

Aplikasi Android Misi Youth Berbasis Flutter

Yoel Oscar^{#1}, Erico Darmawan Handoyo, S.Kom., MT.²

[#]Program Studi Teknik Informatika, Universitas Kristen Maranatha
Jln. Surya Sumantri No 65, Kota Bandung

¹1672063@maranatha.ac.id

³erico.dh@it.maranatha.edu

Abstract — In this modern era, there is a lot of *information* about the activities of young people in a *church*. And each church has its own role to develop young people in it besides, it also requires a centralized source of information. For this reason, with the android application, every young person can get the latest *information* regarding activities in the *church*. For now, the young people of the Assemblies of God *Church* can only get *information* about *youth* activities only through word of mouth or the results shared by others. The purpose of this study is to develop an android application which can help every *youth* congregation in obtaining *information* for activities in the *youth* community of the Assemblies of God *Church*, to be able to realize it, a research method is carried out such as a direct approach, spreading form about what are the needs for church youth, conduct interviews and design applications that will later be made. The application that will be developed will use the *Dart* programming language with the flutter *framework* and the firebase database, where it is hoped that this application can help *church youth* in getting information about activities in the *Church*.

Keywords— *Information, Church, Framework, Youth, Dart.*

I. PENDAHULUAN

Komunitas kaum muda di dalam gereja memiliki peran penting didalam kebangunan rohani setiap individu remaja yang akan beranjak dewasa [1]. Maka dari itu, komunitas kaum muda gereja memiliki beberapa kegiatan, yaitu seperti ibadah setiap hari sabtu, perayaan paskah, perayaan hari Nasional kaum muda, perayaan Natal kaum muda dan *event-event* yang akan diadakan dikemudian hari. Tidak hanya kegiatan di dalam gereja, ada juga renungan, saat teduh, dan pengingat akan membaca Alkitab. Menurut survei yang dilakukan di lingkungan komunitas kaum muda gereja sidang jemaat Allah, Maka dari itu, Aplikasi yang dibuat dimaksudkan dapat memberikan informasi akan kegiatan Dengan studi kasus Komunitas kaum muda Gereja Sidang Jemaat Allah melalui pembuatan aplikasi android misi youth berbasis flutter. Sistem yang dibuat bertujuan untuk mengumpulkan kegiatan, renungan dan informasi lainnya didalam satu sistem, sehingga dapat di akses dan di terima oleh jemaat.

II. KAJIAN TEORI

A. Aplikasi Android

Aplikasi yang akan dibuat Merupakan sebuah Aplikasi dimana Aplikasi ini dapat menampung kumpulan informasi, kegiatan, dan acara yang akan datang di komunitas kaum muda gereja dalam bentuk informasi digital yang dapat di akses oleh jemaat kaum muda. Di era industri 4.0 ini, teknologi sudah sangat berkembang dan hal tersebut dapat membantu semua aktivitas manusia dan juga dapat digunakan untuk meningkatkan pencapaian informasi didalam ruang lingkup aktivitas gereja melalui media *Smartphone* atau komputer sehingga jemaat dapat menerima informasi yang benar [2]. Sebuah riset telah dilakukan ditahun 2019, dimana penggunaan *smartphone* mengalami peningkatan yang cukup signifikan dan penggunaan PC atau Laptop mengalami penurunan. Pengguna lebih memilih untuk memakai *smartphone* untuk mengakses informasi ketika berada diluar ruangan dikarenakan mudah untuk dibawa dan mudah mengakses informasi yang ingin didapat [3].

Aplikasi yang akan dibuat, merupakan aplikasi android dengan *framework* flutter. Karena pengguna Android terus bertambah setiap tahunnya. Pengembangan cukup mudah karena *framework* menyediakan *library* pendukung untuk membuat user interface yang menarik, dan database untuk menyimpan data informasi kegiatan di dalam komunitas kaum muda gereja

B. Framework Flutter

Flutter merupakan *framework* yang diciptakan oleh Google untuk membuat aplikasi berbasis Android dan juga IOS, yang mana bertujuan untuk dimudahkannya *developer* untuk membuat aplikasi secara cepat dan produktif, seperti *Hot Reload*, *catalog widget*, performa yang maksimal dan banyaknya komunitas yang mendukung flutter untuk menjadi *framework* yang terbaik. Flutter sangat mirip dengan beberapa *software* terkenal lainnya.

