

## BAB 3 PERANCANGAN DAN IMPLEMENTASI

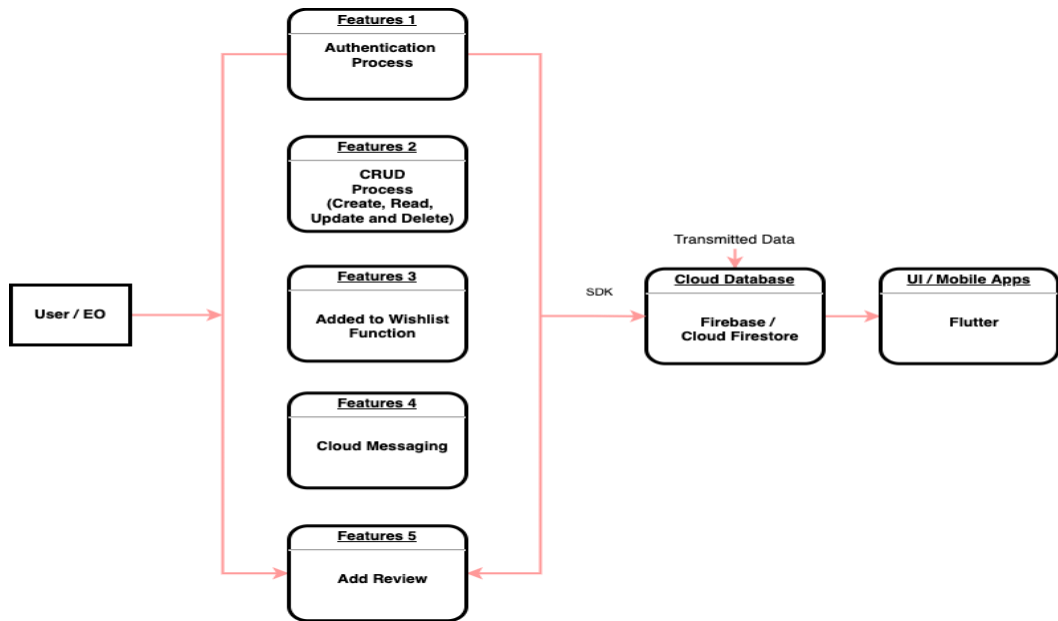
### 3.1 Arsitektur Sistem

Sistem aplikasi *Event Organizer* dapat berjalan ketika user melakukan registrasi terlebih dahulu sehingga user yang belum memiliki akun pada aplikasi diharuskan untuk mendaftar terlebih dahulu. Aplikasi ini ditujukan untuk dua user, yaitu yang pertama *Event Organizer* sebagai penyedia jasa dan *user* yang akan mencari jasa *Event Organizer*. Namun, kedua user tersebut memiliki tampilan User Interface (UI) yang sama. Oleh karena itu, setiap user yang mendaftar ke dalam aplikasi dapat menambahkan jasa nya untuk memasarkann jasa nya atau hanya unttuk keperluan mencari Event Organizer saja.

Fitur utama dari aplikasi ini yaitu fitur CRUD (Create, Read, Update, dan Delete). Fitur CRUD akan dibuat dengan menggunakan *Firestore* dari *Firebase* sebagai *real-time* database yang akan menyimpan seluruh data yang akan ditampilkan di dalam aplikasi. Tujuan dari dibuatnya fitur CRUD ini yaitu untuk memfasilitasi keperluan para *Event Organizer* untuk dapat menambahkan jasa mereka pada aplikasi agar dapat memasarkan jasa mereka dengan mudah. Pada fitur CRUD ini, pemilik jasa *Event Organizer* dapat menambahkan detail-detail jasa mereka seperti, deskripsi dan portofolio dari *event-event* yang telah dilaksanakan, sehingga user dapat dengan mudah mencari *Event Organizer* yang sesuai dengan keinginan mereka.

