BAB II

Tinjauan Pustaka

2.1 Lacak!

Lacak! Merupakan aplikasi berbasis Android yang memiliki tujuan untuk mengurangi tingkat kehilangan barang pribadi serta meningkatkan kemungkinan penemuan barang dengan memanfaatkan GPS *Tracker*. Aplikasi Lacak! bekerja dengan cara melacak keberadaan GPS Tracker yang disimpan pada barang-barang yang memiliki potensi kehilangan seperti mobil, motor, tas, dan yang lainnya. Fokus aplikasi Lacak! adalah untuk meminimalisir kehilangan maupun tertinggalnya barang pribadi serta jika pengguna perlu untuk mengetahui keberadaan benda mereka, aplikasi Lacak! dapat menampilkan informasi mengenai keberadaan benda mereka. Berikut merupakan fitur utama yang ditawarakan oleh aplikasi Lacak!:

1. Tracking Device

Fitur Tracking Device merupakan fitur yang dapat dimanfaatkan pengguna untuk mengetahui keberadaan benda dengan melacak keberadaan GPS Tracker yang sebelumnya sudah dihubungkan dengan aplikasi Lacak!.

2. Kunci Lokasi

Fitur Kunci Lokasi merupakan fitur yang akan memperingati pengguna jika benda mereka keluar dari radius yang telah ditentukan, fitur ini diharapkan dapat meminimalisir tingkat kehilangan.

3. Pemberitahuan

Fitur Pemberitahuan ini adalah fitur yang akan memberi pemberitahuan kepada pengguna jika berjarak lebih dari radius yang telah ditentukan sebelumnya sehingga diharapkan dapat meminimalisir tertinggalnya barang pengguna.

4. Berbagi Lokasi

Fitur berbagi lokasi ini adalah fitur yang memungkinkan pengguna aplikasi Lacak! memberi lokasi GPS *tracker* kepada orang lain. Diharapkan

dengan fitur berbagi lokasi pengguna dapat bergotong royong dengan orang lain untuk menemukan benda yang hilang.

5. Awas Hilang

Fitur awas hilang merupakan fitur yang bertujuan untuk mencegah terjadinya kehilangan dengan memanfaatkan informasi yang didapat dari twitter mengenai kehilangan.

Dengan fitur-fitur utama tersebut, aplikasi Lacak! merupakan aplikasi yang berfokus pada informasi keberadaan suatu benda dengan memanfaatkan GPS Tracker sehingga diharapkan dapat meminimalisir tingkat kehilangan dan juga tertinggalnya suatu barang.

2.2 Kehilangan

Kehilangan merupakan suatu situasi aktual maupun potensial yang dialami oleh individu ketika berpisah dengan sesuatu yang sebelumnya ada, bagi sebagian maupun keseluruhan. Dalam pandangan islam terdapat istilah *luqhatah* yaitu merupkaan harta yang ditemukan di suatu tempat dan tidak diketahui siapa pemiliknya. Kewajiban bagi orang yang menemukan untuk memberitahukan atau mengumumkan kepada masyarakat tentang penemuan barang tersebut [1]. kehilangan sering terjadi di masyarakat tingginya angka kriminalitas maupun ketidakwaspadaan masyarakat menjadikan fenomena kehilangan semakin marak terjadi di masyarakat. Kehilangan dapat terjadi akibat dari keteledoran pemilik maupun terjadinya peristiwa yang mengharuskan seseorang melepas barang maupun hal yang lainnya, sehingga terjadilah sebuah kehilangan.

2.3 GPS

GPS merupakan sebuah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Cara kerja sistem GPS adalah sebagai berikut :

- 1. Memakai perhitungan "triangulation" dari satelit.
- 2. Untuk perhitungan "triangulation", GPS mengukur jarak menggunakan travel time sinyal radio yang memerlukan akurasi waktu yang sangat tinggi.
- 3. Untuk perhitungan jarak, harus diketahui dengan pasti posisi satelit dan ketinggian pada orbitnya.

4. Terakhir adalah mengoreksi delay sinyal waktu perjalanan di atmosfer sampai diterima oleh *receiver*.

Sistem GPS menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) dimana alat ini akan mengumpulkan informasi dari satelit GPS. Sebuah GPS *receiver*mengunci sinyal minimal 3 satelit untuk menghitung posisi 2D (*Latitude*, *Longitude*) dan track pergerakannya. Jika GPS sudah mendapatkan posisi pengguna, maka GPS akan menghitung informasi yang lainnya seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, dan hal yang lainnya. Sinyal yang dikirimkan oleh satelit ke GPS digunakan untuk menghitung waktu perjalanan (*Travel Time*). Waktu perjalanan ini sering juga disebut sebagai *Time of Arrival* (TOA) [9].

2.4 Mobile GIS

Mobile GIS merupakan sebuah kerangka teknologi terintegrasi yang dapat mengakses data spasial dan location based services merlalui perangkat mobile seperti smartphone, Personal Digital Assitance (PDA) maupun yang lainnya. Mobile GIS merupakan sebuah integrasi cara kerja perangkat lunak/keras untuk pengaksesan data dan layanan geospasial melalui perangkat bergerak via jaringan kabel maupun jaringan nirkabel. Secara umum mobile GIS diimplementasikan pada dua area aplikasi utama yaitu Location-Based Service (LBS) dan GIS untuk kegiatan lapangan (Field-Based GIS) [10].