Keuntungan dalam menggunakan *Framework* Flutter :

1. Untuk pengguna, Flutter dapat membuat *interface* aplikasi lebih atraktif, dan ini dapat membuat *user* menjadi lebih tertarik untuk menggunakan aplikasi yang di buat oleh flutter.
2. Flutter mudah dimengerti bagi setiap *developer* awal yang memasuki dunia pembangunan aplikasi *mobile*, dan sangatlah mudah membangun aplikasi dengan flutter, dan menggunakan *framework* flutter, sangatlah memangkas waktu dimana dalam pembuatan aplikasi, efisiensi waktu sangatlah di perlukan sehingga aplikasi dapat di produksi secara tepat waktu dan mengurangi biaya dan tingkat kompleksitas dalam pembuatan aplikasi yang di rancangan.
3. Setiap perancang, dapat memasukan imajinasi mereka kedalam aplikasi yang akan dibangun, sehingga nantinya aplikasi yang dibuat dapat sangat menarik, sehingga *developer* tidak perlu bingung akan model seperti apa aplikasi nanti akan dibuat.

Framework Flutter memiliki karakter yang ringan serta jika *developer* sudah memahami bahasa pemrograman Dart, maka *developer* tidak perlu merasa kesulitan akan *framework* flutter [4].

C. Bahasa pemrograman Dart

Bahasa pemrograman Dart, merupakan bahasa pemrograman yang sangat muda di antara bahasa pemrograman lain. Dart, memiliki berbagai fitur yang menarik, dan bahasa pemrograman Dart berbasis kelas-kelas dan bahasa pemrograman berorientasi objek. dengan fitur tersebut *developer* dapat membuat program yang sangat fungsional dan program yang sangat mumpuni. [8]. Dan contoh dari bahasa pemrograman dart dapat dilihat sebagai berikut

```
lib > main.dart > MyApp > build
1  import 'package:flutter/material.dart';
2
3  void main() => runApp(MyApp());
4
5  class MyApp extends StatelessWidget {
6    // This widget is the root of your application.
7    @override
8    Widget build(BuildContext context) {
9      return MaterialApp(
10         title: 'Flutter Demo',
11         theme: ThemeData(
12           // This is the theme of your application.
13           //
14           // Try running your application with "flutter run". You'll see the
15           // application has a blue toolbar. Then, without quitting the app, try
16           // changing the primarySwatch below to Colors.green and then invoke
17           // "hot reload" (press "r" in the console where you ran "flutter run",
18           // or simply save your changes to "hot reload" in a Flutter IDE).
19           // Notice that the counter didn't reset back to zero; the application
20           // is not restarted.
21           primarySwatch: Colors.blue,
22         ), // ThemeData
23         home: MyHomePage(title: 'Flutter Home Page'),
24       ); // MaterialApp
25     }
26 }
```

Gambar 1 Baris Kode Pemrograman Dart

D. Modularitas

Bahasa pemrograman *Dart* di disain dengan cara menyimpan modul di dalam pikiran, pada bahasa pemrograman *Dart* modularitas *Dart* di simpan melalui *packages*, *libraries*, dan *classes*. Sebuah *Library* mengeluarkan fungsionalitas seperti kumpulan dari antar muka. Pada konsep tersebut, sangat mirip dengan beberapa pemisahan dari objek di dalam pemrograman berorientasi objek, memisahkan beberapa aplikasi kedalam perpustakaan membantu mengurangi tingkat kerumitan dari kode yang dibuat dan memudahkan *developer* dalam melakukan pembaharuan kode, didalam *Dart*, *library* dapat di implementasikan sebagai fungsi yang tidak rumit, seperti, *single class*, *several class*, atau sekumpulan koleksi

