

Full Stack Development in Purpose for Digitalized Data on Connect Group

William^{#1}, Billy Susanto Panca^{#2}, Robby Tan^{#3}

*#Program Studi S1 Teknik Informatika Universitas Kristen Maranatha
Jl. Prof. drg. Surya Sumantri No. 65 Bandung*

¹Williamsutanto98@gmail.com

²Billy.sp@it.maranatha.edu

³Robby.tan@it.maranatha.edu

Abstract — Connect group is a small community of people who want to equally support each other to grow in Christ. Purpose for this development is digitalized data on connect group in church generically. Full stack usage on this development is to develop web and mobile applications. Technologies used are Google Maps, Google Maps Distance Matrix, Firebase, React JS, and React Native. Web application development is used to perform data processing users, schedules, presences, and locations. Mobile application development is used by user to perform features that support connect group activities. Digitalized data on connect group and the use of technologies can be implemented according to the existing requirement and support connect group activities.

Keywords— connect group, data processing, digital

I. PENDAHULUAN

Secara umum definisi komsel (komunitas sel) adalah suatu komunitas kecil yang terdiri dari orang-orang yang ingin sama-sama saling mendukung untuk bertumbuh dalam Kristus. Komsel dilakukan oleh umat nasrani untuk dapat sama-sama saling membangun, mendukung, dan menjaga untuk terus hidup dan bertumbuh dalam Kristus dan kebenaran Firman Tuhan.

Berdasarkan dari hasil pengamatan, masih banyak jemaat yang kesulitan untuk mencari lokasi hanya berdasarkan alamat, informasi yang kurang/ tidak lengkap, sulit untuk menghubungi kontak dari ketua komsel tersebut, dan mengetahui apakah komsel tersebut masih memiliki kuota atau tidak. Oleh karena itu, dibutuhkan aplikasi yang memudahkan jemaat untuk mengakses informasi mengenai komsel yang tersedia. Topik ini diajukan dengan tujuan untuk membuat sistem yang dapat mendukung kegiatan komsel yang sifatnya generik.

Aplikasi yang akan dibangun menyediakan fitur bagi jemaat untuk melihat komsel yang sedang berlangsung di lokasi terdekat, melihat daftar anggota yang mengikuti komsel tersebut, topik/ judul yang dibawakan, ketua pembawa komsel, tanggal, dan tempat pelaksanaan komsel. Harapan dari terbentuknya aplikasi ini dapat memudahkan para jemaat yang ingin mengikuti komsel yang ada, pengelolaan daftar komsel yang terstruktur, dan penjadwalan komsel semakin baik.

II. KAJIAN TEORI

A. Google Maps API (Application Programming Interface)

Google menyediakan Google Maps API yang memungkinkan untuk membangun aplikasi dengan memanfaatkan Google Maps [1]. Google Maps API merupakan sebuah API yang disediakan oleh Google untuk menggunakan peta Google (Google Map) dalam aplikasi yang dibangun. Google Maps API dapat memodifikasi peta dan informasi yang ada di dalamnya [2]. Google Maps API digunakan untuk menampilkan lokasi komsel.

B. Google Maps Android API

Google Maps Android API adalah API yang dikhususkan untuk platform yang memiliki sistem operasi Android. API ini digunakan bagi pengembang aplikasi Android seperti ojek *online*, cuaca, atau semua aplikasi yang memiliki fitur map dan lokasi secara *realtime* [3]. Google Maps Android API digunakan untuk menampilkan lokasi komersial pada aplikasi *mobile*.

C. Google Maps Distance Matrix API

Google Maps Distance Matrix API adalah API yang menyediakan jarak dan waktu tempuh dari matriks tempat asal dan tujuan. Informasi yang didapat yaitu berdasarkan rekomendasi rute dari titik awal dan akhir yang dikalkulasi oleh Google Maps API dan terdapat informasi yang mengandung nilai durasi dan jarak dari setiap pasang titik lokasi [4]. Google Maps Distance Matrix API digunakan untuk mengukur jarak dan waktu tempuh dari lokasi jemaat menuju lokasi komersial.

D. Firebase API

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *mobile apps developer*. Dengan adanya Firebase, *apps developer* bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar untuk mengerjakan backend [5].

Firebase menyediakan layanan DbaaS (Database as a Service) dengan konsep *realtime*. Firebase merupakan penyedia layanan cloud dengan backend sebagai servis. Firebase memiliki produk suatu database yang menyediakan API untuk memungkinkan *developer* menyimpan dan mensinkronisasi data lewat banyak client [6].

Android Firebase adalah cloud service sebagai bisnis backend untuk mengolah data untuk aplikasi *mobile*. Firebase mudah digunakan dan dapat dengan cepat membaca dan menulis data. Firebase dapat digunakan pada aplikasi iOS, Android, dan web dengan data yang didapat secara langsung dan penyimpanan [7].

Firebase Database digunakan untuk penyimpanan data pada aplikasi web dan *mobile*. Firebase Database dipilih karena cepat dan responsif, menggunakan JSON, dan tersedia untuk web dan *mobile*.

E. React JS

React JS adalah sebuah pustaka/ library javascript yang bersifat open source untuk membangun User Interface (UI) yang dibuat oleh Facebook. React JS hanya membuat semua hal yang berkaitan dengan tampilan dan logika. React JS ini diciptakan dengan tujuan untuk membangun aplikasi skala besar dengan data yang berubah dan terus berubah dari waktu ke waktu [8].