2.4.1 Location-Based Service

Location-Based Service atau layanan berbasis lokasi merupakan layanan informasi yang dapat diakses melalui perangkat mobile memanfaatkan jaringan selular. Location-Based Service memiliki kemampuan untuk memanfaatkan lokasi posisi perangkat mobile [10].

2.5 Super Spring M20

Super Spring M20 merupakan sebuah GPS Tracker Portable yang memanfaatkan GPS untuk mendapatkan koordinat perangkat ini. Super Spring M20 memiliki tingkat akurasi < 10 Meter sehingga untuk melacak keberadaan suatu barang Super Spring M20 ini sangatlah baik, terlebih lagi ukurannya yang hanya 80x50x34 mm menjadikan Super Spring M20 ini dapat dibawa-bawa dan

diletakkan pada benda yang akan dilacak keberadaannya. Pengguna Super Spring M20 dapat melihat keberadaan perangkat ini dengan melakukan install aplikasi GPS.id pada Play Store maupun AppStore. Perangkat Super Spring M20 dapat dilihat pada gambar 2.1.



Sumber : Google Image

Gambar Error! No text of specified style in document..1 Perangkat Super Spring M20

2.6 Twitter

Twitter merupakan sebuah media sosial dan layanan *microblogging* yang memungkinkan untuk penggunanya untuk mengirim pesan seacara *realtime*. Pesan yang dikirim ini lebih dikenal dengan sebutan *tweet*. Tweet merupakan sebuah pesan singkat yang panjang pesannya dibatasi hanya 140 karakter. Twitter diciptakan untuk menjadi wadah saling berbagi pengalaman tanpa adanya sekat penghalang. Twitter membantu penggunanya untuk selalu terhubung dengan orangorang terdekatnya. Berikut merupakan beberapa istilah yang ada pada media sosial twitter:

- 1. *Mention*, merupakan istilah untuk memanggil pengguna twitter lain dalam sebuah tweet. Mention dilakukan dengan menuliskan '@' diikuti dengan nama pengguna lain.
- 2. *Hashtag*, merupakan istilah untuk menandai sebuah topik pembicaraan di twitter. Penulisan *hashtag* dimulai dengan tanga '#' diikuti dengan topik yang akan dibahas. *Hashtag* dapat dimanfaatkan untuk *campaign* dengan tujuan meningkatkan visibilitas tweet kepada orang-orang yang tidak mengikuti ataupun diikuti.

- 3. *Emoticon*, merupakan istilah untuk ekspresi wajah yang direpresentasikan dengan kombinasi huruf, tanda baca dan angka. *Emoticon* digunakan untuk mengekspresikan perasaan.
- 4. *Trending topics*, merupakan istilah untuk kumpulan dari topik pembicaraan yang sangat populer di twitter [4].

2.7 Aplikasi

Aplikasi merupakan sebuah program yang siap untuk digunakan oleh pengguna. Program tersebut harus dapat berfungsi sebagaimana mestinya untuk membantu suatu pekerjaan tepat pada sasaran. Aplikasi biasanya berpacu pada suatu komputasi untuk memproses suatu data sehingga data tersebut dapat berubah menjadi sesuatu yang diharapkan. Aplikasi dibangun menggunakan bahasa pemrograman tertentu [5].

2.8 Internet

Internet merupakan singkatan dari *interconnection networking* yang secara sederhakan dapat diartikan sebagai *global network of computer networks*. Pada era siber internet memiliki akselerasi *hypereksponensial* yang sulit untuk dibayangkan. Sehingga dampak dari hal itu adalah terbentuknya sebuah komunitas yang memanfaatkan internet secara maksimum untuk kepentingan hidupnya salah satunya adalah untuk kepentingan berkomunikasi tanpa batas ras, bagnsa, geografi, kelas, dan batasan lainnya [11]. Dengan adanya internet seluruh komunikasi dapat dilakukan tanpa batasan waktu.

2.9 Android

Android merupakan sistem operasi *mobile* berbasis linux yang dirancang untuk perangkat yang memiliki kemampuan layar sentuh seperti *smartphone* maupun *tablet*. Android dibeli Google pada tahun 2005 dan pada tahun 2008 tepatnya pada tanggal 22 Oktober 2008 diluncurkan *smartphone* berbasis Android pertama dengan nama HTC Dream. Dikarenakan Android bersifat *open source* perkembangannya sangat pesat sekali. Sekarang sistem operasi Android tidak hanya dapat berjalan pada *smartphone* maupun *tablet* saja tetapi dapat berjalan pada perangkat seperti laptop, TV maupun *smart watch* [12].

