BAB II
LANDASAN TEORI

2.1 Mobile

Dalam (Prakarsya, A., 2019) Menjelaskan bahwa, *mobile* merupakan suatu istilah yang digunakan untuk menggambarkan suatu aplikasi pada piranti yang berukuran kecil, portable, dan wireless serta mendukung komunikasi. pengguna menginginkan perangkat yang kecil untuk kenyamanan dan mobilitas mereka dan Perangkat mobile juga hanya menghabiskan sedikit daya dibandingkan dengan mesin desktop.

Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* merupakan sebutan untuk aplikasi yang berjalan di *mobile device*. Dengan menggunakan aplikasi *mobile,* dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya.

2.2 Android

Dalam (Sari, Y. P., & Ali, R, 2019) menjelaskan bahwa *Android* merupakan sebuah sistem operasi untuk smartphone dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan antara piranti (*device*) dan penggunanya, sehingga *user* bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi – aplikasi yang tersedia pada *device*. *Android* pertama kali dirilis pada bulan oktober 2003 oleh Andy Rubin, Rich Miner, Nich Sears dan Chris White di bawah sebuah perusahaan bernama *android Inc* di Palo Antom, California. Dan akhirnya *android* diakuisisi oleh *google* pada tahun 2005. *Android* sendiri memiliki beberapa kelebihan yaitu adalah sebagai berikut :

a. *Open Source* alias Gratis
b. Cepat dan *Responsive*
c. *User Friendly*
d. Variasi harga produk yang beragam
e. *Google* sebagai pengembang
f. *Hardware* pendukung yang beragam

2.3 Perangkat Lunak Pengembangan Sistem

Pada pengembangan sistem untuk membangun sebuah aplikasi pengenalan dan pusat informasi adat batak Toba berbasis android diperlukan beberapa perangkat lunak yang digunakan dalam membangun aplikasi tersebut. Beberapa perangkat lunak yang digunakan adalah sebagai berikut :

2.3.1 Android studio

Firly (2018:13) menjelaskan bahwa Android studio merupakan integrated development environment (IDE) atau dalam artian lain merupakan sebuah lingkungan pengembangan terintegrasi resmi yang memang merancang khusus untuk pengembangan system operasi Google Android. Aplikasi ini dibangun di atas sebuah perangkat lunak yang dinamakan Intellij IDEA milik JetBrains. Bisa juga dibilang bahwa android studio merupakan pengganti dari Eclipse android development tool atau ADT sebagai IDE utama dalam pengembangan aplikasi android yang asli.

Android studio diluncurkan pada tanggal 16 mei 2013 dalam konferensi google I/O yang pada saat itu masih dalam tahap pratinjau akses versi 0.1 sebagai perintis. Hingga pada akhirnya versi stabil 3.0 yang liris pada pertengahan bulan oktober 2017 dan menjadi software terlaris dikalangan developer muda. Aplikasi ini dapat digunakan diberbagai sistem operasi yaitu windows,linux dan macOS.

Aplikasi ini menawarkan berbagai fitur canggih yang akan meningkatkan kemampuan produktivitas dalam proses pengembangan aplikasi. Berikut ini adalah beberapa hal yang akhirnya banyak mengundang developer untuk melirik android studio sebagai software pengembang :

a. Dukungan dari C++, NDK dan sekarang kotlin
b. Perkembangan yang up to date
c. Sistem berbasis Gradle yang dinilai fleksibel
d. Lingkungan yang mencakup seluruh perangkat android
e. Emulator yang cepat dan kaya akan fitur

f. Alat pengujian dan kerangka yang juga ekstensif

g. Instant Run

h. Dukungan google cloud platform

2.3.2 Java

Firly (2018:19) menjelaskan bahwa Java merupakan bahasa pemrograman multi platform. Java tidak menyediakan IDE khusus seperti halnya bahasa pemrograman yang lain. Pemrogram bisa menggunakan IDE yang support ke java, misalnya Netbeans, Eclips, TexPad, dan lain-lain. elemen-elemen dasar pemrograman Java terdiri dari Himpunan karakter, Pengenal (identifier), Kata Kunci, Tipe Data Primitif. Tipe data primitif yang didukung oleh bahasa pemrograman Java yaitu byte, short, int, long, float, double, Boolean, char.

2.3.3 Firebase

Menurut (Google Firebase, 2011), *Firebase Realtime Database* merupakan *database* yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika *developer* membuat aplikasi platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *Instance Realtime Database* dan menerima update data terbaru secara otomatis. Menurut Google Firebase (2016), setelah API *Firebase* dimasukkan ke Dalam aplikasi Android atau iOS, pengembang bisa menggunakan fitur *Firebase* dengan coding yang simpel. Beberapa fitur yang disediakan *Firebase,* yaitu :

1. Analytics

Fitur ini membuat para *develope*r aplikasi dapat mengerti bagaimana para pengguna menggunakan aplikasi mereka. SDK *analiytics* menangkap sendiri data yang dibutuhkan oleh para *developer*. Dashboard juga menyediakan detil seperti pengguna paling aktif atau fitur apa saja yang paling digunakan dalam aplikasi tersebut. *Analytic*s juga menyediakan data yang telah dirangkum.

