case* diagram.

Tabel 2.1 Use Case Diagram

Simbol	Nama	Keterangan
	Pengguna	Mewakili dari peran orang, sistem yang lain, atau alat saat berkomunikasi dengan use case.

	Use Case	Abstrak dan interaksi antara pengguna dan sistem.
	Association	Abstraksi dari penghubung antara pengguna dan use case.

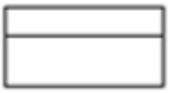
2. *Class* Diagram

   Class diagram yaitu salah satu jenis diagram di UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. *Class* memiliki tiga area pokok yakni :

   - Nama, *class* yang digunakan harus memiliki sebuah nama.
   - Atribut, kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
   - Operasi, proses yang hanya bisa dilakukan oleh sebuah *class*, baik pada kelas itu sendiri atau pada kelas lainnya.

Tabel 2.2 Class Diagram

Simbol	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (descendent) berbagi pelaku dan struktur data dari objek yang ada diatasnya objek induk (ancestor).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

3. Activity Diagram

   Diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. Diagram aktivitas juga bisa menggambarkan proses lebih dari satu aksi dalam waktu bersamaan.

Tabel 2.3 Activity Diagram

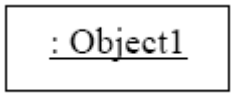
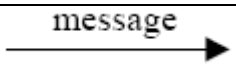
Simbol	Nama	Keterangan
	*Initial Activity*	Awal aktivitas modul sistem aplikasi
	*Action*	Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi
	*Decision*	Digunakan untuk menggambarkan suatu keputusan atau tindakan

		yang harus diambil pada kondisi tertentu.
	*Activity Final Node*	Bagaimana objek dibentuk dan diakhiri.

4. *Sequence* Diagram

   Sequence diagram adalah salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu. Sequence diagram bisa juga menggambarkan urutan atau tahapan yang harus dilakukan untuk bisa menghasilkan sesuatu seperti pada use case diagram.

Tabel 2.4 Sequence Diagram

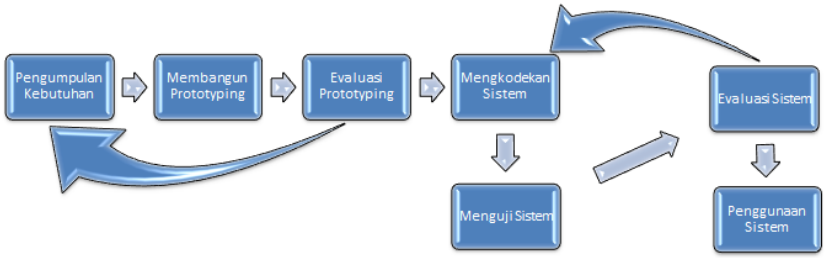
Simbol	Nama	Keterangan
	Object	Instance dari sebuah class dan dituliskan tersusun secara horizontal.
	Pengguna	Pengguna dapat juga berkomunikasi dengan objek, maka dapat diurutkan sebagai kolom. Simbol pengguna sama dengan simbol pada use case.
	*Message*	Digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara objek-objek.

	*Activation*	*Activation* dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah *lifeline*. Activation mengindikasikan sebuah objek yang akan melakukan sebuah aksi.
	Lifeline	*Lifeline* mengindikasikan keberadaan objek dalam basis waktu. Notasi untuk *lifeline* adalah garis putus-putus vertikal yang ditarik dari sebuah objek.