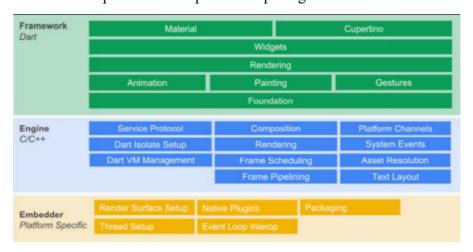
Dengan banyaknya pengguna dan perangkat yang mendukung berjalannya sistem operasi Android, berbanding lurus dengan kebutuhan aplikasi yang diperlukan oleh pengguna. Android merupakan salah satu *mobile operating system* yang saat ini banyak digunakan di berbagai perangkat pintar. Dikarenakan saat ini Android merupakan sistem operasi yang banyak tersebar di masyarakat dan digemari oleh masyarakat, maka banyaknya aplikasi-aplikasi berbasis Android tidaklah aneh lagi [12]. Hingga saat ini Android telah banyak merilis versi-versi sistem operasinya, berikut merupakan versi-versi Android yang dapat dilihat pada tabel 2.1:

Tabel Error! No text of specified style in document..**1 Versi-Versi Android**Sumber: Buku Xamarin Android (2017) [12]

Versi	Nama Kode	Tanggal Rilis	Level API
1.5	Cupcake	30 April 2009	3
1.6	Donut	15 September 2009	4
2.0-2.1	Eclair	26 Oktober 2009	7
2.2	Froyo	20 Mei 2010	8
2.3-2.3.2	Gingerbread	6 Desember 2010	9
2.3.3-2.3.7	Gingerbread	9 Februari 2011	10
3.1	Honeycomb	10 Mei 2011	12
3.2	Honeycomb	15 Juli 2011	13
4.0.3-4.0.4	Ice Cream Sandwich	16 Desember 2011	15
4.1.x	Jelly Bean	9 Juli 2012	16
4.2.x	Jelly Bean	13 November 2012	17
4.3.x	Jelly Bean	24 Juli 2013	18
4.4.x	KitKat	31 Oktober 2013	19
5.x	Lollipop	15 Oktober 2014	21
6.0	Marshmallow	19 Agustus 2015	23
7.0	Nougat	22 Agustus 2016	24

2.10 Flutter

Flutter merupakan sebuah Mobile App SDK milik Google dimana *developer* dapat mengembangkan aplikasi iOS dan juga Android dengan source code yang sama, atau biasa disebut dengan *Single Code Base*. Bahasa yang digunakan oleh Flutter adalah Dart. Flutter memungkinkan *developer* membangun aplikasi secara native dengan menggunakan bahasa pemrograman Dart. Didukung oleh Google sepenuhnya Flutter berkembang dengan sangat cepat sehingga dapat bersaing dengan framework mobile lainnya seperti Xamarin, React Native dan lainnya [13]. Struktur framework pada flutter dapat dilihat pada gambar 2.2 berikut:



Sumber: Journal of Information System, Graphic, Hospitality, and Technology (2020) [13]

Gambar Error! No text of specified style in document..2 Struktur Framework Flutter

Kelebihan dari flutter dibandingkan framework lainnya dapat dilihat pada poin-poin berikut:

1. Produktivitas Tinggi

Flutter merupakan *Cross-Platform* dengan 1 kode Flutter dapat menciptakan aplikasi iOS dan Android. Dengan ini tentunya akan sangat menghemat waktu dan tenaga.

2. Performa Hebat

Dart dapat mengkompilasi kedalam kode *native* sehingga tidak perlu lagi mengakses OEM widget dikarenakan Flutter mempunya widget nya sendiri.

Sehingga Flutter dapat berjalan dengan cepat dan juga mengurangi masalah performa.

3. Development yang Ringkas dan Cepat

Salah satu fitur Flutter yaitu *hot reload* dimana fitur ini berguna untuk menampilkan perubahan yang dilakukan pada kode secara instan pada emulator, simulator dan hardware. Dengan hanya beberapa detik kode yang dirubah dapat dimuat ulang dan aplikasi masih berjalan tanpa perlu melakukan *restart*.

4. Open Source

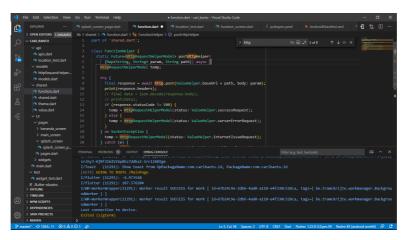
Flutter dan Dart, keduanya *open source* serta gratis untuk digunakan. Keduanya memberikan dokumentasi yang bagus dan lengkap serta ada komunitas yang membantu untuk menyelesaikan masalah yang ditemukan saat proses *development*.

2.11 Visual Studio Code

Visual Studio Code atau yang biasanya disingkat VSCode merupakan sebuah aplikasi code editor yang tidak hanya tersedia untuk sistem operasi Windows tetapi tersedia juga untuk Linux dan Mac OS. Layaknya sebuah code editor lainnya seperti Sublime maupun Atom, Visual Studio Code mendukung berbagai jenis bahasa pemrograman. Mulai dari JavaScript, Java, PHP, C++, dan lainnya. Aplikasi VSCode bahkan dapat mengidentifikasi secara otomatis bahasa pemrograman yang digunakan dan memberikan variasi warna sesuai dengan fungsi dalam rangka kode tersebut. VSCode memiliki ekstensi yang dapat berfungsi untuk mengintegrasikan workspace kita dengan GitHub. Selain itu berikut kelebihan dari VSCode:

- 1. Menyediakan dukungan terhadap bahasa pemrograman maupun framework
- 2. Aplikasinya tergolong ringan.
- 3. Memiliki UI/UX yang simple
- 4. Dapat terintegrasi dengan Git.
- 5. Memiliki banyak ekstensi

Berikut ini merupakan tampilan pada aplikasi Visual Studio Code, yang dapat dilihat pada gambar 2.3



Gambar Error! No text of specified style in document..3 Visual Studio Code
2.12 OOP

OOP (Object Oriented Programming) merupakan sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi maupun berbasis pada sebuah obyek dalam pembangunan program aplikasi. Pada pembuatan program OOP sudah tidak berorientasi linear melainkan berorientasi pada obyek-obyek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah obyek yang didalamnya berisi beberapa perintah-perintah standar sederhana [14]. Dengan orientasi OOP pada sebuah obyek dalam pembangunan aplikasi berikut merupakan kemampuan yang dapat dilakukan oleh pemrograman berbasis obyek ini:

1. Class

Class merupakan gambaran dari sebuah obyek. Class dapat didefinisikan dengan penampung nilai property dan method yang dimana methode merupakan tingkah laku yang dapat dilakukan oleh sebuah obyek.