React JS merupakan JavaScript user interface library yang dikembangkan oleh Facebook. Ini menawarkan ide bagaimana cara bekerja dengan DOM (Document Object Model), mengolah alur aplikasi data, membuat elemen user interface sebagai komponen individu [9].

React JS digunakan untuk membangun aplikasi web. React JS dipilih karena mudah dipahami, fleksibel, efisien, dan cepat.

F. React Native

React Native adalah sebuah framework javascript yang dikembangkan oleh Facebook dan memungkinkan pihak *developer* untuk membuat aplikasi *mobile* pada platform Android atau iOS menggunakan teknologi web. React Native akan mengkompilasi aplikasi ke dalam kode native di mana

untuk Android akan dikompilasi dengan bahasa pemrograman Java dan untuk iOS akan dikompilasi dengan bahasa pemrograman Objective-C [10].

React Native adalah JavaScript framework untuk menuliskan secara native membuat aplikasi mobile untuk iOS dan Android. Ini berdasarkan React, JavaScript library milik Facebook untuk membuat user interfaces, bukan untuk browser, tetapi untuk perangkat mobile [11].

React Native digunakan untuk membangun aplikasi mobile. React Native dipilih karena menggunakan bahasa JavaScript, dapat mengembangkan aplikasi untuk platform Android dan iOS, dan komunitas yang besar.

G. Service-Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) adalah arsitektur yang memiliki tujuan untuk mencapai pelepasan keterikatan dari interaksi antar *software*. Servis merupakan unit kerja yang diselesaikan oleh penyedia layanan untuk mencapai hasil yang diinginkan untuk konsumen yang memakai layanan tersebut. Kedua penyedia dan pemakai layanan memiliki peran yang disediakan oleh perantara untuk menghubungkan keduanya [12].

H. QR Code

QR Code merupakan simbol dua dimensi. QR Code ditemukan pada tahun 1994 oleh Denso, salah satu perusahaan besar milik grup Toyota. Simbol dua dimensi ini awalnya dibuat untuk dipakai dalam kontrol produksi bagian otomotif, kemudian menyebar dan dipakai dalam bidang lain. Sekarang QR Code banyak terlihat dan dipakai dimana-mana [13]. QR Code digunakan untuk memudahkan pembacaan data menggunakan kamera.

I. JavaScript Object Notation (JSON)

JSON merupakan singkatan dari JavaScript Object Notation. JSON adalah format pertukaran data yang dipakai oleh banyak sistem dan sepakat untuk digunakan komunikasi data. JSON berdasarkan pada objek literal JavaScript [14]. JSON dipakai untuk melakukan pertukaran data pada Firebase Database.

III. ANALISIS DAN RANCANGAN SISTEM

A. Analisis Masalah

Adapun masalah yang dihadapi yaitu sebagai berikut:

1. Perekapian (data) komsel dilakukan secara manual.
2. Informasi lokasi pelaksanaan berupa alamat.
3. Tidak ada informasi kuota yang sudah terpenuhi.

B. Analisis Fitur

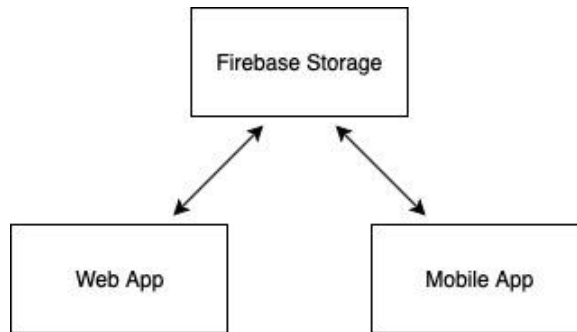
Adapun fitur yang akan dirancang yaitu sebagai berikut:

1. Aplikasi mobile dapat menampilkan jadwal komsel.
2. Aplikasi mobile dapat menampilkan lokasi komsel dengan estimasi jarak dan waktu tempuh.
3. Aplikasi mobile dapat mencatat presensi.
4. Aplikasi web dapat mengolah data jemaat.
5. Aplikasi web dapat mengolah data jadwal.

6. Aplikasi web dapat mengolah data presensi.
7. Aplikasi web dapat mengolah data lokasi.

C. Alur Komunikasi Data

Alur komunikasi data dimulai dari pengguna mengakses aplikasi *mobile* atau aplikasi *web*. Data didapat dari pengaksesan pada Firebase Storage. Pengguna perlu melakukan *login* untuk mendapatkan akses ke aplikasi. Aplikasi *mobile* melakukan *request* pada Firebase Storage untuk mendapatkan data jadwal komsel. Aplikasi *web* melakukan *request* pada Firebase Storage untuk mendapatkan data dan melakukan pengolahan data. Gambar 1 menggambarkan alur komunikasi data pada aplikasi *mobile* dan aplikasi *web*.