2. Authentication

Fitur ini membuat para *developer* dapat mengijinkan pengguna untuk mengakses aplikasi. *Firebase* menyediakan fitur login melalui Gmail, Github, Twitter, Facebook dan juga autentikasi buatan sendiri.

3. Messaging

FCM (*Firebase Cloud Messaging*) membuat para pengguna dapat mengirimkan pesan ke berbagai platform tanpa biaya tambahan. *Messaging* juga dapat digunakan untuk kebutuhan notifikasi.

4. Real-time Database

Database di *Firebase* merupakan database berbasis cloud dan tidak membutuhkan SQL untuk mengambil dan menyimpan data atau bisa disebut juga NoSQL. Database ini sangatlah cepat dan dapat diandalkan yang artinya data dapat dibaharui dan disinkronisasikan dengan cepat. Data juga dijaga meskipun pengguna kehilangan koneksi internetnya.

5. Storage

Firebase juga menyediakan fasilitas penyimpanan. *Firebase* dapat menyimpan dan mengambil konten seperti gambar, video, dan audio langsung dari SDK. Meng-*upload* dan men-*download* juga dilakukan di *background*. Data yang disimpan akan aman dan hanya pengguna yang diijinkan yang dapat mengaksesnya.

6. Hosting

Firebase juga menyediakan fitur *hosting*. *Firebase* mengirimkan konten web secara cepat dan konten selalu dikirim dengan aman.

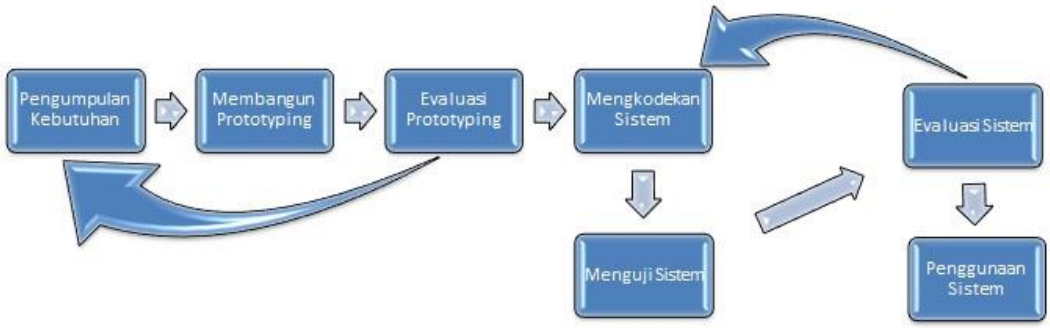
7. Crash reporting

Fitur *crash reporting* di *Firebase* membuat pengembang dapat mengetahui kesalahan ketika terjadi *crash*.

2.4 *Metode Pengembangan Perangkat Lunak*

Prototyping perangkat lunak (software prototyping) atau siklus hidup menggunakan protoyping (life cycle using prototyping) merupakan salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (working model). Tujuannya untuk mengembangkan model menjadi sistem final. Artinya

sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biayanya menjadi lebih rendah. Ada banyak cara untuk memprotoyping, begitu pula dengan penggunaannya. Ciri khas dari metodologi ini adalah pengembang sistem (system developer), klien, dan pengguna dapat melihat dan melakukan eksperimen dengan bagian dari sistem komputer dari sejak awal proses pengembangan.



Gambar 2. 1 Metode *Prototype*

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun prototyping

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).

3. Evaluasi protoptyping

Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototyping direvisi dengan mengulang langkah 1, 2 , dan 3.

4. Mengkodekan sistem

Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

2.5 Pengujian Kotak Hitam

Pressman (2012) menjelaskan bahwa pengujian kotak hitam, juga disebut pengujian perilaku, yang berfokus pada persyaratan fungsional perangkat lunak. Teknik Pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut :

1. fungsi yang salah atau hilang,
2. kesalahan antarmuka,
3. kesalahan dalam struktur data atau akses basis data eksternal,
4. kesalahan perilaku atau kinerja, dan
5. kesalahan inisialisasi dan penghentian.

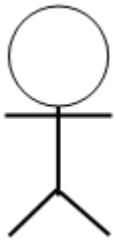
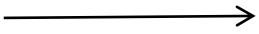
2.6 Unifed Modelling language (UML)

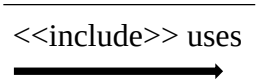
Rosa (2018:133) menjelaskan bahwa *Unifed Modeling Language* (UML) merupakan metode perancangan berortasi objek yang memeriksa syarat – syarat dari sudut pandang kelas-kelas dan objek yang ditemui pada ruang lingkup permasalahan dengan tujuan untuk memahami domain masalah dan meningkatkan ketelitian, konsistensi, kelengkapan analisis. *Unified Modeling Language* (UML) adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.