Fitur selanjutnya yaitu fitur filter. Tujuan lain dari dibuatnya aplikasi ini yaitu untuk memudahkan user dalam mencari *Event Organizer*. Oleh karena itu, fitur filter dibuat dengan tujuan agar user dapat dengan mudah mencari *Event Organizer* berdasarkan daerah dan juga rating dari suatu *Event Organizer*. Selain itu user juga dapat mengirim pesan kepada *Event Organizer* agar memudahkan user dalam berkomunikasi dengan pemilik jasa *Event Organizer*. Arsitektur sitem pada pengembangan aplikasi dapat dilihat pada Gambar 3.1.





Gambar 3.1 Analisis Diagram Aplikasi

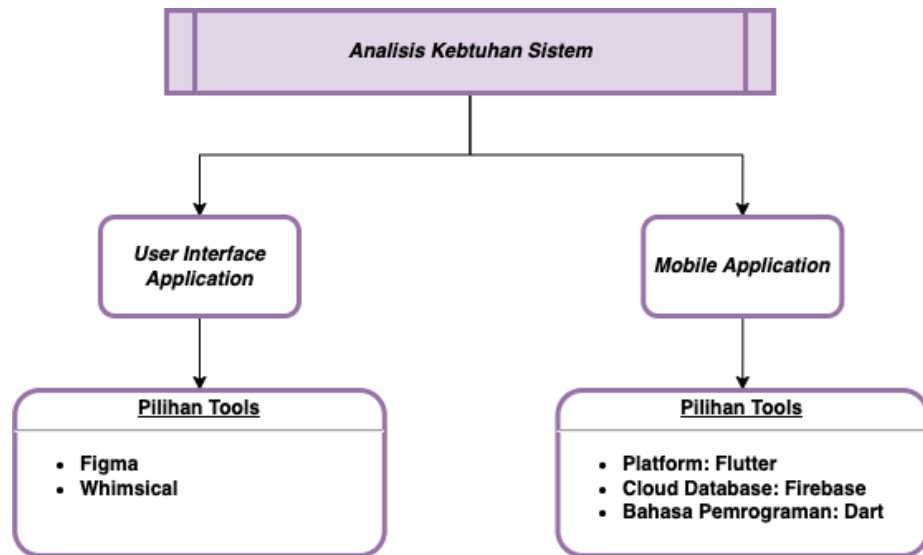
### 3.2 Perancangan Sistem

Perancangan sistem dimulai dengan melakukan analisis terlebih dahulu terhadap kebutuhan sistem. Perancangan sistem pada aplikasi *marketplace Event Organizer* berbasis mobile ini akan dilakukan dengan beberapa tools utama seperti *Framework Flutter* dan *Firebase*.

#### 3.2.1 Analisis Kebutuhan Sistem

Awal perancangan sistem dibutuhkan analisis kebutuhan untuk memenuhi setiap material yang dibutuhkan dalam proses perancangan, Gambar 3.2 di bawah ini akan menjelaskan secara ringkas jenis pilihan material yang akan digunakan.





Gambar 3.2 Analisis Kebutuhan Sistem

Dari analisis kebutuhan sistem tersebut maka dimulai dari pemilihan material desain untuk *software* guna proses perancangan.

### 3.2.1 Perancangan Perangkat Lunak Sistem

Pada penelitian ini, pengembangan seluruh sistem aplikasi marketplace *Event Organizer* menggunakan framework *Flutter* dan *Firebase*. Framework *Flutter* utama-nya digunakan untuk membangun keseluruhan *UI (User Interface)* aplikasi dan juga mengintegrasikan seluruh sistem dengan *Cloud Database* yang digunakan yaitu *Firebase*. *Firebase Realtime Database* adalah database *NoSQL* yang digunakan untuk menyimpan, membaca, menulis dan mengambil data secara *Realtime* yang disinkronkan kepada klien yang terhubung.

Di dalam aplikasi ini, user dapat berperan sebagai user yang dapat melihat daftar *Event Organizer* yang tersedia dalam aplikasi dan juga sebagai *Event Organizer*, yang dapat mengelola data yang ada dalam aplikasi. Pada penelitian ini akan mendukung fitur *CRUD* (*Create, Read, Update dan Delete*), dimana *Event Organizer* dapat menyimpan, membaca, menulis dan mengambil data secara *realtime*.