2.6 Metode Pengembangan Sistem

Pada penelitian ini, metode yang digunakan untuk pengembangan sistem adalah model *prototyping*. Pengertiannya adalah metode pengembangan yang sangat cepat serta pengujian model kerja aplikasi baru melalui proses interaksi yang berulang-ulang sehingga aplikasi bisa digunakan dengan baik. Yang menjadi ciri khas dari metode prototype ini adalah pengembang sistem, client, pengguna akhir, dapat melihat dan langsung melakukan eksperimen sejak awal proses pengembangan. Didalam proses pengembangannya, ada beberapa metode untuk mengerjakan prototype. Berikut ini adalah beberapa tahapan metode Prototype:

2.6.1 Tahapan Metode Prototype



Gambar 2.1 Tahapan Metode Prototype

1. Mengumpulkan Kebutuhan

   Untuk awalnya, hal yang harus dilakukan dalam tahapan metode prototype adalah mengidentifikasi seluruh perangkat dan juga permasalahan. Yang sangat penting dalam tahapan metode prototype adalah analisis dan identifikasi kebutuhan garis besar sistem. Kemudian baru diketahui langkah apa yang akan diperbuat dan permasalahan yang akan dibuat dan penyelesaian solusinya. Pengumpulan kebutuhan sangat penting didalam proses ini.

2. Membangun Prototype

   Setelah mengumpulkan kebutuhan, maka langkah selanjutnya adalah membangun prototype yang berfokus penyajian pelanggan. Misalnya membuat input dan output hasil dari sistem. Untuk sementara baru prototype saja kemudian baru akan ada tindak lanjut yang harus dikerjakan.

3. Evaluasi Prototype

   Sebelum lanjut ke tahap selanjutnya, maka penting untuk kita mengevaluasi dari tahap 1 dan 2. Karena ini sebagai penentu keberhasilan dan proses yang sangat penting dilakukan. Apabila pada tahap 1 dan 2 ada yang kurang atau ada kesalahan, maka untuk ke depannya akan sulit sekali melanjutkan langkah selanjutnya.

4. Mengkodekan Sistem

   Sebelum pengkodean atau kita lebih mengenalnya dengan proses koding, pada proses ini kita menggunakan bahasa pemrograman dan juga biasanya

sangat sulit karena kita harus mengaplikasikan kebutuhan yang dibutuhkan ke dalam bentuk kode program.

5. Menguji Sistem

   Setelah proses koding, kita tentu saja akan melakukan pengujian. Cara untuk menguji bisa menggunakan whitebox atau blackbox. Whitebos yang berarti menguji kodingan sedangkan blackbox yang menguji apakah fungsi-fungsi dari tampilan sudah benar dengan aplikasinya atau belum.

6. Evaluasi Sistem

   Pada tahap ini, kita mengevaluasi dari semua langkah yang sudah dilakukan apakah sudah sesuai dengan kebutuhan yang diperlukan atau belum. Apabila kebutuhan masi belum terpenuhi maka bisa diulangi lagi dari tahap 1 dan 2.

7. Menggunakan Sistem

   Pada tahap ini, maka sistem sudah selesai dan layak digunakan kemudian diserahkan pada pengguna. Perlu diketahui bahwa aplikasi yang sudah berhasil dibuat, harus dilakukan maintenance secara berkala agar sistem bisa tetap terjaga dan berfungsi sebagaimana harusnya.

2.7 Real Time Chat

Chatting mengacu pada segala bentuk komunikasi menggunakan internet, secara spesifik mengacu kepada percakapan berbasis internet antara dua orang atau lebih. Percakapan di internet dapat menggunakan perangkat lunak seperti pengirim whatsapp, line, telegram, MiChat, dan lain-lain. Percakapan bisa berupa teks atau suara, mengirim pesan dengan teks atau suara kepada pengguna lain yang sedang online dan pengguna tersebut membalas pesan anda dengan teks atau suara, itulah proses terjadinya chatting. Ada hal-hal yang harus diperhatikan dalam melakukan proses chatting, yakni :

1. Real Time

Aplikasi chatting dijelaskan sebagai suatu aplikasi yang memungkinkan pengguna berkomunikasi tekstual secara langsung dengan pengguna lain.

2. *Nickname*

Pengguna harus memasukkan *nickname* sebagai identifikasi unik didalam suatu sistem / aplikasi. Berguna agar pengguna aplikasi lain bisa mengenali kita atau dikenali oleh sistem selama masa sesi berlangsung hingga si pengguna keluar dari ruang chat.