Sebuah *package* merupakan tempat yang terdapat *pubspec.yaml* file dan mungkin terdapat beberapa sumber atau *library* tertentu. Di dalam *pubspec.yaml* terdapat informasi mengenai *package*, *authors*, dan *dependencies* yang terdapat di dalam *pubspec.yaml* [8] dan dibawah ini merupakan contoh dari isi dari *pubspec.yaml* :

```
! pubspecyaml
1 name: android_new_test
2 description: A new Flutter project.
3
4 # The following defines the version and build number for your application.
5 # A version number is three numbers separated by dots, like 1.2.43
6 # followed by an optional build number separated by a +.
7 # Both the version and the builder number may be overridden in flutter
8 # build by specifying --build-name and --build-number, respectively.
9 # In Android, build-name is used as versionName while build-number used as versionCode.
10 # Read more about Android versioning at https://developer.android.com/studio/publish/versioning
11 # In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.
12 # Read more about iOS versioning at
13 # https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundation
14 version: 1.0.0+1
15
16 environment:
17   sdk: ">=2.1.0 <3.0.0"
18
19 dependencies:
20   flutter:
21     sdk: flutter
22
23   # The following adds the Cupertino Icons font to your application.
24   # Use with the CupertinoIcons class for iOS style icons.
25   cupertino_icons: ^0.1.2
26
27 dev_dependencies:
28   flutter_test:
29     sdk: flutter
30
```

Gambar 2 Pubspec.yaml

E. Firebase Database

Sistem yang dibuat akan menggunakan database firebase untuk digunakan sebagai penyimpanan data dalam aplikasi android misi youth berbasis flutter. Database ini memiliki fitur *Real Time Database* dimana fitur ini dapat mempermudah proses *backend*.

Database ini merupakan sistem yang dikembangkan oleh Google untuk membantu developer dalam pengembangan aplikasi *Android*. Database ini memiliki sistem secara Cloud, dan database ini memiliki fitur yang mana data dapat diakses melalui *local cache* [9]. Fitur yang terdapat didalam firebase seperti *firebase remote config*, *firebase notification* dan, *Firestore Real Time Database*. Berikut penjelasan Fitur yang terdapat di Firebase, yaitu:

1. *Firestore Remote Config* Merupakan fitur yang memungkinkan konfigurasi didalam aplikasi tanpa harus pembaharuan aplikasi didalam penyedia layanan publish aplikasi
2. *Firestore Real Time Database* merupakan layanan yang mengembangkan antarmuka pemrograman aplikasi yang di selaraskan dengan seluruh klien dan disimpan di *Firestore Cloud*.
3. *Firestore Notification* merupakan layanan yang memungkinkan pemberitahuan kepada pengguna yang di tujukan oleh pengembang.

III. ANALISIS DAN PERANCANGAN SISTEM

A. Gambaran Umum

Penelitian ini memiliki studi kasus jemaat kaum muda Gereja Sidang jemaat Allah yang beralamatkan di Jl. Gatot Subroto No.18 Cimahi. Tujuan penelitian ini adalah untuk memberikan solusi bagi pengurus komunitas kaum muda dan jemaat komunitas kaum muda Gereja Sidang Jemaat Allah akan kebutuhan informasi didalam kegiatan kaum muda melalui perangkat Android. Kunci utama dari perancangan sistem ini adalah memuat semua informasi mengenai kegiatan komunitas kaum muda di gereja yang dibentuk menjadi sebuah aplikasi pada nantinya.