*Event Organizer* dapat menyimpan inputannya pada Database. Data yang akan di inputkan berupa data-data yang diperlukan *Event Organizer*, seperti kategori, deskripsi, *founder*, logo atau gambar *Event Organizer*, *location*, nama *Event Organizer*, dan portofolio, serta *rating*. Platform perancangan dashboard sistem dapat dilihat pada Gambar 3.3.

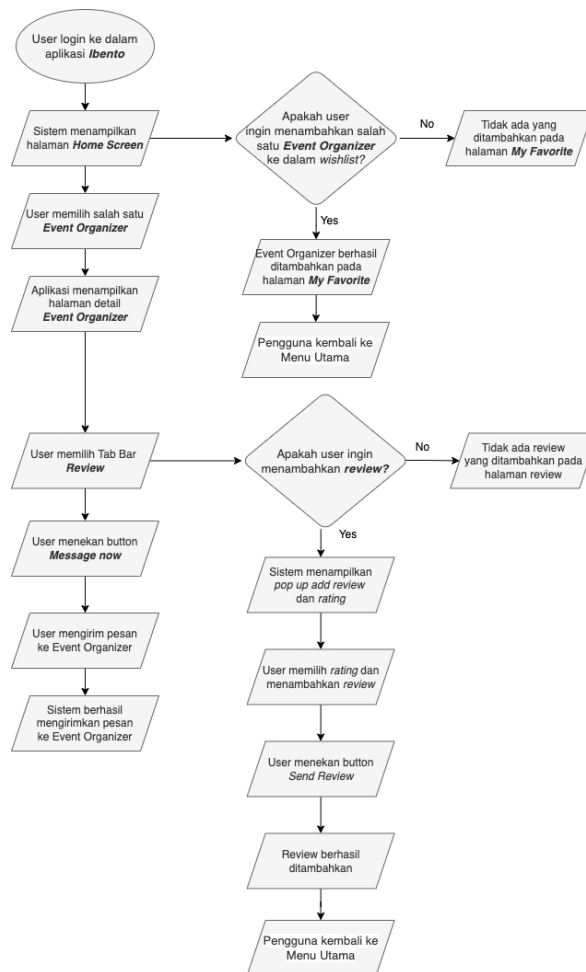




Gambar 3.3 Platform perancangan dashboard sistem

### 3.3 Perancangan dan Implementasi Perangkat Lunak Sistem

Pada bagian ini membahas tentang perancangan sistem keseluruhan yang meliputi perancangan perangkat lunak (*software*). Dalam perancangan ini diimplementasikan konsep dasar yang dibahas sebelumnya. Flowchart cara kerja dari sisi user aplikasi marketplace Event Organizer berbasis mobile ini dapat dilihat pada Gambar 3.4. Sementara, flowchart cara kerja dari sisi *event organizer* aplikasi marketplace Event Organizer berbasis mobile ini dapat dilihat pada Gambar 3.5.



Gambar 3.4 Flowchart Cara Kerja (Sisi user)





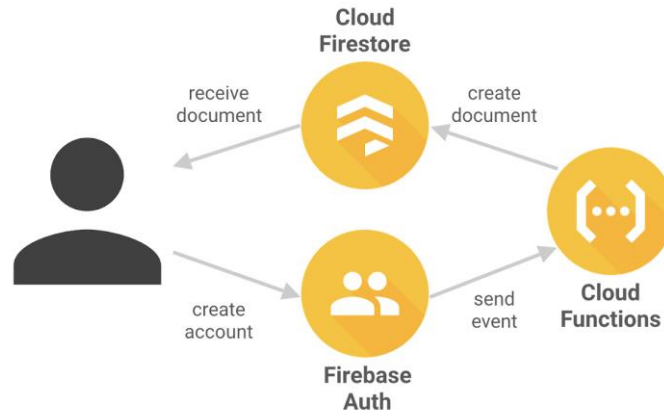


Gambar 3.5 Flowchart Cara Kerja (Sisi EO)