2. Obyek

Obyek merupakan hasil inisiasi dari sebuah *class*, mengandung *resource* yang telah didefinisikan pada *class*. Dengan kata lain obyek dapat didefinisikan sebagai hasil cetakan dari *class*.

3. Encapsulation

Encapsulation merupakan mekanisme untuk "membungkus" sebuah property maupun method. Dalam istilah lain seringkali disebut dengan "Information Hiding". Terdapat 3 jenis "pembungkusan" yang dapat

dilakukan yaitu *private, protected,* dan *public*. Jenis "pembungkusan" tersebut dapat dikatakan sebagai *modifier* yang digunakan untuk mendefinisikan tingkat visibilitas sebuah *property* maupun *method* di dalam sebuah *class*.

4. Polymorphism

Polymorphism membuat obyek-obyek yang berasal dari subclass yang berbeda. Hal ini terjadi ketika memilih method yang sesuai untuk diimplementasikan ke obyek tertentu berdasarkan pada subclass yang memiliki method bersangkutan.

Kondisi yang harus dipenuhi untuk dapat mengimplementasikan polymorphism adalah :

- a. Method yang dipanggil harus melalui variable dari basis class atau superclass
- b. Method yang dipanggil harus juga menjadi method dari basis class.
- c. Parameter method harus sama baik pada superclass maupun subclass.
- d. Access modifier attribute pada subclass tidak boleh lebih terbatas dari basic class.

5. Constructor dan Destructor

OOP memungkinkan untuk menyatakan *method* ketika sebuah obyek diinisiasikan yang otomatis akan terpanggil ketika obyek dibuat yang disebut dengan *constructor*. Begitu pula dengan *destructor* yang otomatis akan panggil ketika tidak ada referensi lain.

6. Inheritance

Inheritance merupakan cara untuk menggunakan kembali sebuah *class* atau untuk mendirikan *subtype* dari obyek yang sudah ada. *Inheritance* memungkinkan untuk *sublass* menggunakan *method* maupun *attribute* yang sudah didefinisikan oleh *class* dasarnya [14].

2.13 MVVM

MVVM (*Model-View-ViewModel*) merupakan variasi dari *Model-View-Controller* (MVC) yang dirancang sebagai *platform* pengembangan antarmuka pengguna modern. MVVM merupakan turunan dari model MVC. *Controller* pada

model MVC digantikan dengan *ViewModel* yang tugasnya adalah mengontrol penampilan dari *View*. MVVM memiliki kemampuan baru dalam *data-binding* yang terdapat pada modul *ViewModel* [15].

2.14 Database

Database merupakan kumpulan *file-file* yang saling berkaitan dan berinteraksi satu sama lain, membentuk suatu relasi dan saling terhubung satu sama lain. Sebuah Database menunjukan suatu kumpulan data yang dipakai dalam suatu ruang lingkup perusahaan maupun instansi. Pengelohan Database merupakan suatu cara yang dilakukan yang mana *file-file* tersebut disusun,diurut,diambil maupun ditampilkan dalam bentuk laporan sehingga *file-file* tersebut rapih dan dapat di *index* dengan baik [16].

2.15 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan untuk mengembangkan aplikasi web untuk klien. TcX sendiri merupakansebuah perusahaan pengembang *software* dan konsultan Database. Saat ini MySQL sudah diakuisisi oleh Oracle Corp. MySQL sendiri merupakan salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web dan membutuhkan database untuk pengelolaan datanya. MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses database-nya sehingga mudah untuk digunakan. MySQL juga bersifat *open source* dan gratis. MySQL didistribusikan dengan lisensi *open source* GPL (*General Public Lisence*) [17].

2.16 API

API (Application Programming Interface) merupakan sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi maupun sebuah bahasa pemrograman tertentu yang dapat dignakan oleh programmer saat membangun sebuah perangkat lunak. API memungkinkan para programmer yang sedang mengembangkan suatu perangkat lunak menggunakan fungsi yang sebelunya telah dibuat. Dalam API terdapat fungsi-fungsi atau perintah-perintah

untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah di mengerti oleh *programmer* [18]. Berikut merupakan keuntungan dalam menggunakan API:

- 1. API dapat digunakan utnuk semua bahasa pemrograman atau OS asalkan paket-paket API sudah terpasang.
- 2. API menggunakan bahasa yang lebih terstruktur dibandingkan harus menggunakan *system call*.
- 3. API memudahkan *programmer* dalam mengembangkan sebuah perangkat lunak.