3. *Public Room*

Komunikasi yang terjadi diantara 2 pengguna atau lebih, terjadi didalam ruang chatting virtual. Saat masuk, pengguna langsung tergabung ke dalam public room dimana ruang tersebut pesan dari seorang pengguna bisa dilihat dan dibaca oleh semua pengguna lain yang berada didalam ruang chatting virtual.

4. *Send Message*

Saat pengguna selesai mengetik isi pesan, pengguna menekan tombol enter / send message untuk mengirim pesan tersebut. Teks pesan percakapan dari semua pengguna didalam suatu area tampilan percakapan diurutkan berdasarkan waktu diterimanya pesan tersebut oleh server. Pesan yang paling baru ditampilkan dipaling bawah agar tidak membingungkan pengguna lain.

5. *Auto Refresh*

Auto refresh berguna untuk meng-update secara otomatis setiap beberapa detik sekali dan pada saat pengguna mengirimkan atau menerima pesan baru sehingga pengguna dapat selalu mengetahui percakapan pengguna lain yang sedang berlangsung tanpa harus melakukan refresh halaman secara manual.

2.7.1 Konsep Chat Real Time Pada Aplikasi Our Paws

Pada aplikasi our paws, pengguna aplikasi dapat membuka obrolan cukup dengan men-download aplikasi Our Paws yang ada pada Google PlayStore, melakukan pendaftaran akun, login, kemudian melihat / membuat postingan tentang apa yang dibutuhkan. Konsep yang digunakan pada real time chat pada aplikasi Our Paws adalah relasi one-to-one. Relasi one to one adalah relasi dimana setiap satu baris data pada tabel satu hanya berhubungan dengan satu baris data di tabel dua.

Artinya masing-masing hanya memiliki satu hubungan saja. Relasi ini digunakan pada relasi pengguna dan userlogin. Dimana satu pengguna hanya memiliki satu akun untuk login dan satu akun login hanya dimiliki oleh pengguna.

2.6 Penelitian Terkait

Berikut ini adalah tabel penelitian terkait dengan real time chat :

Tabel 2.5 Penelitian Terkait Tentang Real Time Chat dan Database Firebase

Nama Peneliti	Judul Penelitian	Tahun	Hasil Penelitian
Muhammad Agung, Roslina, Ria Eka Sari	Implementasi Aplikasi Pembuatan Chat	2020	Pengguna *smartphone* umumnya membutuhkan sebuah media untuk bertukar pesan yang mudah dan dapat digunakan secara efisien kapanpun dan dimanapun, aplikasi ini telah dibangun dan dirancang menggunakan perangkat lunak Android Studio, dapat dijalankan dengan baik pada *smartphone* android tanpa *forceclose*, dapat digunakan kegiatan bertukar pesan antar sesama pengguna pada

			smartphone android secara *realtime* menggunakan koneksi internet.
Agus Tri Wahyu	IMPLEMENTASI FIREBASE UNTUK PEMESANAN SERVIS MOTOR BERBASIS ANDROID	2020	Aplikasi implementasi *Firebase Api* pada pemesanan jasa servis berbasis android dapat menyimpan data pengguna dan jasa servis pada database *firebase*, aplikasi servis online *implementasi firebase* ini dapat menampilkan data dari *firebase database* untuk ditampilkan di aplikasi atau *frontend*, aplikasi telah berhasil mempertemukan antara pengguna dengan jasa servis pada pemesanan dengan memanfaatkan teknologi *Google Maps, API,* untuk mempertemukan antara pengguna dan jasa servis.
Komang Aryasa, Yuego Elly Kurniawan	Implementasi *Firebase Realtime*	2019	Firebase realtime database telah diterapkan pada aplikasi sehingga

	Database Untuk Aplikasi Pemesanan Menu Berbasis Android.		bagian dapur dapat menerima orderan pesanan secara realtime dan aplikasi yang dihasilkan pada penelitian ini dapat digunakan untuk melakukan pemesanan menu pada smartphone android.