

# Pengenalan Google Firebase Untuk Hybrid Mobile Apps Berbasis Cordova

**Mirza Ilhami**

STMIK Mikroskil; Jln. Thamrin no. 124, 140, 112 Medan, +62 61 4567789

Jurusan Teknik Informatika, STMIK Mikroskil, Medan

e-mail: [me@mirzailhami.com](mailto:me@mirzailhami.com)

## **Abstract**

*Often developers have limitations when developing applications that are supported across multiple platforms. Due to, Android native and iOS natively have different programming languages. Java for Android, while Swift for iOS. Of course the developers must master the two programming languages to build applications in both. This is certainly not efficient in terms of time. Therefore, the existence of hybrid apps technology saves developers time, because hybrid app can work on multiplatform. The app developer simply builds one app and then runs it on the Android and iOS platform. Another constraint that app developers often encounter is when developing REST APIs for built applications. Need to master a server-based programming language such as PHP or ASP to be able to build a good API. Coupled with building relational databases such as MySQL or PostgreSQL to store data from built applications. This research implements how hybrid apps can run on multiple platforms and utilize Google Firebase as a database without having to use a relational database such as SQL. Google Firebase provides REST APIs that can be accessed directly through hybrid apps without the need to build APIs using server-based programming. Tests are performed with some features of Google Firebase such as Authentication, Realtime Database, Storage and Cloud Messaging.*

**keywords**—*hybrid apps, google firebase, android, ios*

## **Abstrak**

*Seringkali para pengembang memiliki keterbatasan ketika mengembangkan aplikasi yang didukung di beberapa platform. Dikarenakan, Android native dan iOS native memiliki bahasa pemrograman yang berbeda. Java untuk Android, sedangkan Swift untuk iOS. Tentu para pengembang harus menguasai dua bahasa pemrograman tersebut untuk membangun aplikasi dikeduanya. Ini tentu tidak efisien dari segi waktu. Karena itu, adanya teknologi hybrid apps sangat menghemat waktu para pengembang, karena hybrid app dapat bekerja di multiplatform. Pengembang aplikasi cukup membangun satu aplikasi dan kemudian menjalankannya di platform Android and iOS. Kendala lainnya yang sering dijumpai pengembang aplikasi adalah ketika mengembangkan REST API untuk aplikasi yang dibangun. Perlu menguasai bahasa pemrograman berbasis server seperti PHP atau ASP agar mampu membangun API yang baik. Ditambah lagi dengan membangun database relasional seperti MySQL atau PostgreSQL untuk menyimpan data dari aplikasi yang dibangun. Penelitian ini*

*mengimplementasikan bagaimana hybrid apps dapat berjalan di multi platform dan memanfaatkan Google Firebase sebagai database tanpa harus menggunakan database relasional seperti SQL. Google Firebase menyediakan REST API yang dapat diakses langsung melalui hybrid apps tanpa perlu membangun API menggunakan pemrograman berbasis server. Ujicoba dilakukan dengan beberapa fitur dari Google Firebase seperti Autentikasi, Realtime Database, Storage dan Cloud Messaging.*

