

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Beberapa acuan yang digunakan dalam pengembangan sistem dan aplikasi ini ditampilkan pada tabel 2.1 di bawah ini :

**Tabel 2.1** Perbedaan dan Tinjauan Pustaka

No	Penulis	Objek	Teknologi	Pelacakan <i>Real-time</i>	Bahasa Pemrograman
1	Moh., Mustofa Ali (2015)	Pencarian Lokasi Hotel	<i>Webservice</i>	Tidak	Java, PHP
2	Nova Agustina dan Slamet Risnanto (2016)	Pencarian Lokasi Wisata Kota Cimahi	<i>Webservice</i>	Tidak	Jquery, Javascript, Node.JS
2	Buru, Dadang Juwoto (2017)	Perangkat Bergerak	<i>Firebase</i>	Ya	Java, JSON
3	Fadillah (2017)	Pencarian Lokasi Kendaraan Antar Jemput	<i>Webservice</i>	Ya	Java, PHP
4	Wicaksono (2017)	Pencarian Lokasi Apotek	<i>Webservice</i>	Tidak	Java, PHP
5	Hafiz Dharmawan (2018)	Pencarian Petugas PMI & Pendonor Darah	<i>Firebase</i>	Ya	Java, JSON

Pada tahun 2015 pernah dilakukan penelitian dengan judul “Pencarian Lokasi Hotel Berbasis *Android* Menggunakan Metode *Location Based Service* (Studi Kasus Kota Semarang)” yang dibuat oleh Mustofa Ali Moh dari Universitas Dian Nuswantoro Semarang. Aplikasi tersebut dapat melakukan pencarian lokasi hotel terdekat di kota Semarang. Teknologi yang digunakan yaitu web servis untuk pengambilan dan pengiriman data.

Pada tahun 2016 pernah dilakukan penelitian dengan judul “Aplikasi *Location Based Service* Untuk Informasi Dan Pencarian Lokasi Pariwisata Di Kota Cimahi Berbasis *Android*” yang dibuat oleh Nova Agustina dan Slamet Risnanto dari Teknik Informatika Sekolah Tinggi Teknologi Bandung. Aplikasi tersebut dapat melakukan pencarian lokasi wisata yang berada di kota Cimahi dan menentukan rute jalan terdekat. Teknologi yang digunakan yaitu *Node.js* sebagai *backend* dan menggunakan *framework phonegap* untuk membangun *frontend* berbasis *Android*.

Pada tahun 2017 dilakukan penelitian dengan judul “Aplikasi Pendeteksi Lokasi Perangkat Bergerak Menggunakan Teknologi *Cloud Computing* Dengan *Firebase Relatime Database* Berbasis *Android*” sebuah aplikasi real-time berbasis android untuk pelacakan perangkat bergerak” oleh Dadang Juwoto dari STMIK Akakom Yogyakarta. Aplikasi ini melakukan pelacakan lokasi pada perangkat bergerak secara real-time untuk dibagikan kepada pengguna yang dapat menolong

pengguna. Aplikasi ini menggunakan layanan *Firebase* untuk kebutuhan berbagi lokasi secara *realtime*.

Pada tahun 2017 dilakukan penelitian dengan judul “Aplikasi Sistem Pelacakan Lokasi Kendaraan Antar Jemput Anak Sekolah Berbasis *Android*” oleh Fadillah dari Universitas Bandar Lampung. Aplikasi ini dapat melacak kendaraan pada saat menjemput anak sekolah secara *realtime*. Teknologi yang digunakan adalah web servis REST API dan menggunakan *looping* pada *Android* untuk mengirimkan lokasi *latitude* dan *longitude* secara berkala.

Pada tahun 2017 dilakukan penelitian dengan judul “Aplikasi Pencarian Lokasi Apotek Di Kabupaten Sleman” oleh Cahyo Wicaksono dari Universitas Sanata Darma Yogyakarta. Aplikasi ini dapat melakukan pencarian apotek terdekat di wilayah Sleman. Teknologi yang digunakan adalah web servis REST API. Aplikasi ini juga dapat membantu pengguna untuk mencari obat lebih efisien.

Sedangkan pada penelitian ini, aplikasi akan dikembangkan untuk melakukan pencarian petugas PMI dan pendonor darah untuk kebutuhan permintaan donor darah. Aplikasi ini menggunakan *Firebase* yang digunakan untuk melakukan pelacakan dan memberikan informasi stok darah secara *real-time*. Aplikasi ini diberikan fitur pencarian lokasi petugas PMI terdekat dengan radius 10 km.

## 2.2 Dasar Teori

### 2.2.1 PMI

PMI atau Palang Merah Indonesia adalah suatu organisasi netral dan independen yang ada di Indonesia dan bergerak di bidang kemanusiaan. PMI terbentuk atas dasar bangsa Indonesia sendiri meski dipengaruhi oleh palang merah yang sikapnya universal. Terbentuknya PMI bermula dari dorongan jiwa manusia dan atas ke dasaran nasional. Ketika melaksanakan kegiatan PMI selalu menjaga prinsip yang dimilikinya yaitu kemanusiaan, kenetralan, sukarela, kemandirian, kesemestaan, dan kesatuan (Bobsusanto , 2016).