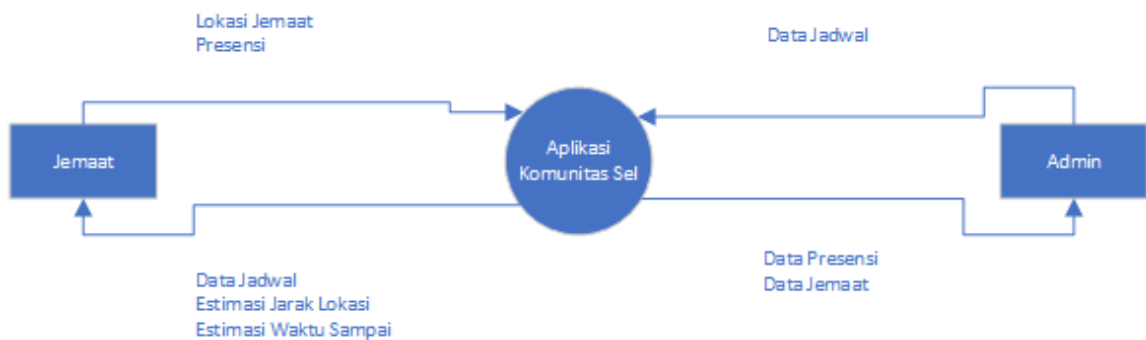
Gambar 1. Alur Komunikasi Data

D. Data Flow Diagram

Data Flow Diagram dibagi menjadi sebagai berikut:

1. Diagram Konteks

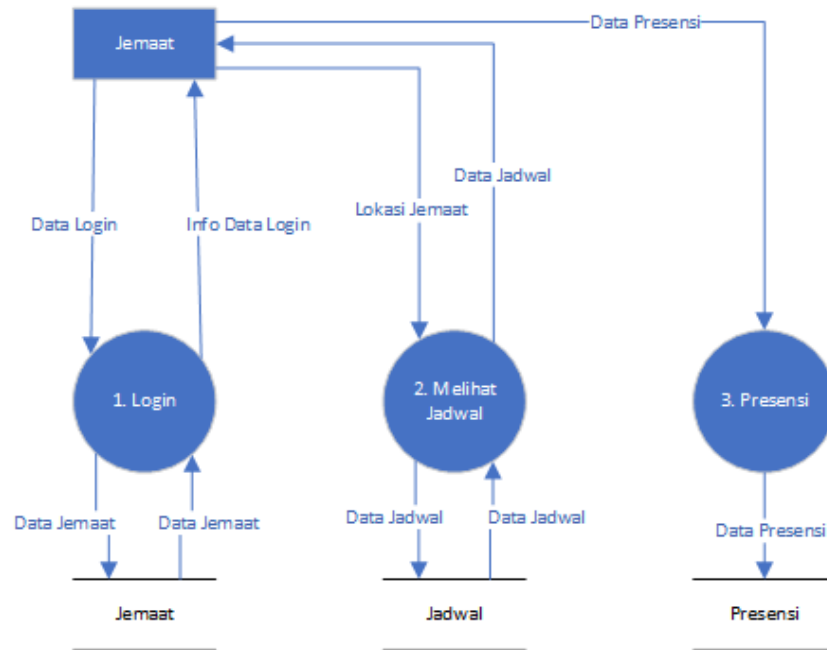
Diagram Konteks adalah diagram yang menggambarkan secara umum yang menjadi masukan, proses dan keluaran yang terjadi pada sebuah sistem. Berikut Diagram Konteks ditunjukkan pada Gambar 2.



Gambar 2. Diagram Konteks

2. DFD Level 1 Pada Jemaat

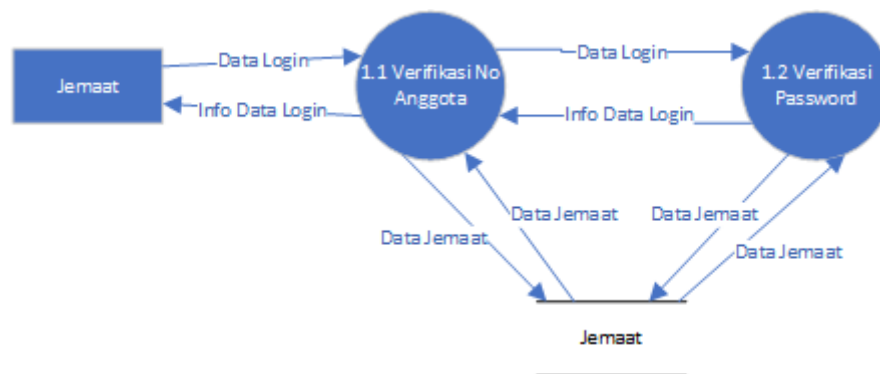
Berikut DFD Level 1 Pada Jemaat yang akan dibangun ditunjukkan pada Gambar 3.



Gambar 3. DFD Level 1 Pada Jemaat

3. DFD Level 2 Login Pada Jemaat

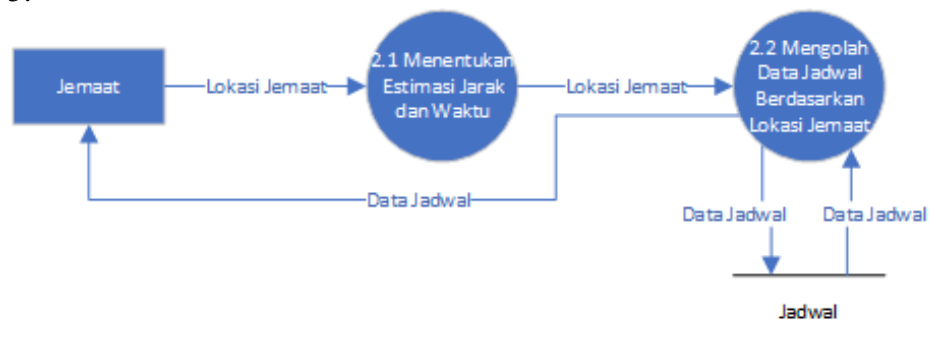
Berikut DFD Level 2 Login Pada Jemaat yang akan dibangun ditunjukkan pada Gambar 4.