**Kata kunci**—*hybrid apps, google firebase, android, ios*

## 1. PENDAHULUAN

Pasar aplikasi *mobile* saat ini berjumlah lebih dari 2 juta aplikasi, diunduh bermiliar kali setiap tahun dari *app store* (seperti Play Store dan App Store) [1]. Berdasarkan penelitian yang dilakukan Google pada 1 januari – 31 maret 2016 terhadap 11,964 pengguna Google dengan lintas perangkat, berusia 18 – 49 tahun di Amerika Serikat, didapatkan fakta bahwa 80% dari jumlah orang tersebut menggunakan perangkat *smartphone*, 67% menggunakan perangkat komputer dan 16% menggunakan perangkat tablet di setiap harinya [2]. Bahkan setiap individu saat ini bisa memiliki lebih dari 1 perangkat *smartphone* di saku mereka. Tentunya jumlah pengguna *smartphone* akan semakin bertambah kedepannya. Semakin pesatnya jumlah pengguna *smartphone* ini, tentu sangat mempengaruhi perkembangan teknologi dibidang aplikasi *mobile* pula. Namun, para pengembang memiliki kendala ketika ingin mengembangkan aplikasi untuk beberapa *platform* sekaligus (seperti Java untuk Android dan Swift untuk iOS), selain itu juga tantangan dan pemeliharaan menjadi faktor kendala utama . Sebagai solusi yang mungkin, perusahaan-perusahaan dan banyak pengembang aplikasi mulai menginvestasikan waktu dan tenaganya untuk mulai mempelajari aplikasi hybrid. Aplikasi *hybrid* dibangun menggunakan teknologi standar web seperti HTML, CSS, JavaScript [3]. Dan semua *service request* ke server dilakukan dari sisi JavaScript. Dalam konteks ini, dapat dikatakan bahwa framework aplikasi hybrid (seperti Apache Cordova) memungkinkan pengembang aplikasi untuk membuat aplikasi *mobile* lintas platform berbasis web dengan menyediakan fitur-fitur *native* dan menjembatani semua *service request* dari kode berbasis web ke *platform* API yang sesuai [4]. Walaupun aplikasi *hybrid* memiliki kelebihan, namun ia juga masih memiliki banyak kekurangan seperti keterbatasan akses ke fitur *hardware*, penurunan kinerja dan lainnya [6]. Perkembangan

Apache Cordova ini melahirkan banyak framework aplikasi *hybrid* seperti: Ionic Framework [5], Sencha, Kendo UI, Onsen UI dan lainnya.

Untuk dapat membangun sebuah aplikasi, maka dibutuhkan media penyimpanan berupa database yang dapat diakses dari sisi server menggunakan bahasa pemrograman berbasis server seperti PHP dan ASP. Aplikasi *hybrid* akan mengirimkan *request* ke server yang kemudian diteruskan ke *database* untuk melakukan *create*, *read*, *update* atau *delete* data. Tentunya ini juga akan menjadi tantangan teknis bagi pengembang karena mereka perlu menyediakan REST API yang dapat diakses dengan baik oleh aplikasi yang akan dibangun. Dari segi waktu dan sumber daya tentu ini akan menghabiskan banyak usaha. Belum lagi para pengembang juga perlu memastikan keamanan dari API tersebut serta bagaimana sistem autentikasi bekerja. Hadirnya Google Firebase sangat menjawab tantangan tersebut, karena Google Firebase telah menyediakan *realtime database* NoSQL dengan struktur data JSON (JavaScript Notation) yang dapat dengan mudah diakses melalui kode web di aplikasi *hybrid*. Selain *realtime database*, ia juga menyediakan fitur autentikasi pengguna (menggunakan email dan password), *storage* (sebagai ruang penyimpanan file) dan *cloud messaging* (dapat digunakan untuk mengirim notifikasi). Dengan kata lain, hadirnya Google Firebase telah menghemat waktu para pengembang ketika ingin mengembangkan aplikasi.

Fokus utama penelitian ini adalah ujicoba Google Firebase terhadap aplikasi *hybrid* pada perangkat Android dengan memanfaatkan beberapa fitur dari Google Firebase seperti: Autentikasi, Realtime Database, Storage dan Cloud Messaging. Google Firebase menyediakan fitur autentikasi yang memungkinkan pengguna melakukan login menggunakan email, akun facebook atau akun google. *Realtime database* dapat digunakan oleh pengembang untuk membuat aplikasi yang bersifat *realtime* seperti aplikasi chat, tentu saja Realtime Database ini dapat pula digunakan untuk membuat aplikasi yang tidak perlu bersifat *realtime* seperti *e-commerce*. Google Firebase juga menyediakan Storage (kuota gratis hingga 5GB) untuk menyimpan file dan dapat disinkronkan kedalam aplikasi. Terakhir, fitur yang wajib dimiliki oleh aplikasi adalah Cloud Messaging. Firebase Cloud Messaging memungkinkan aplikasi untuk mengirimkan notifikasi kepada pengguna, seperti berupa pemberitahuan info tertentu, promo produk