B. Fitur-fitur aplikasi

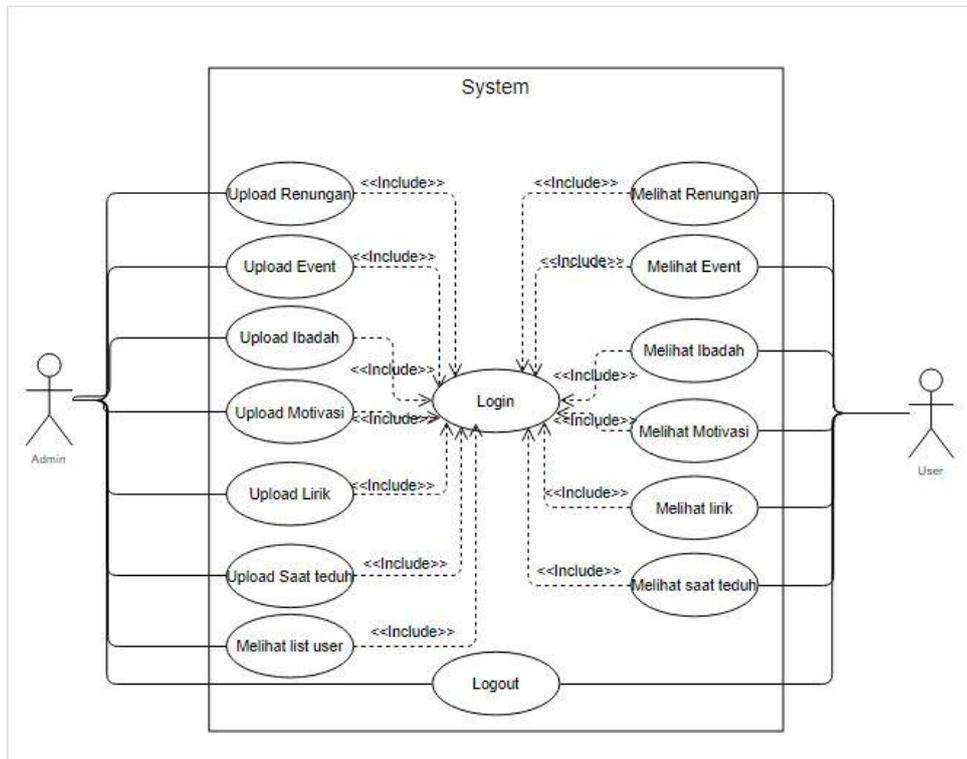
Berikut ini merupakan penjelasan dari fitur-fitur yang akan dibuat didalam Aplikasi Android Misi youth berbasis Flutter

1. Aplikasi ini dapat menyimpan data login.
2. Fitur untuk melihat event-event yang akan dijalankan.

3. Fitur untuk melihat ibadah kaum muda.
4. Fitur untuk melihat renungan.
5. Fitur untuk melihat saat teduh.
6. Fitur untuk menampilkan quotes motivasi dan ayat alkitab.

C. Rancangan use case diagram

Didalam Sistem yang akan di buat, terdapat dua jenis pengguna yaitu Pengurus kaum muda atau sebagai *Admin* dan Jemaat sebagai *User*. Admin dapat melakukan *Login*, menambahkan jadwal, menambahkan dan mengedit data jemaat, menambahkan renungan, menambahkan data ibadah, menambahkan pengumuman, dan *Logout*. *User* dapat melakukan *Login*, membaca renungan, melihat pengumuman, melihat data ibadah dan *Logout*. Berikut merupakan gambar Use Case dari penjelasan diatas:



Gambar 3 Use Case Alur Aplikasi Android Misi Youth

D. Skenario use case diagram

Berdasarkan Gambar 3 *Use Case Diagram* diatas, berikut adalah penjelasan dari skenario *use case diagram* yang telah di buat:

1) *Login*

- Nama *Use Case* : *Login*
- Aktor : User / Jemaat dan Admin / Pengurus Gereja
- Deskripsi : User / Admin mengisi data di halaman *login*
- Pre-condition* : User / Admin Belum pernah melakukan *login*
- Post-condition* : User / Admin telah melakukan *login* dan berinteraksi dengan program.

TABEL III
SKENARIO LOGIN ADMIN DAN USER

Aksi User/Admin	Reaksi Sistem
1.User/Admin Melengkapi <i>credential</i> yang di perlukan	
2.User/Admin Menekan tombol Login	
	3.Melakukan validasi.
	4.Jika berhasil, maka program akan menampilkan <i>dashboard</i> .

2) *Melihat Renungan*

- Nama *use case* : Membaca renungan.
Aktor : *User/Jemaat*.
Deskripsi : *User* membaca renungan.
Pre-condition : *User* melihat list renungan
Post-condition : *User* telah membaca renungan yang dipilih.