2.17 Firebase API

Firebase adalah sebuah API yang disediakan oleh Goole, untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, ios, maupun website. Salah satu fitur yang sangat populer adalah hadirnya realtimedatabase merupakan fasilitas yang penyimpanan dan pengambilan data dari database sangat cepat. Selain memiliki fitur realtimedatabase Firebase memiliki fitur-fitur seperti authentication, database, storage, hosting, push notification dan lain-lain [19].

Firebase API menawarkan berbagai macam fitur yang dapat digunakan secara gratis. Penggunaan fitur Firebase API secara gratis adalah untuk memberikan dukungan terhadap pengembang-pengembang di seluruh dunia untuk menggunakan API ini untuk bersama-sama belajar dan mengembangkan teknologi. Firebase API dapat digunakan secara gratis dengan batas pemakaian wajar. Jika melebihi pemakaian tersebut maka pengguna API ini harus membayar. Produk yang disediakan oleh Firebase API sangatlah beragam, tidak hanya untuk menyimpan data saja. Berikut merupakan beberapa produk yang ditawarkan oleh Firebase API:

1. Cloud Firestore

Firebase API menyediakan sebuah *database* yang menggunakan NoSQL yang di-*hosting* di *cloud*. Sehingga seluruh *database* disimpan berbasis *cloud* tanpa memerlukan *server* tambahan.

2. ML Kit

ML Kit merupakan produk yang ditawarkan oleh Firebase API yang memungkinkan pengembang suatu aplikasi menggunakan Machine Learning dalam mengembangkan aplikasinya.

3. Authentication

Firebase API memberikan fitur untuk melakukan *register* maupun *login* dengan beberapa metode seperti : Akun Google, Akun Facebook, Akun Twitter, Termasuk akun *Anonymous*. Penggunaan akun *anonymous* memungkinkan untuk pengguna tidak diharuskan melakukan *register* maupun *login* terlebih dahulu untuk menggunakan suatu layanan pada aplikasi yang menggunakan fitur ini.

4. Realtime Database

Realtime Database memungkinkan pengguna untuk mendapatkan update secara realtime jika terdapat perubahan data pada database. Selain itu Firebase API memungkinkan untuk aplikasi yang menggunakan fitur ini untuk menyimpan data secara lokal ketika perangkat tidak terhubung dengan internet. Setelah perangkat terhubung dengan internet maka data yang sebelumnya telah disimpan secara lokal dapat segera diubah pada database.

5. Crashlytics

Crashlytics merupakan fitur yang memungkinkan untuk pencatatan *error* yang terjadi secara terus menerus paa aplikasi yang menggunakan fitur ini. Sehingga hasil analisis dapat dilihat secara jelas dan dapat ditangani. Firebase API menyediakan sebuah *dashboard* untuk memantau aplikasi.

6. Firebase Cloud Messaging

Firebase Cloud Messaging merupakan solusi perpesanan antar platform tanpa biaya serta tidak memiliki kuota. FCM memanfaatkan jaringan seluler ataupun wi-fi untuk memberikan solusi push notification sehingga developer dapat langsung mengirimkan notifikasi ke aplikasi dengan mudah [20].

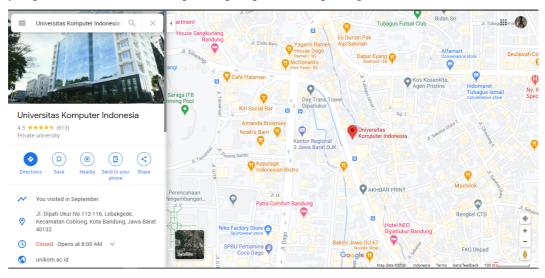
Dengan berbagai fitur menarik yang disediakan oleh Firebase API tidak heran banyak yang menggunakan Firebase API untuk mengembangkan aplikasinya.

Perusahaan sebesar Alibaba.com pun menggunakan Firebase API untuk menjalankan layanan mereka. Firebase API dapat digunakan dengan mudah karena Google telah menyediakan *tutorial* untuk menggunakan produk mereka secara lengkap dan mudah dipahami.

2.18 Google Maps API

Google Maps merupakan sebuah jasa peta globe virtual gratis dan online yang disediakan oleh Google sejak tahun 2005 dan terus berkembang hingga saat ini. Selain mendapatkan tampilan peta dunia Google Maps juga dapat memberikan informasi pendukung berupa informasi jalan, lokasi layanan publik, bisnis dan sebagainya. Selain itu Google Maps menawarkan layanan API yang memungkinkan pada *developer* untuk mengintegrasikan Google Maps kedalam sistem mereka [11].

Google Maps API merupakan sebuah layanan (*Services*) yang diberikan oleh Google kepada para pengguna untuk memanfaatkan Google Map dalam mengembangkan aplikasi mereka. Google Maps API menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki. Contoh Google Map dapat dilihat pada gambar 2.4 berikut.