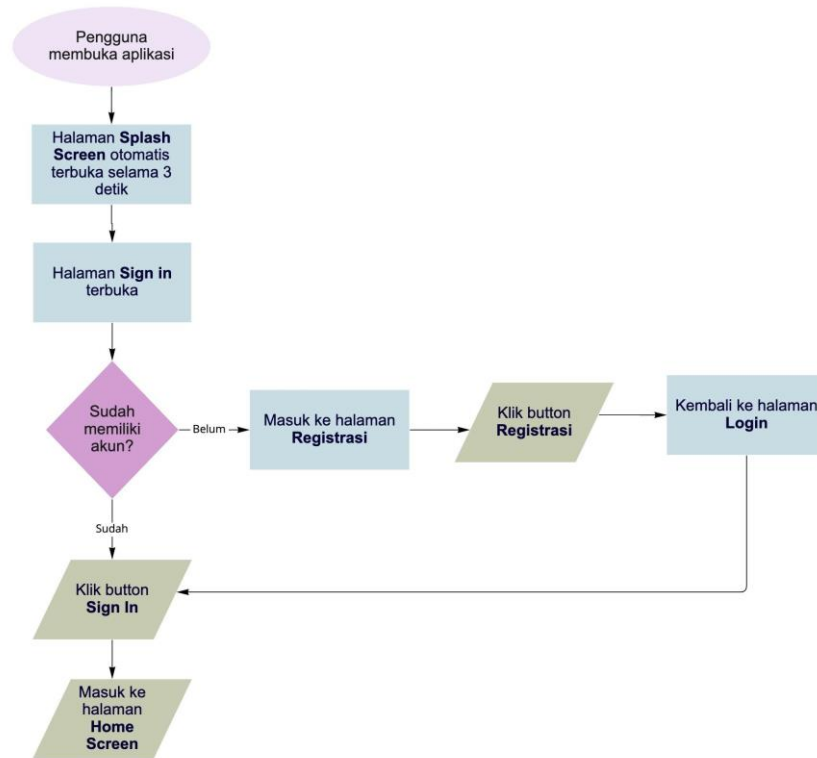
### 3.3.1 Proses Interaksi Sistem pada Fitur *Authorization*

Fitur *Authorization* pada aplikasi marketplace *Event Organizer* berbasis *mobile* ini merupakan fitur utama yang sangat penting. User tidak akan dapat masuk ke dalam aplikasi jika user belum mendaftar atau belum memiliki akun pengguna pada aplikasi *Ibento*. Pertama-tama aplikasi akan masuk ke halaman *Splash Screen* terlebih dahulu, lalu otomatis akan masuk ke halaman *Login / Sign In*. Proses interaksi sistem pada fitur *authorization* ini akan ditunjukkan melalui diagram pada Gambar 3.6. Fitur *authorization* ini menggunakan *firebase* sebagai layanan *back-end* untuk mendukung proses *authorization*, gambaran dari proses *firebase* dapat dilihat pada Gambar 3.7.