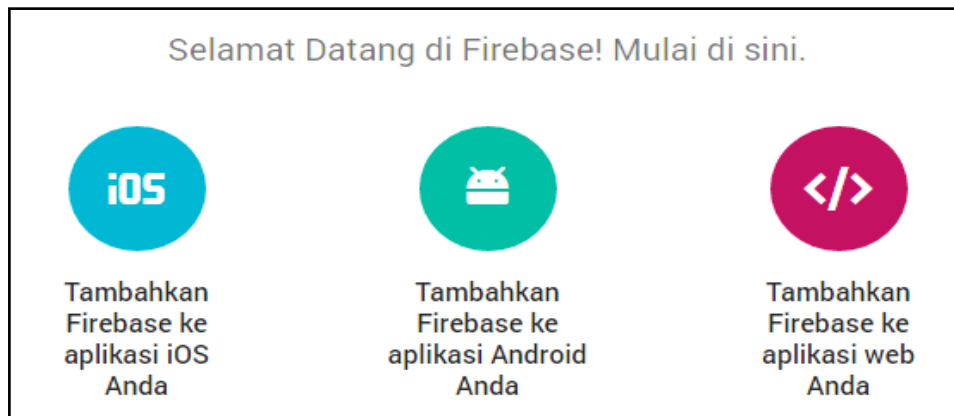
dan diskon (aplikasi *e-commerce*), notifikasi komentar (aplikasi sosial media) atau lainnya.

## 2. METODE PENELITIAN

Dalam penelitian ini, akan diimplementasikan penggunaan beberapa fitur Google Firebase pada hybrid apps. Framework yang akan digunakan adalah Ionic Framework. Ionic memungkinkan pengembang untuk membangun aplikasi hybrid menggunakan HTML, CSS dan JavaScript (AngularJS) yang merupakan teknologi standar web. Selama pengembangan di desktop, aplikasi *hybrid* dapat dijalankan melalui browser di desktop.

### 2.1 Memulai Google Firebase

Semua fitur Google Firebase dapat diakses melalui halaman konsol Firebase dengan alamat <https://console.firebase.google.com>. Firebase sendiri memungkinkan untuk digunakan pada platform iOS, Android dan Web. Karena penelitian ini menggunakan aplikasi hybrid berbasis web, maka implementasi dilakukan pada *platform* berbasis web.



**Gambar 1.** Memulai Google Firebase

Gambar di atas adalah tahapan awal memulai Google Firebase untuk aplikasi *hybrid*, yaitu memilih “Tambahkan Firebase ke aplikasi web Anda”. Untuk menambahkan Firebase ke aplikasi berbasis web, yang perlu dilakukan adalah menambahkan cuplikan kode seperti di bawah ini ke HTML.

```
// TODO: Replace with your project's customized code snippet
<script src="https://www.gstatic.com/firebasejs/3.4.0/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: '<your-api-key>',
    authDomain: '<your-auth-domain>',
    databaseURL: '<your-database-url>',
    storageBucket: '<your-storage-bucket>'
  };
  firebase.initializeApp(config);
</script>
```

Cuplikan kode di atas berisi informasi inisialisasi untuk mengkonfigurasi Firebase JavaScript SDK agar dapat menggunakan Autentikasi, Storage dan Realtime Database.

## 2.2 Firebase Authentication

Sebagian besar aplikasi tentu membutuhkan autentikasi untuk mengetahui identitas pengguna. Mengetahui identitas pengguna memungkinkan aplikasi untuk secara aman menyimpan data pengguna di *cloud* dan memberikan pengalaman pribadi yang sama di semua perangkat pengguna. Firebase menyediakan layanan backend, SDK yang mudah digunakan dan pustaka siap pakai untuk mengautentikasi pengguna ke aplikasi. Saat ini Firebase mendukung autentikasi menggunakan sandi, sosial media populer seperti Google, Facebook, Twitter dan lainnya. Ujicoba autentikasi yang dilakukan pada penelitian ini menggunakan email dan kata sandi.

### a. Registrasi

Ketika pengguna pertama kali mengakses aplikasi, maka akan ditampilkan formulir registrasi sederhana dengan meminta pengguna tersebut memasukkan email dan kata sandi. Secara otomatis setelah proses registrasi selesai, maka pengguna tersebut masuk ke aplikasi.

### b. Login

Pengguna yang telah berhasil melakukan registrasi, maka ia bisa login ke aplikasi menggunakan email dan kata sandi tersebut.