Gambar 4. DFD Level 2 Login Pada Jemaat

4. DFD Level 2 Melihat Jadwal Pada Jemaat

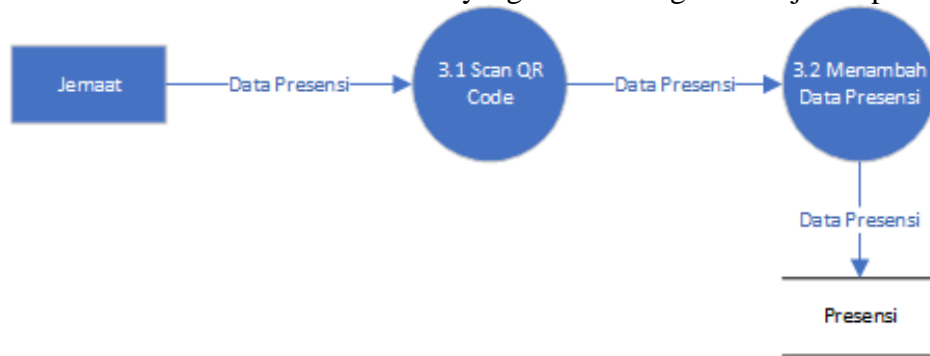
Berikut DFD Level 2 Melihat Jadwal Pada Jemaat yang akan dibangun ditunjukkan pada Gambar 5.



Gambar 5. DFD Level 2 Melihat Jadwal Pada Jemaat

5. DFD Level 2 Presensi Pada Jemaat

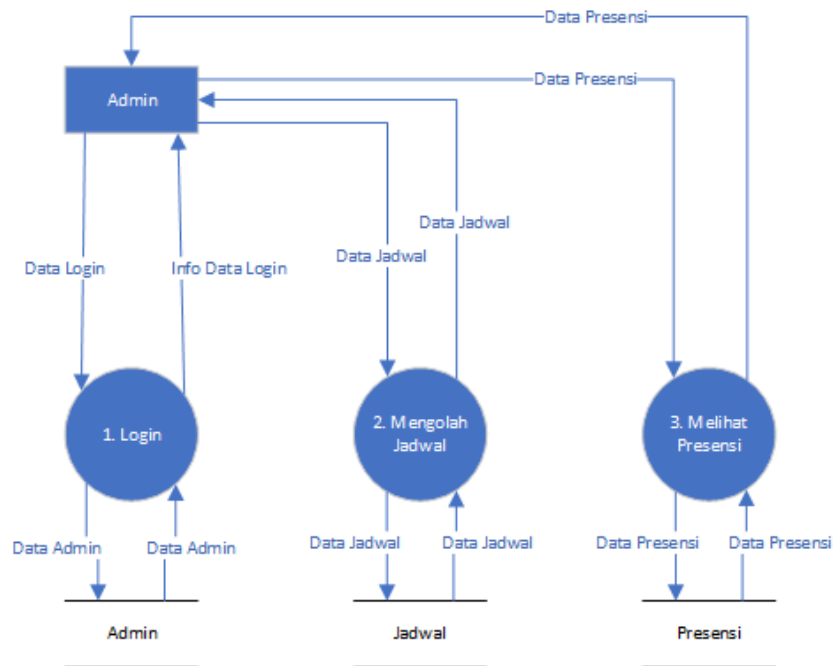
Berikut DFD Level 2 Presensi Pada Jemaat yang akan dibangun ditunjukkan pada Gambar 6.



Gambar 6. DFD Level 2 Presensi Pada Jemaat

6. DFD Level 1 Pada Admin

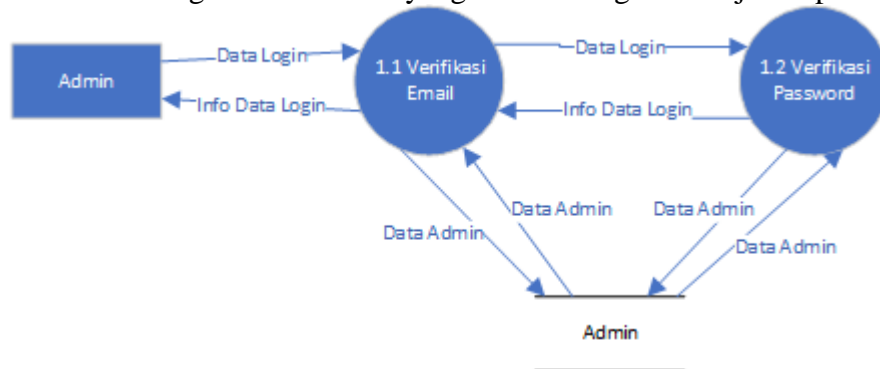
Berikut DFD Level 1 Pada Admin yang akan dibangun ditunjukkan pada Gambar 7.