2.6.1 *Use Case Diagram*

Rosa & Shalahuddin (2018:155) menguraikan bahwa *Use Case Diagram* merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya use case tidak hanya penting pada tahap analisis, tetapi juga sangat penting untuk perancangan, untuk mencari kelas-kelas yang terlibat dalam aplikasi, serta untuk melakukan pengujian. Diagram use case bersifat statis, diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Simbol – simbol *Use Case* dapat dilihat pada tabel 2.1.

Tabel 2. 1 *Simbol Use Case Diagram*

Simbol	Keterangan
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit dan aktor.
Aktor / *Actor* 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.
Asosiasi / *Association* 	Komunikasi antar aktor dan *Use Case* yang berpartisipasi.
Ekstensi / *extend* <<extend>> 	Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambah dapat berdiri sendiri walau tanpa *Use Case* tambahan.

Generalisasi / generalization 	Hubungan generalisasi dan spesialisasi antara dua buah Use Case yang mana fungsi yang satu lebih umum dari yang lainnya.
Include / Use Case  <<include>> uses	Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambahkan memerlukan Use Case ini untuk menjalankan fungsinya.

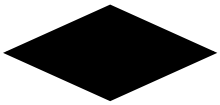
Tabel 2. 1 *Simbol Use Case Diagram* (lanjutan *)*

2.6.2 *Activity Diagram*

Rosa & Shalahuddin (2018:161) menguraikan bahwa *Activity diagram* menggambarkan workflow (alir kerja) atau aktivitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. Tahap perancangan activity diagram menjabarkan masing – masing activity pada perancangan use case. Simbol– simbol *Activity Diagram* dapat dilihat pada tabel 2.2.

Tabel 2. 2 *Simbol Activity Diagram*

Simbol	Keterangan
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Status awal 	Bagaimana objek dibentuk atau diawali.
Status akhir 	Bagaimana objek dibentuk dan diakhiri.

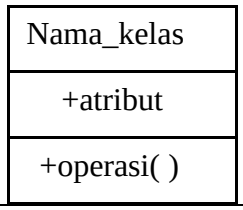
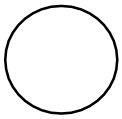
Percabangan / join 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

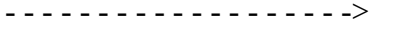
Tabel 2. 2 *Simbol Activity Diagram (* lanjutan *)*

2.6.3 *Class Diagram*

Gunawan & Sari (2017:319) menguraikan bahwa *Class diagram* menggambarkan atribut / poperti suatu system, sekaligus menawarkan layanan untuk memanipulasi keadaan metode atau fungsi. Class diagram menggambarkan struktur dan deskripsi class, package, dan objek beserta hubungan satu sama lain. Simbol – simbol yang ada pada *Class Diagram* ditunjukan oleh Tabel 2.3.

Tabel 2. 3 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur system.
Antarmuka / *interface*  **Nama_*interface***	Sama dengan konsep *interface* dalam pemograman berorientasi objek.
Asosisasi / *association*	Relasi antarkelas dengan makna umum, asosia biasanya disertai dengan *multiplicity*.

Asosisasi berarah / *directed association* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
generalisasi 	Relasi antarkelas dengan makna Generalisasi – spesialisasi (umum khusus).
Kebergantungan / *dependency* 	Relasi antarkelas dengan makna kebergantungan antarkelas.
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua –bagian

Tabel 2. 3 *Simbol Class Diagram (* lanjutan *)*

2.7 Penelitian Terdahulu

Berikut merupakan jurnal terkait dengan penelitian terdahulu :Berikut merupakan jurnal terkait dengan penelitian terdahulu :

Tabel 2. 4 Daftar Jurnal Terkait

No	Nama	Judul	Terbit/Tahun	Keterangan
1.	Harni Kusniyati ,Nicky Saputra p. Sitanggang	Aplikasi Edukasi Budaya Batak Toba Samosir Bebasis Android	Jurnal T. Informatika Vol. 9 No. 1, April 2016	Penelitian ini membahas mengenai Budaya batak Toba di Samosir.
2.	Radius Tanone, Sushendra Ipol.	Perancangan dan Implementasi Aplikasi Android Streaming (Studi Kasus FTI Universitas	Jurnal Teknik Informatika dan Sistem Informasi,	Penelitian ini membahas mengenai aplikasi android streaming dapat mengatasi masalah yang ada dengan memudahkan mahasiswa dan dosen untuk mengikuti

		Kristen Satya Wacana Salatiga)	Volume 1 Nomor 3 Desember 2015.	kegiatan secara realtime pada saat proses kegiatan sedang berlangsung tanpa harus datang langsung ke tempat pelaksanaan
3.	Eva Junita S.	Upacara kematian *Saurmatua* pada adat masyarakat batak tobadi desa Purbatua kecamatan Purbatua kabupaten Tapanuli Utara	JOM FISIP Vol. 3 No. 1 Februari 2016	Penelitian ini membahas mengenai adat kematian Saur Matua bagi adat batak Toba.

Tabel 2. 4 Daftar Jurnal Terkait (*lanjutan*)