TABEL III
SKENARIO MEMBACA RENUNGAN USER

Aksi User	Reaksi Sistem
1.User menekan tombol renungan.	
2.User memilih renungan yang akan dibaca	
	3.sistem menampilkan isi dari renungan yang di pilih user

3) *Melihat Event*

- Nama *use case* : Membaca *event*.
Aktor : *User/Jemaat*.
Deskripsi : *User* membaca *event*.
Pre-condition : *User* melihat isi *event*.
Post-condition : *User* telah membaca *event*.

TABEL III
SKENARIO MELIHAT EVENT

Aksi User	Reaksi Sistem
1.User menekan tombol event.	
2.User memilih event yang akan dibaca	
	3.sistem menampilkan isi dari event yang di pilih user

4) *Melihat Lirik*

- Nama *use case* : Membaca lirik.
Aktor : *User/Jemaat*.
Deskripsi : *User* membaca lirik.
Pre-condition : *User* melihat isi lirik.
Post-condition : *User* telah melihat lirik.

TABEL III
SKENARIO MELIHAT LIRIK USER

Aksi User	Reaksi Sistem
-----------	---------------

1.User menekan tombol menu lirik.	
2.User memilih lirik yang akan dibaca	
	3.sistem menampilkan isi dari lirik yang di pilih user

5) *Melihat Ibadah*

Nama *use case* : Membaca ibadah.
Aktor : *User/Jemaat*.
Deskripsi : *User* membaca ibadah.
Pre-condition : *User* melihat isi ibadah.
Post-condition : *User* telah membaca ibadah.

TABEL III
SKENARIO MELIHAT IBADAH USER

Aksi User	Reaksi Sistem
1.User menekan tombol ibadah.	
2.User memilih ibadah mana yang akan di lihat	
	3.sistem menampilkan isi dari ibadah yang di pilih user

6) *Melihat Saat teduh*

Nama *use case* : Membaca saat teduh.
Aktor : *User/Jemaat*.
Deskripsi : *User* membaca saat teduh.
Pre-condition : *User* melihat isi saat teduh.
Post-condition : *User* telah membaca saat teduh.

TABEL III
SKENARIO MELIHAT SAAT TEDUH USER

Aksi User	Reaksi Sistem
1.User menekan tombol saat teduh.	
2.User memilih saat teduh mana yang akan di lihat	
	3.sistem menampilkan isi dari saat teduh yang di pilih user

7) *Melihat Motivasi*

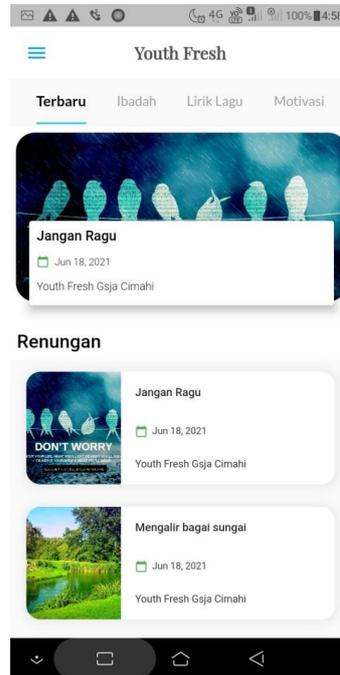
Nama *use case* : Membaca motivasi.
Aktor : *User/Jemaat*.
Deskripsi : *User* membaca motivasi.
Pre-condition : *User* melihat isi motivasi.
Post-condition : *User* telah membaca motivasi.