Gambar 3.6 Proses Firebase Authentication



miro

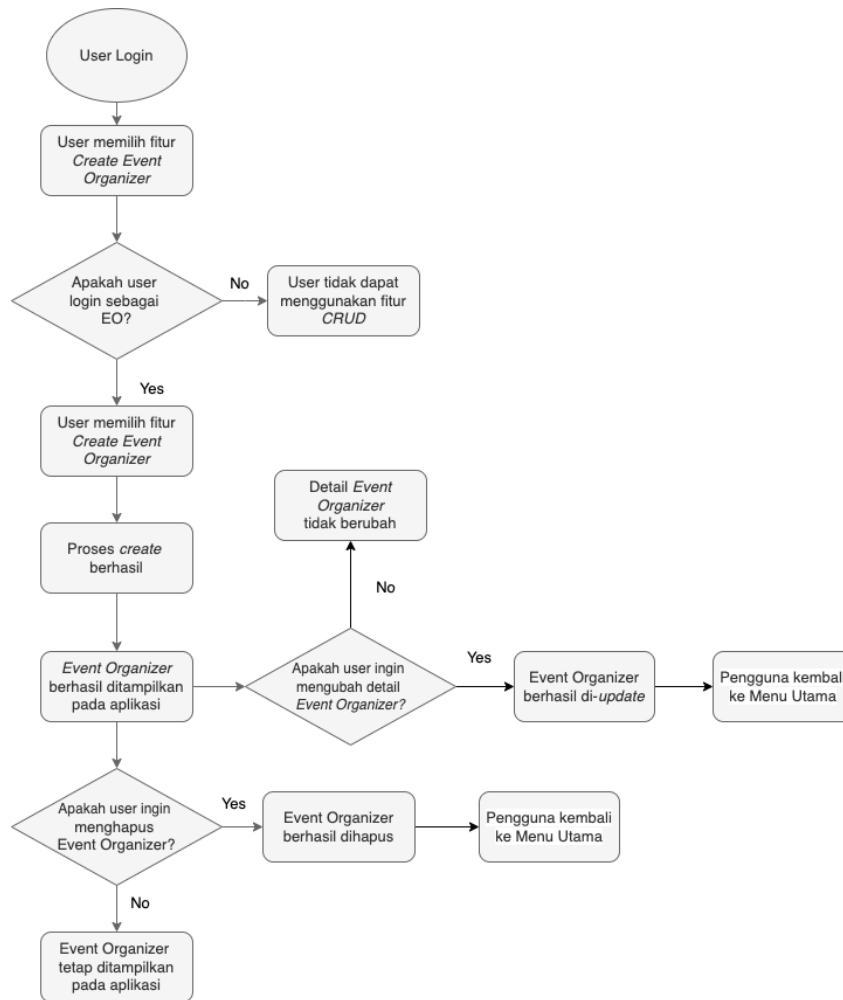
Gambar 3.7 Flowchart Proses Login dan Register

### 3.3.2 Proses Interaksi Sistem pada Fitur CRUD (Create, Read, Update dan Delete)

Pada aplikasi marketplace *Event Organizer* berbasis *mobile* menyediakan fitur *CRUD* (*Create, Read, Update, dan Delete*). Fitur ini disediakan untuk memudahkan pemilik *Event Organizer* dalam menambahkan jasa yang dimilikinya.



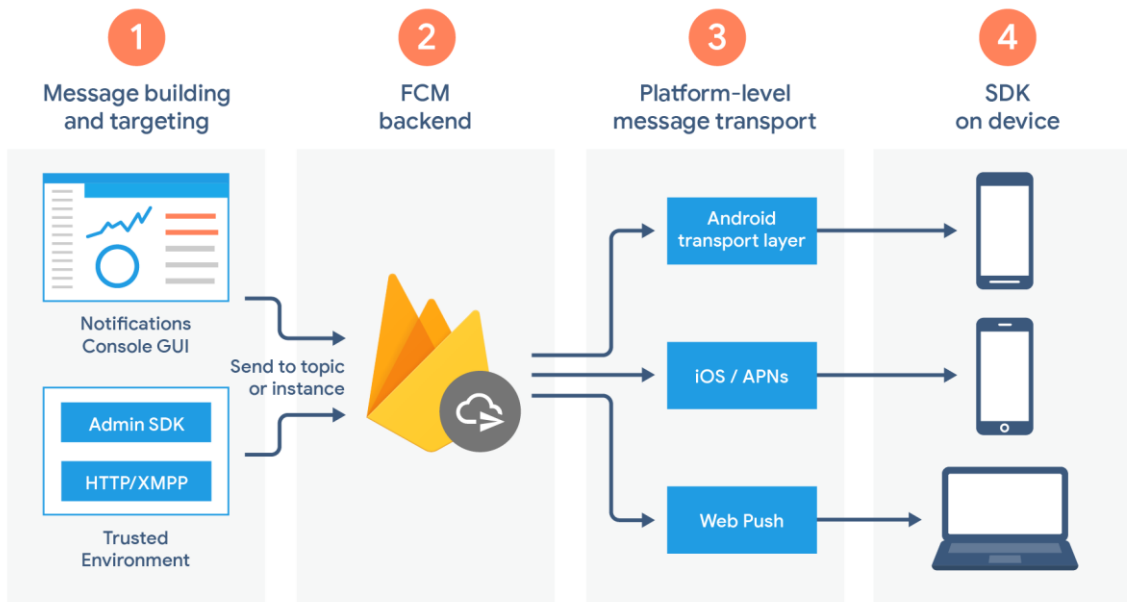
Proses interaksi sistem pada fitur *CRUD* ini akan ditunjukkan melalui diagram pada Gambar 3.8.