### c. Logout

Logout atau keluar dari aplikasi.

Selain fitur di atas tersebut, Google Firebase Authentication juga menyediakan fitur untuk mengelola pengguna seperti ubah kata sandi, ubah email dan verifikasi email.

### 2.3 Firebase Realtime Database

Firebase Realtime Database merupakan *cloud database*. Data disimpan dalam format JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika membangun aplikasi *hybrid* lintas platform, seperti Android dan iOS maka semua klien berbagi satu *instance* Realtime Database dan secara otomatis menerima pembaruan dengan data tertentu. Firebase Realtime Database adalah basis data NoSQL dan karena itu memiliki optimalisasi dan fungsionalitas yang berbeda dibandingkan dengan basis data relasional. Membuat database Firebase bisa melalui import file JSON ke konsol Firebase, atau dapat juga dibuat langsung melalui halaman konsol Realtime Database secara manual.

### 2.4 Firebase Storage

Firebase Storage digunakan untuk menyimpan data seperti gambar, audio dan video. Sebagian aplikasi pasti menggunakan file gambar, audio atau video. Adanya Firebase Storage sangat memudahkan proses unggahan dan unduhan untuk aplikasi. Beberapa kelebihan utama dari Firebase Storage adalah sebagai berikut:

a. Strong

Firebase Storage dapat melakukan unggahan dan unduhan file di semua kualitas jaringan internet. Ia dapat berhenti atau melanjutkan, sehingga menghemat waktu dan *bandwidth* pengguna.

b. Secure

Firebase Storage terintegrasi dengan Firebase Authentication untuk menyediakan autentikasi yang mudah dan intuitif untuk pengembang. Model keamanan dapat diatur berdasarkan nama file, ukuran, tipe konten dan metadata lainnya.

c. Scalable

Firebase Storage didukung oleh Google Cloud Storage hingga skala petabyte. Ini sangat penting, ketika aplikasi yang dari prototipe kemudian berkembang viral dengan jutaan pengguna.

Proses unggahan ke *storage* dapat dilakukan melalui aplikasi pengguna menggunakan AngularJS. Ini lebih mudah diimplementasikan karena tidak perlu menggunakan pemrograman berbasis server seperti PHP atau ASP. Ketika file telah berhasil diunggah ke storage, maka akan mengembalikan link unduhan (*download link*) yang dapat disimpan ke Firebase Database.

## 2.5 Firebase Cloud Messaging

Firebase Cloud Messaging merupakan solusi perpesanan lintas *platform* yang memungkinkan aplikasi mengirimkan pesan dan pemberitahuan ke aplikasi pengguna dengan tanpa biaya (gratis). Ada Firebase Cloud Messaging, aplikasi bisa memberitahu aplikasi pengguna ketika ada email baru atau pemberitahuan dan notifikasi. Ini cukup bagus untuk meningkatkan retensi pengguna dan mendorong pengguna aplikasi agar kembali mengakses aplikasi. FCM juga menyediakan konsol notifikasi dengan alamat: <https://firebase.google.com/docs/notifications> yang dapat digunakan untuk mengirimkan pesan ke aplikasi pengguna. Gambar di bawah ini menunjukkan bagaimana arsitektur dari FCM ini.



**Gambar 2.** Firebase Cloud Messaging

Pesan notifikasi bisa dikirimkan ke pengguna melalui Notifications Console GUI dan HTTP request seperti terlihat pada gambar di atas.