Gambar 7. DFD Level 1 Pada Admin

7. DFD Level 2 Login Pada Admin

Berikut DFD Level 2 Login Pada Admin yang akan dibangun ditunjukkan pada Gambar 8.



Gambar 8. DFD Level 2 *Login* Pada Admin

8. DFD Level 2 Mengolah Jadwal Pada Admin

Berikut DFD Level 2 Mengolah Jadwal Pada Admin yang akan dibangun ditunjukkan pada Gambar 9.



Gambar 9. DFD Level 2 Mengolah Jadwal Pada Admin

9. DFD Level 2 Melihat Presensi Pada Admin

Berikut DFD Level 2 Melihat Presensi Pada Admin yang akan dibangun ditunjukkan pada Gambar 10.



Gambar 10. DFD Level 2 Melihat Presensi Pada Admin

E. Database REST API

Database yang digunakan yaitu *Firestore Database REST API*. Berikut *REST API* yang digunakan pada aplikasi *web* dan *mobile*.

1. Users

Field yang dibutuhkan ada *id* dengan tipe data *string*, *church* dengan tipe data *string*, *dateOfBirth* dengan tipe data *timestamp*, *email* dengan tipe data *string*, *memberNumber* dengan tipe data *string*, *name* dengan tipe data *string*, *password* dengan tipe data *string*, *phone* dengan tipe data *string*, *sex* dengan tipe data *string*.

2. Schedules

Field yang dibutuhkan ada *id* dengan tipe data *string*, *chiefId* dengan tipe data *string*, *date* dengan tipe data *timestamp*, *description* dengan tipe data *string*, *locationId* dengan tipe data

- string, quota dengan tipe data number, slot dengan tipe data number, title dengan tipe data string.
3. Presences
Field yang dibutuhkan ada id dengan tipe data string, attendanceDate dengan tipe data timestamp, bookingDate dengan tipe data timestamp, scheduleId dengan tipe data string, userId dengan tipe data string.
 4. Locations
Field yang dibutuhkan ada id dengan tipe data string, address dengan tipe data string, geopoint dengan tipe data geopoint, name dengan tipe data string.

IV. IMPLEMENTASI

A. Implementasi API

Berikut ini adalah implementasi API yang digunakan pada aplikasi *mobile* dan *web*.

1) Implementasi Google Maps Android API:

Implementasi Google Maps Android API dilakukan pada aplikasi mobile. Pertama yang dilakukan yaitu meng-install package yang bernama “react-native-maps”. Kedua, menuliskan kode import pada package yang telah di-install. Ketiga, menuliskan kode seperti pada Gambar 11 untuk menampilkan lokasi map yang ada.

```
1. import MapView from 'react-native-maps';  
2.  
3. <MapView  
4.   initialRegion={{  
5.     latitude: 37.78825,  
6.     longitude: -122.4324,  
7.     latitudeDelta: 0.0922,  
8.     longitudeDelta: 0.0421,  
9.   }}  
10. />
```

Gambar 10. Google Maps Android API

2) Implementasi Geolocation Service API:

Implementasi Geolocation Service API dilakukan pada aplikasi mobile. API ini berfungsi untuk mendapatkan titik lokasi pengguna. Pertama yang dilakukan yaitu meng-install package yang bernama “react-native-geolocation-service”. Kedua, menuliskan kode import pada package yang telah di-install. Ketiga, menuliskan kode seperti pada Gambar 12 untuk mendapatkan titik lokasi pengguna dengan menggunakan GPS.


```
1. import Geolocation from 'react-native-geolocation-service';
2.
3. if (hasLocationPermission) {
4.   Geolocation.getCurrentPosition(
5.     position => {
6.       console.log(position);
7.     },
8.     error => {
9.       // See error code charts below.
10.      console.log(error.code, error.message);
11.    },
12.    { enableHighAccuracy: true, timeout: 15000, maximumAge: 10000 }
13.  );
14. }
```

Gambar 11. Geolocation Service API

3) Implementasi Firebase API Mobile:

Implementasi Firebase API dilakukan pada aplikasi mobile. Pertama yang dilakukan yaitu menginstall package yang bernama “firebase”. Kedua, menuliskan kode import pada package yang telah di-install. Ketiga, menuliskan kode config pada firebase agar dapat digunakan seperti pada Gambar 12.

```
1. import * as firebase from 'firebase';
2. import 'firebase/firestore';
3.
4. const config = {
5.   apiKey: '',
6.   authDomain: '',
7.   databaseURL: '',
8.   projectId: '',
9.   storageBucket: '',
10.  messagingSenderId: '',
11.  appId: '',
12. };
13.
14. firebase.initializeApp(config);
15.
16. const databaseRef = firebase.firestore();
```

Gambar 12. Firebase API Mobile

4) Implementasi Google Maps Distance Matrix API:

Implementasi Google Maps Distance Matrix API dilakukan pada aplikasi mobile. Cara menggunakan API yaitu dengan mengirim request pada URL yang disediakan oleh Google. Kode pemrograman yang digunakan dapat dilihat pada Gambar 13. Parameter yang dibutuhkan yaitu origins, destinations, dan key.