Sumber: Google Maps

Gambar Error! No text of specified style in document..4 Tampilan Google

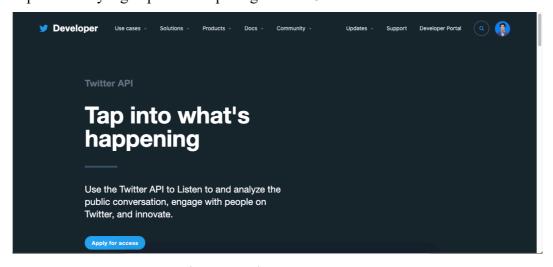
Maps

2.19 Super Spring API

Super Spring merupakan penyedia perangkat GPS baik GPS Navigasi maupun GPS *Tracker* yang sudah berdiri sejak 2008. Pada penelitian ini penggunaan Super Spring API adalah untuk mendapatkan data-data dari *device* GPS *tracker* sehingga dapat dimanipulasi di *server* untuk menyelesaikan permasalahan yang diangkat dalam penelitian ini. Super Spring API saat ini dapat digunakan untuk login, listing device serta beberapa fungsi untuk mengambil data dari sebuah *device* GPS.

2.20 Twitter API

Twitter API merupakan sebuah program atau aplikasi yang disediakan oleh twitter untuk mempermudah developer dalam mengakses data-data yang ada pada twitter. Pendaftaran serta dokumentasi twitter API dapat diakses pada website https://dev.twitter.com setelah mendaftar developer akan menadpatkan consumer key, consumer access, access token dan juga access token secret yang nantinya dapat digunakan untuk mengakses data-data twitter [5]. Pada penelitian ini penggunaan twitter API adalah untuk mendapatkan dapat mengenai kehilangan dengan memanfaatkan hashtag untuk mencari topik pembicaraan kehilangan. Data yang diperoleh dari twitter nantinya akan diolah di server untuk memperingati pengguna jika pada sebuah daerah sering terjadi kehilangan, sehingga diharapkan dapat meminimalisir terjadinya kehilngan. Berikut merupakan tampilan halaman developer twitter yang dapat dilihat pada gambar 2.5.



Sumber: Developer Twitter

Gambar Error! No text of specified style in document..5 Tampilan Google Maps

2.21 JSON

JSON (*JavaScript Object Notation*) merupakan sebuah format dengan tujuan untuk memasukan data ke dalam sebuah variabel. JSON memiliki format yang sederhana sehingga mudah dimenegerti dan diimplementasikan oleh manusia, dan mudah juga untuk komputer dalam melakukan parsing. JSON merupakan format teks yang sepenuhnya independent [21].

JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999). Dengan kemampuan untuk konversi bahasa pemrograman dari keluarga-C seperti, C, C++, C#, Java, JavaScript, Perl, Phyton, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa *data-intercharge* yang ideal. JSON dibagun dalam dua struktur:

- 1. Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah *object, record, struct,* tabel *hash, key list* atau *associative array*.
- 2. Nilai-nilai yang tersusun secara *ordered list*. Biasa disebut dengan *array*, *vector list* atau daftar dalam bahasa pemrograman.

JSON merupakan data yang *universal*, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung JSON dalam berbagai format. Hal ini memungkinkan untuk melakukan pertukaran data antar bahasa pemrograman yang juga mendukung struktur JSON [21]. Contoh format JSON dapat dilihat pada tabel 2.2.

Tabel Error! No text of specified style in document..**2 Contoh Format JSON**Sumber: Jurnal Merpati (Menara Penelitian Akademika Teknologi Informasi
(2014) [21]

```
{"status":"Ok",

"item":[

{"id":"1","bulan":"Januari","total":"700"},

{"id":"2","bulan":"Februari","total":"400"},

{"id":"3","bulan":"Maret","total":"200"},
```

```
{"id":"4","bulan":"April","total":"800"},

{"id": "5","bulan":"Mei","total":"230"},

{"id":"6","bulan ":"Juni","total":"400"},

{"id":"8","bulan":"Agustus","total":"350"},

{"id":"9","bulan":"September","total":"800"},

{"id":"10","bulan":"Oktober","total":"350"},

{"id ":"11","bulan":"November","total":"400"},

{"id":" 12","bulan":"Desember","total":"600"}

]}
```

2.22 PHP

PHP merupakan singkatan dari (*Hypertext Preprocessor*) yang merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang ditulis dijalankan sepenuhnya oleh *server* tetapi disertakan pada halaman HTML biasa [17].

2.23 UI

UI (*User Interface*) adalah penghubung antara manusia sebagai pengguna (*User*) dengan *software* pada perangkat (*interface*). Desain *User Interface* pada perangkat misalnya *smartphone*, tablet atau berbagai perangkat lainnya harus baik sehingga mudah digunakan oleh penggunanya [22]. Pengimplementasian *user interface* yang baik ditujukan utuk agar pengguna dapat menggunakan aplikasi lebih lama dan nyaman.

2.24 UX

UX (*User Experience*) adalah pengalaman (*Experience*) yang dialami oleh pengguna (*User*) pada saat mengakses suatu sistem. Sistem tersebut bisa berupa website, aplikasi smartphone, atau sebagainya. *User Experience* yang baik akan memudahkan pengguna dalam pengoperasian aplikasi [22]. Sehingga pengguna dapat secara efektif dan efisien dalam menggunakan aplikasi.

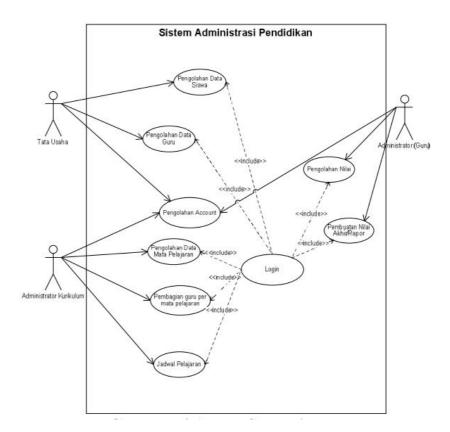
2.25 UML

UML (*Unified Model Language*) merupakan himpunan struktur dan teknik untuk pemodelan dan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupkan dasar bagi tool desain berorientasi objek dari IBM [23].