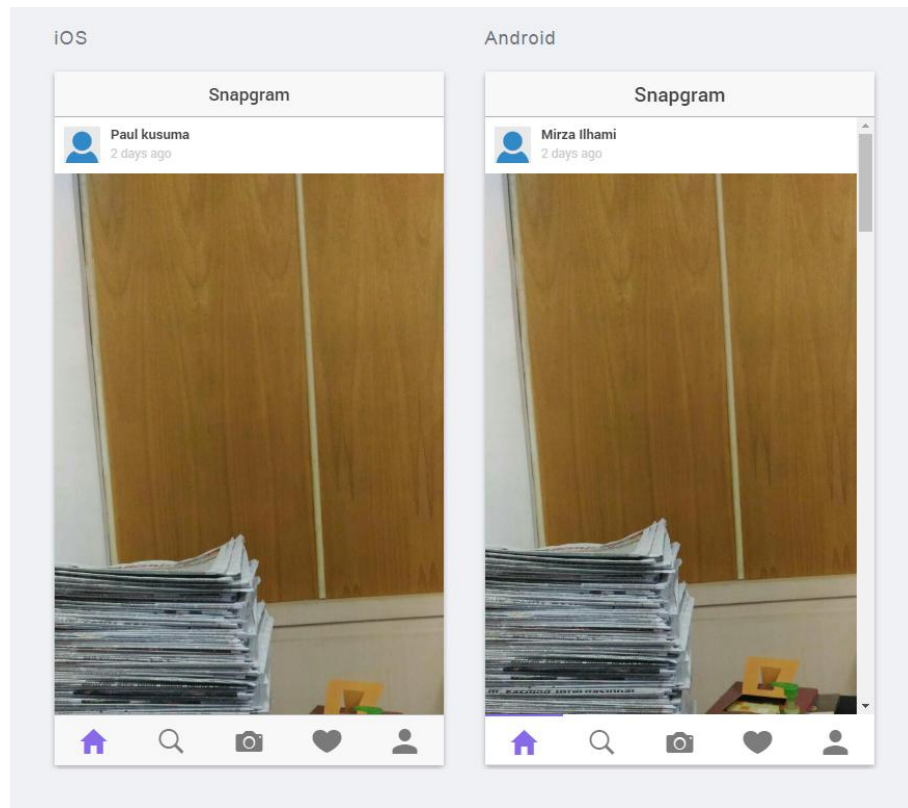
## 2.6 Implementasi Google Firebase ke Aplikasi

Aplikasi hybrid yang digunakan pada penelitian ini adalah Ionic Framework dengan menggunakan *browser* desktop dan perangkat Android untuk ujicobanya. Tahapan implementasi Google Firebase ke aplikasi adalah sebagai berikut:

- a. Instalasi npm dan Ionic Framework di desktop melalui Command Line Interface (CLI)
- b. Mendesain aplikasi menggunakan HTML dan CSS
- c. Memasukkan kode librari Firebase ke index.html
- d. Tampilkan aplikasi di browser desktop
- e. Build aplikasi di perangkat Android di Windows

Sebelum aplikasi dijalankan pada perangkat Android, pengembangan aplikasi dilakukan pada *browser* desktop dengan menggunakan aplikasi text editor sebagai tool untuk menulis kode program. Hasil dari aplikasi *hybrid* di *browser* Chrome adalah seperti berikut.





**Gambar 3.** Tampilan aplikasi di browser Chrome

Tampilan di atas merupakan hasil dari aplikasi *hybrid* yang dibangun dengan Ionic Framework di browser Chrome desktop. *Tool* lain yang digunakan untuk membantu dalam proses debugging adalah Chrome Inspect Element. *Tool* ini cukup penting untuk mengetahui ketika ada masalah atau kode program yang salah.