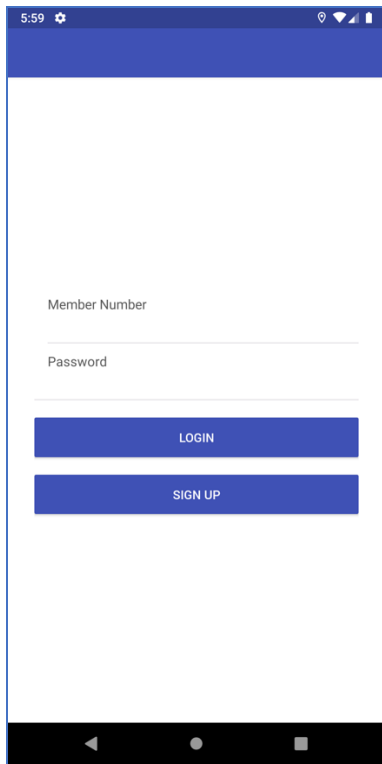
```
1. await fetch(
2.   `https://maps.googleapis.com/maps/api/distancematrix/json?origins=
3.   LATITUDE, LONGITUDE&destinations=LATITUDE, LONGITUDE&key=API_KEY`,
4. )
5.   .then(response => response.json())
6.   .then(responseJson => {
7.     distance = responseJson.rows[0].elements[0].distance.value;
8.     duration = responseJson.rows[0].elements[0].duration.value;
9.   })
10.  .catch(error => console.log(error));
```

Gambar 13. Google Maps Distance Matrix API

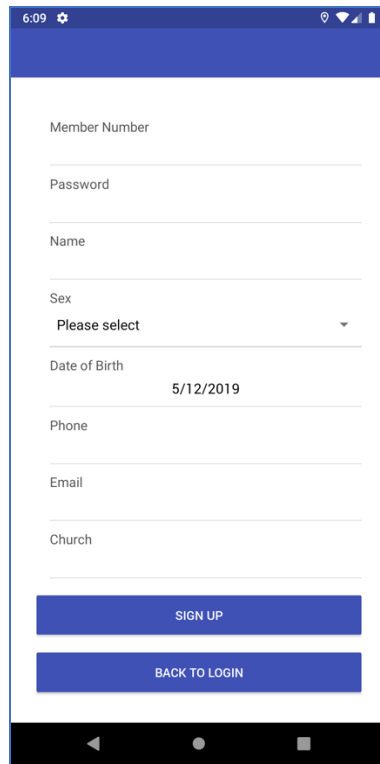
B. Implementasi API

Berikut ini adalah implementasi API yang digunakan pada aplikasi *mobile* dan *web*.

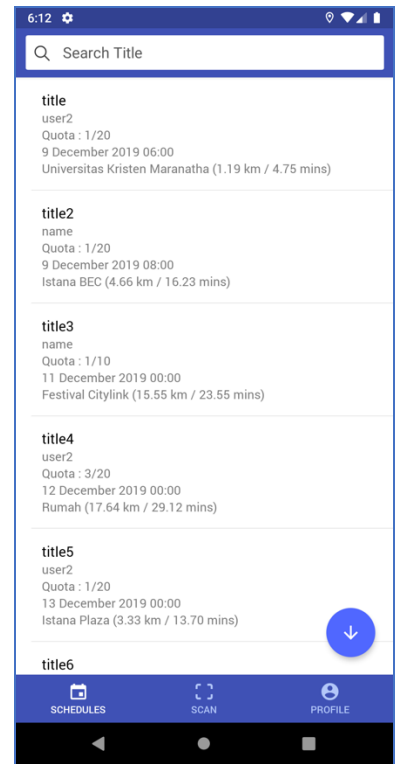
- 1) *Implementasi Login*: Gambar 14 adalah hasil implementasi tampilan login.
- 2) *Implementasi Sign Up*: Gambar 15 adalah hasil implementasi tampilan sign up.
- 3) *Implementasi List Jadwal*: Gambar 16 adalah hasil implementasi tampilan list jadwal.