2.25.1 Use Case

Use-Case (kasus penggunaan) adalah teknik berdasarkan skenario untuk elisitasi persyaratan yang pertama kali diperkenalkan pada metode Objectory (Jacobson et al., 1993). Use-case sekarang telah menjadi fitur dasar notasi UML untuk mendeskripsikan model sistem berorientasi objek. Dalam bentuknya yang paling sederahana, use-case mengidentifikasi aktor yang terlibat dalam interaksi dan nama tipe interaksi tersebut [24].

Use case merupakan sebuah diagram yang menjelaskan interaksi sebuah sistem dengan aktor yang terlibat dengan sistem yang nantinya akan dibuat. Use case secara eksplisit menjelaskan fungsi-fungsi yang nantinya akan digunakan oleh aktor (Pengguna). Sehingga pembuatan Use Case harus diperhatikan karena Use Case yang nantinya akan menjadi acuan dalam pembuatan dan pengembagan perangkat lunak nantinya [25]. Berikut merupakan contoh Use Case yang dapat dilihat pada gambar 2.5.



Sumber: Jurnal Fakultas Sains & Teknologi UIN Sumatra Utara Medan (2016) [26]

Gambar Error! No text of specified style in document..6 Contoh Usecase

Use Case Description, merupakan tabel yang digunakan untuk membuat dan menjelaskan keterangan terperinci mengenai tiap-tiap Use Case [27]. Use Case Description merupakan tabel yang akan menjelaskan bagaimana suatu fungsi yang terhubung dengan aktor berjalan. Use Case Description menjadi suatu acuan dalam pembuatan fungsi pada perangkat lunak sehingga nantinya perangkat lunak yang dihasilkan sesuai dengan apa yang diharapkan. Contoh Use Case Description dapat dilihat pada tabel 2.3 berikut.

Tabel Error! No text of specified style in document..3 Contoh Use Case

Description

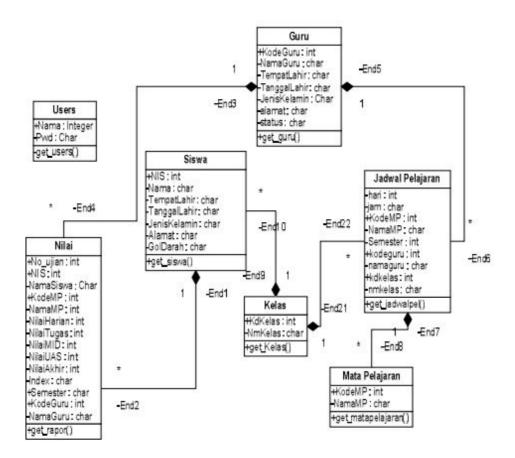
Use Case	Melihat info biodata
Name:	
Scenario:	Melihat info biodata melalui ponsel J2ME

Triggering	Mahasiswa menekan tombol biodata di halaman utama		
Event:			
Brief	Ketika mahasiswa menekan tombol biodata di halaman utama,		
Description:	sistem menghubungi webserver SION, mem-parsing dan		
	menampikan halaman biodata		
Actors:	Mahasiswa		
Related Use	-		
Cases:			
Stakeholders	-		
:			
Preconditions	Data login (NIM dan Password) di RMS ponsel tidak boleh		
:	kosong		
Flow of	Actor	System	
Activities:			
	1. Mahasiswa Menekan	1. –	
	Tombol Biodata di halaman	2. Sistem menampilkan animasi	
	utama	loading sebagai tanda	
		bahwa proses sedang	
		berjalan	
		3. sistem mengambil data NIM,	
		password, dan pilihan	
		format (XML atau JSON)	
		di RMS ponsel	
		4. Sistem membuka koneksi	
		НТТР	
		5. Sistem meminta halaman	
		biodata ke webserver SION	
		sesuai dengan NIM,	
		Password, dan pilihan	

		format yang didapat dari	
		langkah 3	
		6. Sistem mem-parsing	
		halaman biodata yang	
		masih berupa format XML	
	atau JSON		
	7. Sistem menampilkan		
		informasi biodata yang	
		sudah di- <i>parsing</i>	
		8. Sistem menghilangkan	
		animasi loading	
		9. Sistem menutup koneksi	
		НТТР	
Exception	5.1 Jika sistem tidak mampu menghubungi webserver SION,		
conditions:	sistem akan menampilkan pesan koneksi error dan		
	langsung menuju langkah 8		
	6.1 Jika sistem tidak mampu mem-parsing halaman biodata,		
	sistem menampilkan pesan error dan langsung menuju		
	langkah 8		

2.25.2 Class Diagram

Class Diagram terdiri atas kelas-kelas yang memiliki nama, *field-field* di dalam kelas, dan tindakan-tindakan (kadang disebut sebagai metode) yang dilakukan atas kelas [28]. Class Diagram merupakan diagram yang menggambarkan class yang terlibat didalam sistem. Class diagram dapat saling berelasi satu sama lain. Contoh dari Class Diagram dapat dilihat pada gambar 2.6.