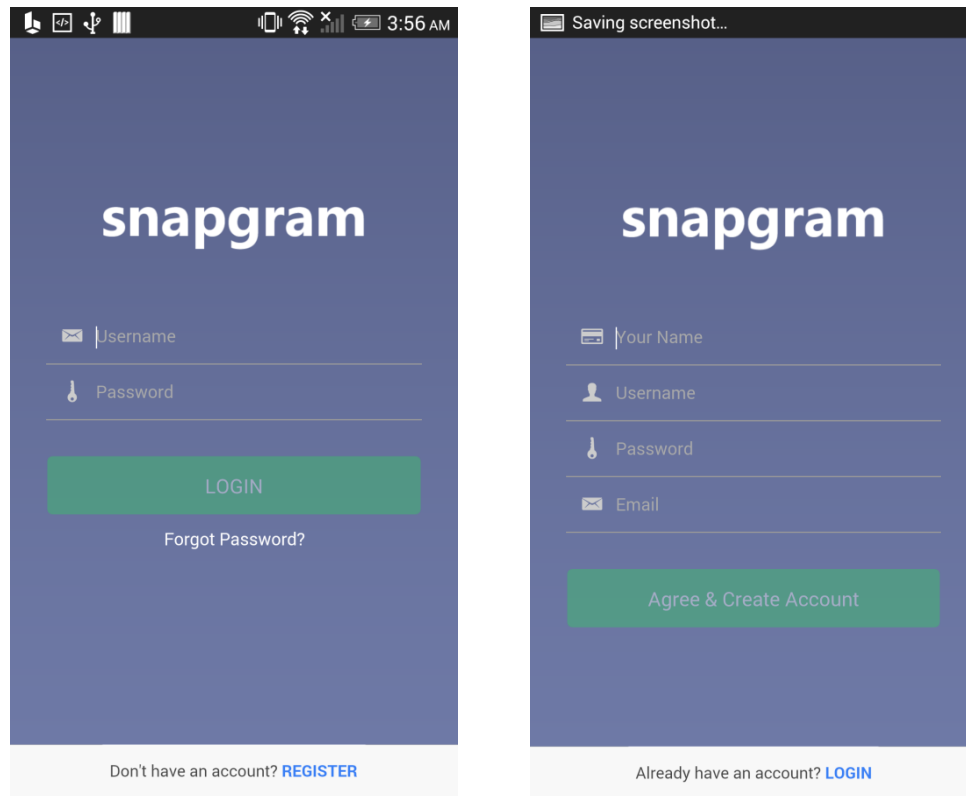
### 3. HASIL DAN PEMBAHASAN

Hasil implementasi beberapa fitur Google Firebase pada *hybrid* apps adalah berupa prototipe aplikasi Android yang memiliki fitur standar seperti: login, registrasi, tambah data, unggah file serta kirim dan terima notifikasi ke pengguna.

#### 3.1 Login dan registrasi

Fitur registrasi dan login sudah sangat umum ada disetiap aplikasi, maka pada penelitian ini dilakukan ujicoba bagaimana Google Firebase menangani registrasi dan

login menggunakan Firebase Authentication. Form registrasi dan login yang dihasilkan pada perangkat Android adalah sebagai berikut.