Gambar 14. Implementasi Tampilan *Login*

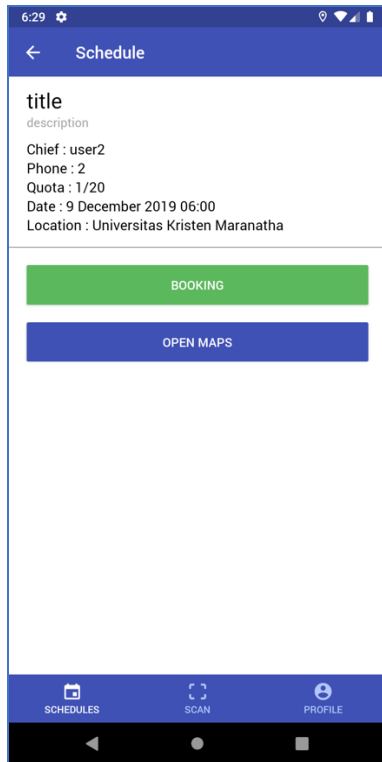


Gambar 15. Implementasi Tampilan *Sign Up*

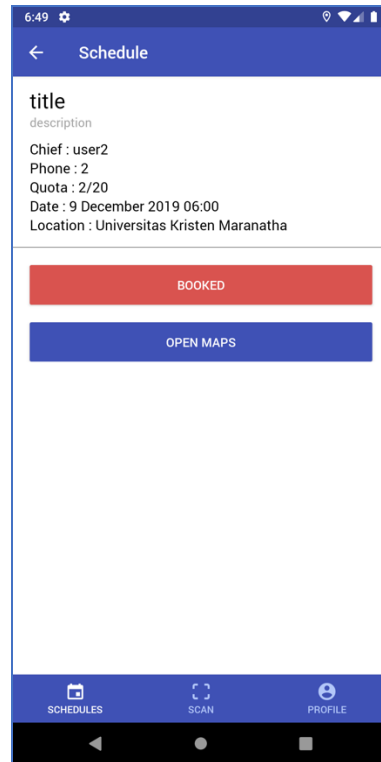


Gambar 16. Implementasi Tampilan *List Jadwal*

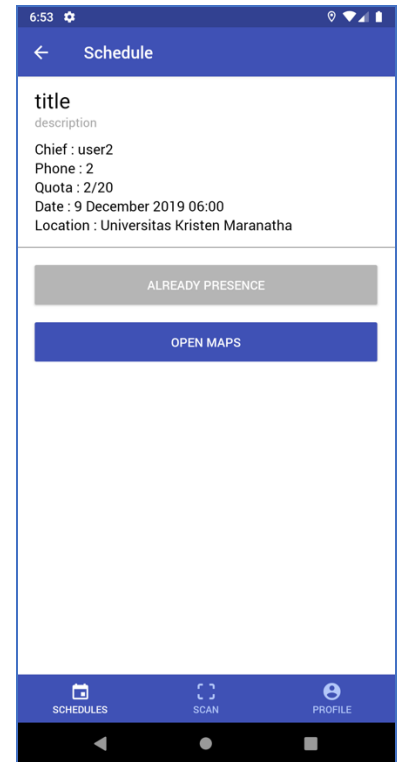
4) *Implementasi Detail Jadwal*: Gambar 17 adalah hasil implementasi tampilan detail jadwal dengan kondisi user tidak melakukan daftar pada jadwal tersebut. Gambar 18 adalah hasil implementasi tampilan detail jadwal dengan kondisi user sudah melakukan daftar pada jadwal tersebut. Gambar 19 adalah hasil implementasi tampilan detail jadwal dengan kondisi user sudah melakukan presensi pada jadwal tersebut.



Gambar 17. Implementasi Tampilan *Detail* Jadwal Kondisi Pertama



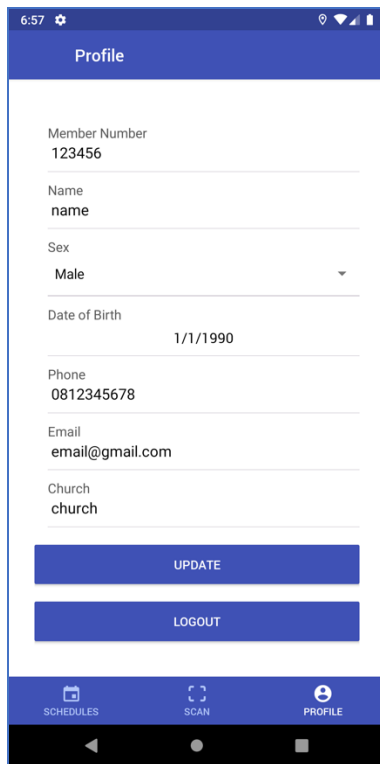
Gambar 18. Implementasi Tampilan *Detail* Jadwal Kondisi Kedua



Gambar 19. Implementasi Tampilan *Detail* Jadwal Kondisi Ketiga

5) *Implementasi Mengubah Profile*: Gambar 20 adalah hasil implementasi tampilan profile yang menampilkan data user yang sedang login dan tombol untuk melakukan logout.

6) *Implementasi Scan QR Code*: Gambar 21 adalah hasil implementasi dari tampilan scan yang digunakan untuk melakukan presensi pada jadwal dengan scan QR code.



Gambar 20. Implementasi Tampilan *Profile*



Gambar 21. Implementasi Tampilan *Scan*

V. PENGUJIAN

A. Pengujian *Black Box*

Pengujian pada aplikasi ini dilakukan dengan menggunakan metode *black box testing*. Pengujian *black box* dibagi menjadi dua yaitu pada aplikasi web dan mobile.

B. Evaluasi

Pengujian berikutnya dilakukan lewat evaluasi, di mana responden merupakan kepala pengurus komsel di salah satu gereja di Kota Bandung. Pengujian dilakukan dengan sejumlah fitur yang diajukan pada responden setelah mencoba aplikasi. Evaluasi dilakukan oleh kepala pengurus komsel di salah satu gereja di Kota Bandung. Evaluasi dibagi menjadi dua, yaitu evaluasi pada aplikasi *web* dan aplikasi *mobile*. Tabel 1 adalah hasil evaluasi pada aplikasi *web*. Tabel 2 adalah hasil evaluasi pada aplikasi *mobile*.