TABEL III
SKENARIO MELIHAT MOTIVASI USER

Aksi User	Reaksi Sistem
1.User menekan tombol motivasi.	
2.User memilih motivasi mana yang akan di lihat	
	3.sistem menampilkan isi dari motivasi yang di pilih user

IV. IMPLEMENTASI

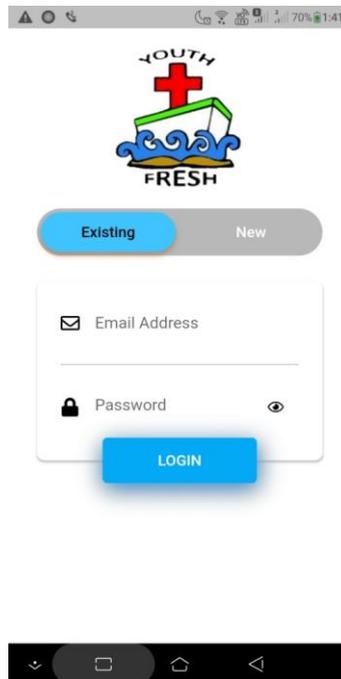
A. Dashboard User



Gambar 4 Halaman Utama User

Gambar diatas merupakan halaman *Dashboard User* dimana tampilan ini merupakan tampilan sebelum *login*. User dapat melihat informasi seperti renungan terbaru, ibadah yang telah di jalankan, lirik lagu rohani, dan motivasi harian, akan tetapi jika user telah melakukan login, maka isi *drawer* dari user akan bertambah menu nya. Dan pada halaman utama ini ketika user mengklik salah satu judul maka akan menampilkan isi dari judul yang telah di klik.

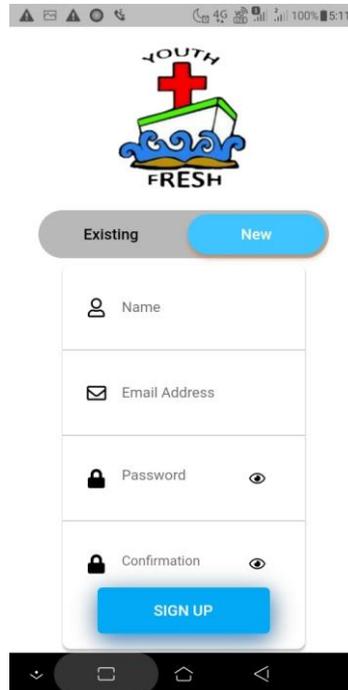
B. Login User



Gambar 5 Halaman Login User

Gambar *login user* merupakan halaman *login user*, *User* dapat melakukan login dengan memasukkan *email* dan *password* dengan catatan jik *user* telah memiliki akun, dan bila belum memiliki akun maka user dapat melakukan registrasi dengan cara melakukan slide ke *menu new*

C. Registrasi user



Gambar 6 Halaman Registrasi User

Gambar 6 merupakan halaman registrasi *user*, *User* dapat melakukan registrasi dengan cara memasukkan nama, email, password dan melakukan konfirmasi terlebih dahulu jika password yang apakah sama atau tidak, dan setelah semua *data* telah terisi maka secara otomatis *user* akan dialihkan kedalam halaman utama user.

D. Halaman Dashboard User setelah login



Gambar 7 Halaman Utama Ketika User Telah Login

Gambar diatas merupakan halaman *dashboard user* setelah login dimana pada tampilan ini *user* dapat melihat informasi seperti renungan terbaru, ibadah yang telah di jalankan, lirik lagu rohani, dan motivasi harian,.

V. PENGUJIAN

A. Test Black Box

Implementasi aplikasi android dilakukan dengan menggunakan metode *black box testing*. Dimana *black box testing* merupakan pengujian yang terfokus kepada fungsional dari aplikasi, dan mencari hasil akhir dari *testing* tersebut.

1) Pengujian Login Admin

Berikut merupakan data dari pengujian yang telah di lakukan pada aplikasi yang dibuat.