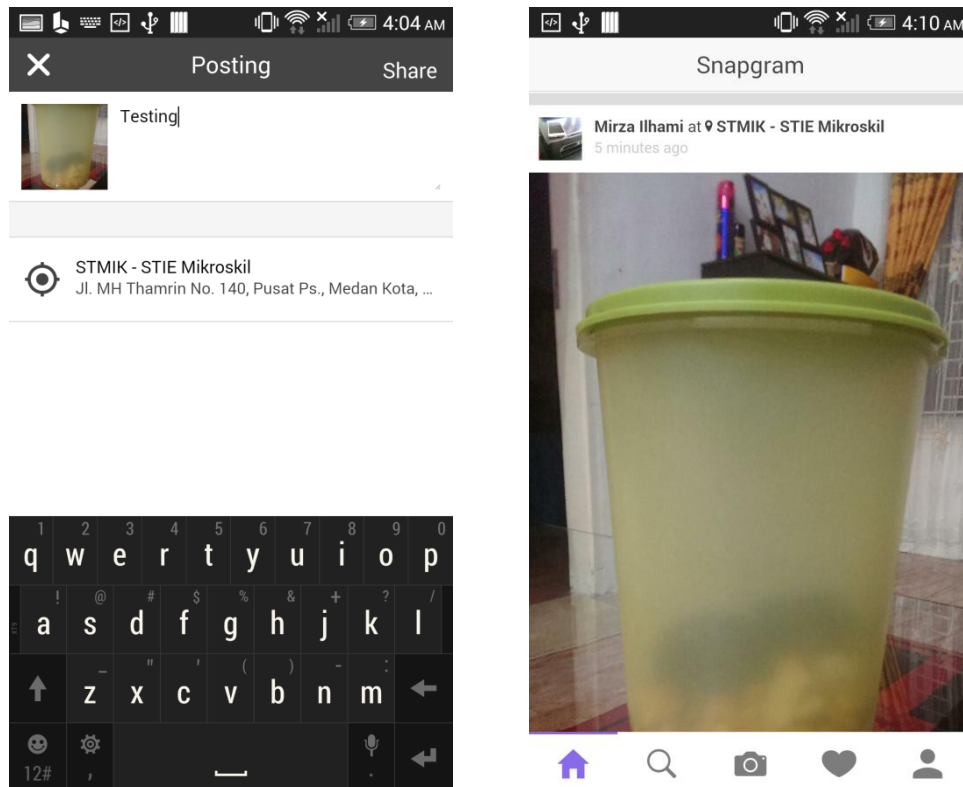


**Gambar 4.** Tampilan login dan registrasi di Android

Pada gambar registrasi di atas, pengguna diminta untuk mengisi email, username dan password yang kemudian data tersebut akan disimpan ke Firebase Authentication. Sedangkan gambar login di atas adalah menampilkan form login yang meminta pengguna untuk memasukkan username dan password.

### 3.2 Unggah berkas

Untuk melakukan ujicoba bagaimana Firebase Storage menangani proses unggahan file, maka pada penelitian ini ditambahkan fitur untuk menginput status (seperti pada Instagram), berupa file gambar dan caption serta tag lokasi.

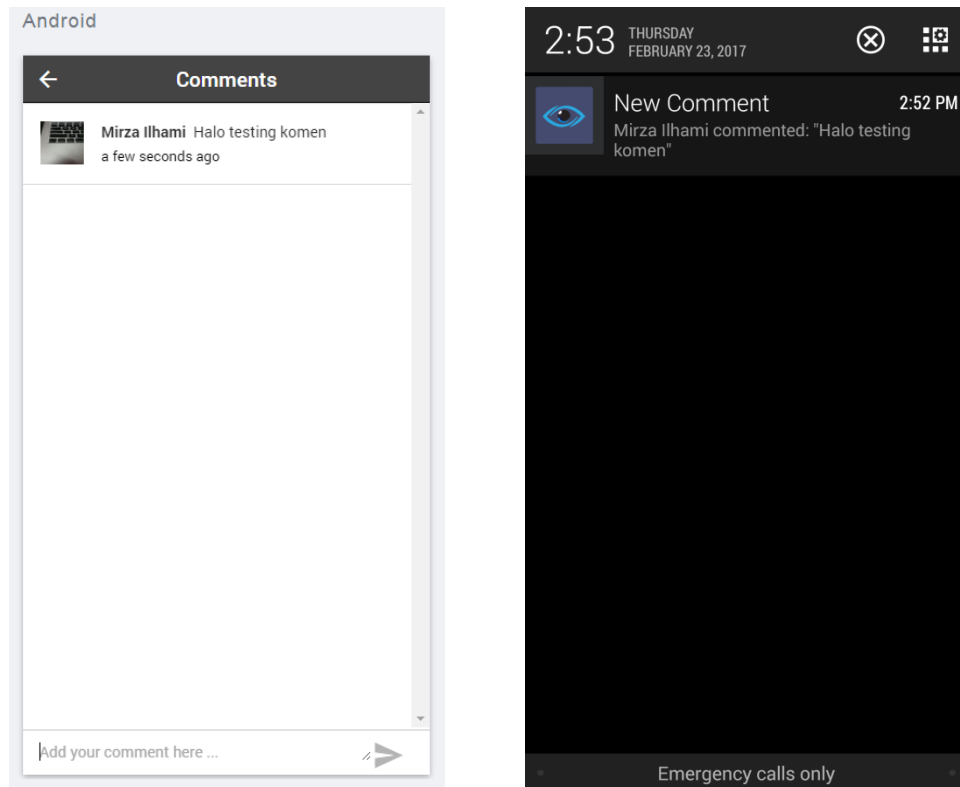


**Gambar 5.** Tampilan posting status dan home di Android

Gambar sebelah kiri di atas merupakan tampilan ketika pengguna ingin melakukan posting berupa unggah file foto dari kamera/galeri, menuliskan caption dan tag lokasi. Sedangkan gambar sebelah kanan di atas menampilkan hasil dari postingan pengguna.

### 3.3 Kirim dan terima notifikasi

Pengujian kirim dan terima notifikasi dilakukan dengan menggunakan Firebase Cloud xMessaging. Dalam penelitian ini, notifikasi akan dikirimkan ketika ada yang melakukan like atau komen terhadap status postingan pengguna. Perangkat yang digunakan untuk ujicoba adalah Android. Disediakan dua akun untuk mengujicoba Firebase Cloud Messaging, yaitu akun pengguna yang melakukan komen dan akun pengguna yang menerima komen. Hasilnya adalah seperti berikut.