### 2.2.2 Android

*Android* adalah nama resmi dari sistem operasi atau OS (Operating System) yang berbasis dari *kernel Linux*. Sistem operasi ini banyak digunakan pada perangkat bergerak seperti ponsel cerdas atau *smartphone* yang didukung dan dikembangkan oleh Google (Gunawan , 2017).

*Android* memiliki *code* versi dan API (*Application Programming Interface*) level yang berbeda-beda. API Level adalah nilai integer yang secara unik mengidentifikasi revisi API kerangka kerja yang ditawarkan oleh versi platform *Android*. Penulis akan menggunakan *Android* versi 6.0 *Marshmallow* karena merupakan versi *Android* yang terbanyak digunakan pada saat ini. Berikut

distribusi versi *android* menurut tabel 2.2 berikut.

**Tabel 2.2** Api Level Versi Code dan Distribusi *Android*

No	Version Platform	Api Level	Version Code	Distribusi
1	Android 2.3.7	10	GINGERBREAD	0.3%
2	Android 4.0	15	ICE CREAM SANDWICH	0.4%
3	Android 4.1	16	JELLY BEAN	1.5%
4	Android 4.2	17		2.2%
5	Android 4.3	18		0.6%
6	Android 4.4	19	KIT KAT	10.3%
7	Android 5.0	21	LOLLIPOP	22.4%
8	Android 6.0	23	MARSHMALLOW	25.5%
9	Android 7.0	24	NOUGAT	22.9%
10	Android 7.1.1	25		8.2%
11	Android 8.0	26	OREO	4.9%

(Android , 2018)

### 2.2.3 Firebase

*Firebase* adalah *realtime database* yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK *Android*, *iOS*, dan *JavaScript*, semua klien akan berbagi sebuah *instance realtime database* dan

menerima update data terbaru secara otomatis (Firebase , 2018).

Salah satu kemampuan utama *Firestore* yaitu bersifat *real-time*. Sebagai ganti permintaan HTTP biasa, *Firestore Realtime Database* menggunakan sinkronisasi data—setiap kali data berubah, semua perangkat yang terhubung akan menerima update dalam waktu milidetik. Memberikan pengalaman yang kolaboratif dan *imersif* tanpa perlu memikirkan kode jaringan (Firebase , 2018).

*Firestore* merupakan database *NoSQL*, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. API *real-time Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan anda untuk membangun pengalaman *real-time* yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut. Berikut cara menulis data di *Firestore* :

**Tabel 2.3** Cara Menulis Data Di Firestore

No	Method	Penjelasan
1	set	Menulis atau mengganti data ke lokasi yang ditetapkan, seperti messages/users/<username>
2	update	Mengupdate beberapa kunci untuk lokasi yang ditetapkan tanpa mengganti semua data.
3	push	Menambahkan ke daftar data dalam database. Setiap kali mendorong node baru ke dalam daftar, database menghasilkan kunci unik,

		seperti messages/users/<unique-user-id>/<username>
4	transaction	Gunakan fitur transaction ketika bekerja dengan data kompleks yang bisa rusak karena update bersamaan

(Firebase , 2018).

Adapun cara untuk membaca data atau *listener* peristiwa untuk ketika terdapat perubahan data:

**Tabel 2.4** *Listener* Untuk Membaca Data

No	Method Callback	Penjelasan
1	ValueChanged	Membaca dan mendeteksi perubahan di seluruh konten lokasi.
2	ChildAdded	Mengambil daftar item atau mendeteksi penambahan daftar item. Disarankan untuk digunakan dengan ChildChanged dan ChildRemoved untuk memantau perubahan daftar.
3	ChildChanged	Mendeteksi perubahan pada item dalam daftar. Gunakan dengan ChildAdded dan ChildRemoved untuk memantau perubahan daftar.
4	ChildRemoved	Mendeteksi item yang sedang dihapus dari daftar. Gunakan dengan ChildAdded dan ChildChanged untuk memantau perubahan daftar.
5	ChildMoved	Mendeteksi perubahan pada urutan item dalam urutan daftar. Peristiwa ChildMoved selalu mengikuti peristiwa ChildChanged, yang menyebabkan urutan item berubah (berdasarkan metode 'urutkan berdasarkan' saat ini).

(Firebase , 2018).

#### 2.2.4 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan

dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, *JavaScript*, *Perl*, *Python* (Json , 2018).

### **2.2.5 Android Studio**

*Android Studio* adalah Lingkungan Pengembangan 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 dibuat untuk mempercepat pengembangan dan membantu Anda membuat aplikasi berkualitas tinggi untuk setiap perangkat *Android* (Android , 2018).

### **2.2.6 Mapbox**

*Mapbox* adalah platform data lokasi untuk aplikasi seluler dan web. *Mapbox* menyediakan blok bangunan untuk menambahkan fitur lokasi seperti peta, penelusuran, dan navigasi ke setiap pengalaman yang anda buat. *Mapbox* mengubah cara orang bergerak di sekitar kota dan menjelajahi dunia kita. *Mapbox* menjangkau lebih dari 300 juta orang setiap bulan. (Mapbox, 2018)

### 2.2.7 Rational Rose

*Rational Rose* adalah *tools* permodelan visual untuk pengembangan system berbasis objek yang handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan system. *Rational Rose* mendukung permodelan bisnis yang membantu para pengembang memahami system secara komprehensif. Aplikasi ini juga membantu analisis system dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas system secara keseluruhan sesuai dengan harapan dan keinginan pengguna (Mulyadi , 2013).