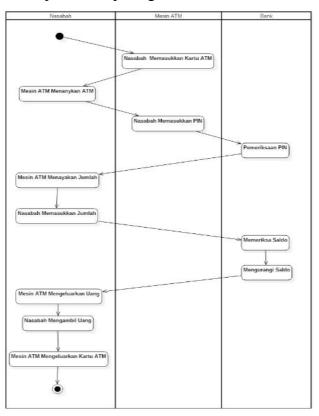
Sumber : Jurnal Fakultas Sains & Teknologi UIN Sumatra Utara Medan (2016) [26]

Gambar Error! No text of specified style in document..**7 Contoh** *Class Diagram* Contoh pada gambar 2.6 terdapat keterangan berupa enkapsulasi terhadap *variable* dan metoda. Terdapat 3 jenis enkapsulasi yaitu :

- 1. Private, tidak dapat dipanggil dari luar *class* yang bersangkutan yang dilambangkan dengan notasi (-).
- 2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya yang dilambangkan dengan notasi (#).
- 3. *Public*, dapat dipanggil oleh siapa saja yang dilambangkan dengan notasi (+) [29].

2.25.3 Activity Diagram

Activity diagram adalah diagram UML yang digunakan untuk menggambarkan alur aktivitas dari satu proses. Misalnya proses memesan makan, aktivitasnya adalah menelpon restoran, memesan makan dan memberikan alamat kita, kemudian tunggu pesanan datang dan bayar. Activity diagram memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya, dengan kata lain diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus kita ikuti. Contoh activity diagram dapat dilihat pada gambar 2.7.



Sumber: Rekayasa Perangkat Lunak Menggunakan UML dan Java (2009)
[25]

Gambar Error! No text of specified style in document..8 Contoh Activity

Diagram

2.25.4 Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan interaksi antar objek. Sequence diagram secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang

interaction Tarik Saldo ATM Nasabah Mesin ATM Jaringan Bank Bank 1 : Masukkan Kartu Permintaan Password 3 · Masukkan Passy 6: Rekening Bank Ok 7 : Rekening Ok Permintaan Jumlal 10 : Masukkan Jumlah 12: Memproses Transaksi 13 : Transaksi Berhasi 15 : Mengeluarkan Uang Tuna 14 : Transaksi Berhasi 16 : Ambil Uang Tuna 17 : Pertanyaan Lanjut Transak Mengeluarkan Kartu ATM 21 : Ambil Kartu Tampilkan Layar Awa

melewati objek ini dalam sebuah use case [30]. Sequence diagram merupakan diagram yang menjelaskan komunikasi data maupun pesan antar obyek.

Sumber: Rekayasa Perangkat Lunak Menggunakan UML dan Java (2009)
[31]

Gambar Error! No text of specified style in document..9 Contoh Sequence

Diagram

2.26 Pengujian

Pengujian *software* adalah proses untuk memastikan apakah semua fungsi sistem bekerja dengan baik, dan mencari apakah masih ada kesalahan pada sistem. Pengujian atau *testing software* sangat penting untuk dilakukan. Pengujian ini bertujuan untuk menjamin kualitas *software*, dan juga menjadi peninjauan terakhir terhadap spesifikasi, desain dan pengkodean [31].

2.26.1 Black Box Testing

Pendekatan black box *testing*, pendekatan ini melakukan pengujian terhadap fungsi operasional *software*. Pendekatan ini biasanya dilakukan oleh penguji yang tidak ikut serta dalam pengkodean *software*.

- 1. *Black Box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- 2. *Black Box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup error dengan kelas yang berbeda dari metode *white box testing*
- 3. Black box testing melakukan pengujian tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites juga disebut sebagai behavioral testing, specification-based testing, input/output testing atau functional testing.
- 4. Pada *black box testing* terdapat jenis teknik desain tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan diantaranya : equivalence class partitioning, boundary value analysis, state transitions testing, cause-effect graphing.

Kategori *error* yang akan diketahui melalui *black box testing* diantaranya adalah: Fungsi yang hilang atau tak benar, *Error* dari antar-muka, *Error* dari struktur data atau akses eksternal *database*, *Error* dari kinerja atau tingkah laku, *Error* dari inisialisasi dan terminasi Pendekatan black box *testing*, pendekatan ini melakukan pengujian terhadap fungsi operasional *software*. Pendekatan ini biasanya dilakukan oleh penguji yang tidak ikut serta dalam pengkodean *software*.

- 5. *Black Box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- 6. *Black Box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup error dengan kelas yang berbeda dari metode *white box testing*
- 7. *Black box testing* melakukan pengujian tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites juga disebut sebagai

- behavioral testing, specification-based testing, input/output testing atau functional testing.
- 8. Pada *black box testing* terdapat jenis teknik desain tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan diantaranya : equivalence class partitioning, boundary value analysis, state transitions testing, cause-effect graphing.

Kategori *error* yang akan diketahui melalui *black box testing* diantaranya adalah: Fungsi yang hilang atau tak benar, *Error* dari antar-muka, *Error* dari struktur data atau akses eksternal *database*, *Error* dari kinerja atau tingkah laku, *Error* dari inisialisasi dan terminasi [31].