**Gambar 6.** Tampilan tambah komentar dan terima notifikasi

Gambar kiri di atas menampilkan ketika salah satu pengguna melakukan komentar di status pengguna yang lain melalui *browser* Chrome, yang kemudian komentar disebut menjadi notifikasi seperti tampil pada gambar sebelah kanan di aplikasi Android pengguna.

#### 4. KESIMPULAN

Penelitian ini masih berupa pengenalan terhadap teknologi aplikasi *hybrid* berbasis Cordova dan beberapa fitur-fitur dari Google Firebase. Berdasarkan hasil penelitian yang telah dilakukan dalam menghasilkan prototipe aplikasi *hybrid* dengan Google Firebase dapat disimpulkan bahwa:

1. Aplikasi hybrid lebih mudah dikembangkan karena tidak perlu menggunakan aplikasi besar seperti Xcode atau Android Studio, cukup menggunakan text editor biasa dan bahasa standard web seperti HTML, CSS dan JavaScript
2. Firebase sangat cocok digunakan dengan aplikasi hybrid karena memiliki fitur-fitur yang sangat mendukung, seperti: Cloud Messaging, Authentication, Realtime Database, Storage dan lainnya
3. Tersedia beberapa plugin dari Cordova untuk akses terhadap hardware dari perangkat Android dan iOS, seperti: Camera, Geolocation, File dan lainnya
4. Kekurangan dari aplikasi hybrid adalah terkait dengan performansi.

Untuk penelitian lanjutan, penulis berencana untuk melanjutkan penelitian yang lebih detil terkait dengan aplikasi *hybrid* berbasis Cordova dan Google Firebase ini, seperti membandingkan performansi aplikasi hybrid pada beberapa platform, mengembangkan aplikasi marketplace untuk UMKM dan menulis buku lanjutan tentang aplikasi hybrid.

## 5. SARAN

Penelitian lebih lanjut terkait dengan prototipe yang dihasilkan dari penelitian ini tentu masih terbuka luas. Penelitian ini masih bisa dilanjutkan dengan menyempurnakan prototipe yang ada, memperkaya fitur dan mempercantik *user interface*. Riset lain yang bisa dilakukan dari penelitian ini adalah mencari tahu dan membandingkan bagaimana pengalaman pengguna selama menggunakan aplikasi hybrid berbasis Cordova ini dengan aplikasi sejenis yang dibangun dengan native programming (seperti Java atau Swift).

## UCAPAN TERIMA KASIH

Selama penelitian ini tidak lupa penulis mengucapkan terima kasih kepada pak Roni Yunis, S.Kom, MT yang telah memberikan masukan dan kritiknya. Kemudian juga terima kasih saya kepada Pusat Penelitian dan Pengabdian Masyarakat (P3M) STMIK Mikroskil dan panitia NERIS yang telah banyak membantu terbitnya makalah ini.

#### DAFTAR PUSTAKA

- [1] I. Malavolta, S. Ruberto, T. Soru and Terragni, "End Users' Perception of Hybrid Mobile Apps," in *IEEE International Conference on Mobile Services*, 2015.
- [2] Google, "How People Use Their Devices," September 2016. [Online]. Available: <https://storage.googleapis.com/think/docs/twg-how-people-use-their-devices-2016.pdf>. [Accessed 17 February 2017].
- [3] I. Malavolta, S. Ruberto, T. Soru and V. Terragni, "Hybrid Mobile Apps in the Google Play Store: An Exploratory Investigation," in *International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2015.
- [4] A. Khandeparkar, R. Gupta and B. Sindhya, "An Introduction to Hybrid Platform Mobile," *International Journal of Computer Applications*, vol. 118, p. 15, 2015.
- [5] Anditya and M. Ilhami, *Jago Bikin Aplikasi Smartphone*, Yogyakarta: Andi Offset, 2015.
- [6] L. Corral, A. Sillitti and G. Succi, "Mobile multiplatform development: An experiment for," *Procedia Computer Science*, vol. 10, pp. 736-743, 2012.