No	Test case	Harapan	Keluaran	Hasil
1	Memasukan <i>email</i> dan tanpa memasukan <i>password</i>	Tidak dapat <i>login</i> dikarenakan field <i>password</i> kosong	Tidak dapat <i>login</i> dikarenakan field <i>password</i> kosong	Sesuai
2	Memasukan <i>password</i> dan tanpa memasukan <i>email</i>	Tidak dapat <i>login</i> dikarenakan field <i>email</i> kosong	Tidak dapat <i>login</i> dikarenakan field <i>email</i> kosong	Sesuai
3	memasukan email tanpa menggunakan symbol @	Tidak dapat login.	Tidak dapat login.	Sesuai
4	Memasukan <i>credential</i> yang tidak terdaftar	Tidak dapat <i>login</i> Karena akun tidak terdaftar	Tidak dapat <i>login</i> Karena akun tidak terdaftar	Sesuai
5	Memasukan <i>email</i> yang benar dan <i>password</i> yang benar	Berhasil <i>login</i>	Berhasil <i>login</i>	

				Sesuai
--	--	--	--	--------

VI. KESIMPULAN

A. Simpulan

Dari hasil pembahasan diatas, dapat disimpulkan bahwa aplikasi android misi *youth* adalah sebagai berikut :

1. Dalam aplikasi ini, terdapat fitur yang dapat memudahkan *admin* dalam melakukan pengelolaan dan menyimpan data dari mulai data renungan sampai data motivasi, dan User dapat menerima data dengan sangat jelas dan mudah.
2. Aplikasi ini merupakan aplikasi yang hanya bisa di akses oleh android saja.
3. User dapat menggunakan email diluar google.

B. Saran

Dari hasil kesimpulan, dapat disarankan jikalau aplikasi android misi *youth* di kedepan hari dapat diakses melalui *multi platform* bukan hanya android saja.

DAFTAR PUSTAKA

- [1] R. J. Keeley, *Menjadikan Anak-Anak Kita Bertumbuh dalam Iman*, Yogyakarta: Andi, 2009.
- [2] H. Aekkanopan, "HKBP Menyongsong Pelayanan Berbasis Digital Menghadapi Revolusi Industri 4.0," HKBP, 13 September 2019. [Online]. Available: <https://hkbpaekkanopan.or.id/2019/09/13/hkbp-menyongsong-pelayanan-berbasis-digital-menghadapi-revolusi-industri-4-0/>. [Diakses 17 Oktober 2020].
- [3] D. V. Makassar, "Penggunaan Perangkat Mobile Indonesia Diprediksi jadi ketiga terbesar di Dunia," Makassar Digital Valley, 12 December 2019. [Online]. Available: <https://makassardigitalvalley.id/pengguna-perangkat-mobile-indonesia-diprediksi-jadi-ketiga-terbesar-di-dunia/>. [Diakses 17 Oktober 2020].
- [4] CSO, "How to prevent your mobile app from getting hacked," CSO, 18 April 2017. [Online]. Available: <https://www.csoonline.com/article/3188880/how-to-prevent-your-mobile-app-from-getting-hacked.html>. [Diakses 17 Oktober 2020].
- [5] D. Engines, "System Properties Comparison Firebase Realtime Database vs. MySQL vs. SQLite," Db-Engines, [Online]. Available: <https://db-engines.com/en/system/Firebase+Realtime+Database%3BMySQL%3BSQLite>. [Diakses 17 Oktober 2020].
- [6] L. Hakim, *Membangun Web Berbasis PHP dengan framework Codeigniter*, Yogyakarta: Lokomedia, 2010.
- [7] P. Mainkar dan S. Giordano, *Google Flutter Mobile Development Quick Start Guide*, Birmingham-Mumbai: Packt, 2019.
- [8] S. Akopkoyants, *Master the art of programming high-performance applications with Dart*, Birmingham-Mumbai: Packt, 2014.
- [9] L. Moroney, *The Definitive Guide to Firebase*, Seattle, Washington, USA: Apress, 2017.
- [10] T. Arifianto, *Membuat Interface Aplikasi Android lebih keren dengan LWUIT*, Yogyakarta: Andi, 2011.
- [11] S. Hermawan, *Mudah Membuat Aplikasi Android*, Yogyakarta: Andi, 2011.
- [12] Wikipedia, "Android(Sistem Operasi)," Wikipedia, November 2018. [Online]. Available: [https://id.wikipedia.org/wiki/Android_\(sistem_operasi\)](https://id.wikipedia.org/wiki/Android_(sistem_operasi)). [Diakses 24 September 2020].