TABEL I
HASIL EVALUASI APLIKASI *WEB*

No	Fitur	Terpenuhi
1.	<i>Login</i>	✓
2.	Mengolah data jemaat (melihat, menambah, mengubah, menghapus)	✓
3.	Mengolah data jadwal komsel (melihat, menambah, mengubah, menghapus)	✓
4.	Mengolah data presensi (melihat, mengubah, menghapus)	✓
5.	Mengolah data lokasi (melihat, menambah, mengubah, menghapus)	✓
6.	Menambah jadwal komsel dengan titik lokasi	✓
7.	Melihat daftar presensi yang dilakukan oleh jemaat pada jadwal komsel	✓
8.	Menyimpan gambar <i>QR code</i> pada jadwal komsel untuk digunakan sebagai presensi oleh jemaat	✓

TABEL II
HASIL EVALUASI APLIKASI *MOBILE*

No	Fitur	Terpenuhi
1.	<i>Login</i>	✓
2.	Melihat daftar jadwal komsel yang akan diadakan	✓
3.	Melihat jarak dan waktu tempuh dari lokasi jemaat menuju lokasi komsel	✓
4.	Melihat slot dan kuota pada jadwal komsel	✓
5.	Melakukan <i>booking</i> untuk memesan kuota	✓
6.	Melihat lokasi komsel dengan Google Maps	✓
7.	Melakukan presensi dengan <i>scan QR code</i> yang disediakan oleh ketua komsel	✓

VI. SIMPULAN DAN SARAN

A. Simpulan

Setelah melakukan analisis, implementasi, dan pengujian aplikasi web dan mobile komunitas sel, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Dari hasil implementasi aplikasi, semua kebutuhan yang telah dianalisis dapat terlaksana menggunakan aplikasi web dan mobile. Aplikasi web dapat digunakan untuk melakukan otentifikasi dan otorisasi, mengelola data jemaat, jadwal, presensi, dan lokasi. Aplikasi mobile dapat digunakan untuk melakukan otentifikasi dan otorisasi, melihat jadwal yang dilengkapi jarak dan waktu tempuh, melakukan presensi. Perubahan sistem manual menjadi digital dan penggunaan teknologi dari fitur yang telah dianalisis dapat diimplementasi sesuai requirement yang ada. Aplikasi web dan mobile sudah memenuhi hasil dari analisis fitur yang dibutuhkan dan dapat terlaksana dengan baik berdasarkan pengujian black box.
2. Penggunaan API mendukung proses pembuatan sistem dan dapat mengakomodir fitur – fitur yang dibutuhkan untuk memenuhi requirement yang ada. Fitur pada aplikasi yang sudah diimplementasi dapat mendukung kegiatan komsel di gereja berdasarkan hasil evaluasi pada salah satu gereja di Kota Bandung.

B. Saran

Saran untuk pengembangan aplikasi di masa yang akan datang adalah optimasi kemampuan untuk pengolahan data, keamanan, peningkatan tampilan, peningkatan kecepatan waktu akses, dan petunjuk penggunaan bagi pengguna baru. Pengembangan fitur yang mencakup kegiatan komsel secara generik perlu dilakukan agar dapat mendukung kegiatan komsel di sebagian besar gereja.

DAFTAR PUSTAKA

- [1] G. Svennerberg, *Beginning Google Maps API 3*, New York: Apress, 2010.
- [2] M. Dian, "Tutorial Google Maps API: Menampilkan Peta Google Map di dalam Web," 30 December 2016. [Online]. Available: <https://www.petanikode.com/google-map-dasar/>.
- [3] M. David, "Berkenalan Dengan Google Map API," 15 June 2017. [Online]. Available: <https://www.dumetschool.com/blog/berkenalan-dengan-google-map-api>.
- [4] Google Developers, "Distance Matrix API - Developer Guide," [Online]. Available: <https://developers.google.com/maps/documentation/distance-matrix/intro>.
- [5] "Apa itu Firebase?," 7 July 2016. [Online]. Available: <https://www.saungit.org/2016/07/apa-itu-firebase.html>.
- [6] I. Koestiawan, "Pengertian Dan Sejarah Firebase," 13 April 2018. [Online]. Available: <https://jogjaweb.co.id/blog/catatan/pengertian-dan-sejarah-firebase>.
- [7] B. Stonehem, *Google Android Firebase: Learning the Basics*, First Rank Publishing, 2016.
- [8] S. Muslim, "Pendahuluan — Seminggu Menjadi Master React JS — Seri Tingkat Dasar," 14 August 2017. [Online]. Available: <https://medium.com/skyshidigital/1-pendahuluan-seminggu-menjadi-master-react-js-seri-tingkat-dasar-7b6ec7b04af>.
- [9] A. Fedosejev, *React.js Essentials*, Birmingham: Packt Publisher, 2015.
- [10] A. Muhandian, "Persiapan Awal untuk Belajar React Native," 28 July 2018. [Online]. Available: <https://www.petanikode.com/react-native-dasar/>.
- [11] B. Eisenman, *Learning React Native: Building Native Mobile Apps with JavaScript*, Sebastopol: O'Reilly, 2016.
- [12] H. He, "What Is Service-Oriented Architecture," 2003.

- [13] T. J. Soon, "QR Code," *Section Three*, 2008.
- [14] L. Bassett, *Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON*, Sebastopol: O'Reilly Media, 2015.
- [15] B. Rahmadhian, Y. Ariyanto dan D. A. Irawati, "APLIKASI PEMESANAN OJEK ONLINE BERBASIS ANDROID MENGGUNAKAN METODE DIJKSTRA," vol. 3, no. 1, p. 60, 1 November 2016.
- [16] M. D. M. Zainurachman dan B. S. Andoko, "PENGEMBANGAN WEBSITE SISTEM Pencarian dan Publikasi Tempat di Kota Malang Menggunakan Geolocation," vol. 2, no. 3, p. 106, 1 May 2016.