Gambar 3.8 Proses Sistem CRUD

### 3.3.3 Proses Interaksi Sistem pada Fitur *In-App Messaging*.

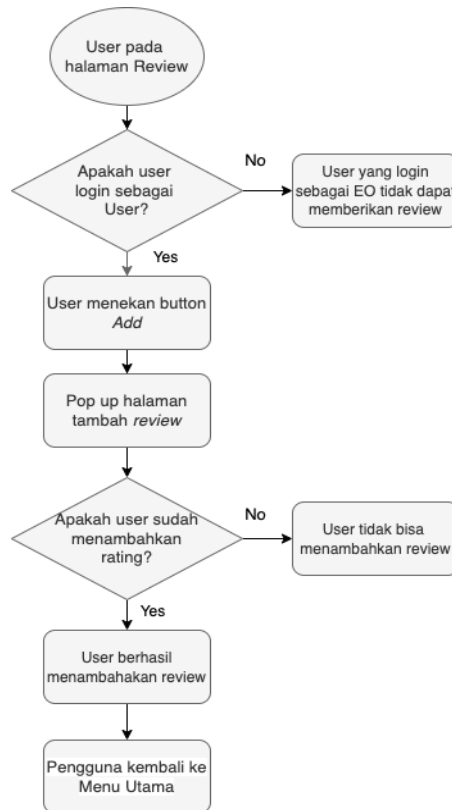
Pada aplikasi marketplace *Event Organizer* berbasis *mobile* ini menyediakan fitur *Messaging* yang bertujuan agar memudahkan user untuk berkomunikasi dengan *Event Organizer*. Fitur ini merupakan fitur yang cukup penting pada aplikasi ini untuk menjaga dari tindak penipuan yang sedang marak terjadi. Chat dapat direkam atau disimpan oleh sistem untuk dijadikan sebagai barang bukti apabila telah terjadi penipuan. Oleh karena itu, aplikasi ini merekomendasikan komunikasi *Event Organizer* dengan pengguna *Event Organizer* melalui fitur chat yang telah disediakan. Fitur ini dibangun menggunakan layanan firebase yaitu *Firestore* dan *Firebase Cloud Messaging (FCM)*. *Firestore* adalah solusi penyimpanan data terdistribusi yang dapat digunakan untuk menyimpan data tanpa biaya. *Firebase Cloud Messaging (FCM)* adalah solusi pertukaran pesan lintas platform yang dapat digunakan untuk mengirim pesan tanpa biaya. Proses interaksi sistem pada fitur *Messaging* ini akan ditunjukkan melalui diagram pada Gambar 3.9.



Gambar 3.9 Proses Sistem *Messaging*

### 3.3.4 Proses Interaksi Sistem pada Fitur *Add Review dan Rating*

Pada aplikasi marketplace *Event Organizer* berbasis *mobile* ini juga menyediakan fitur add review dimana user dapat memberikan *review* dan *rating* pada *Event Organizer*. Proses interaksi sistem pada fitur *Messaging* ini akan ditunjukkan melalui diagram pada Gambar 3.10.



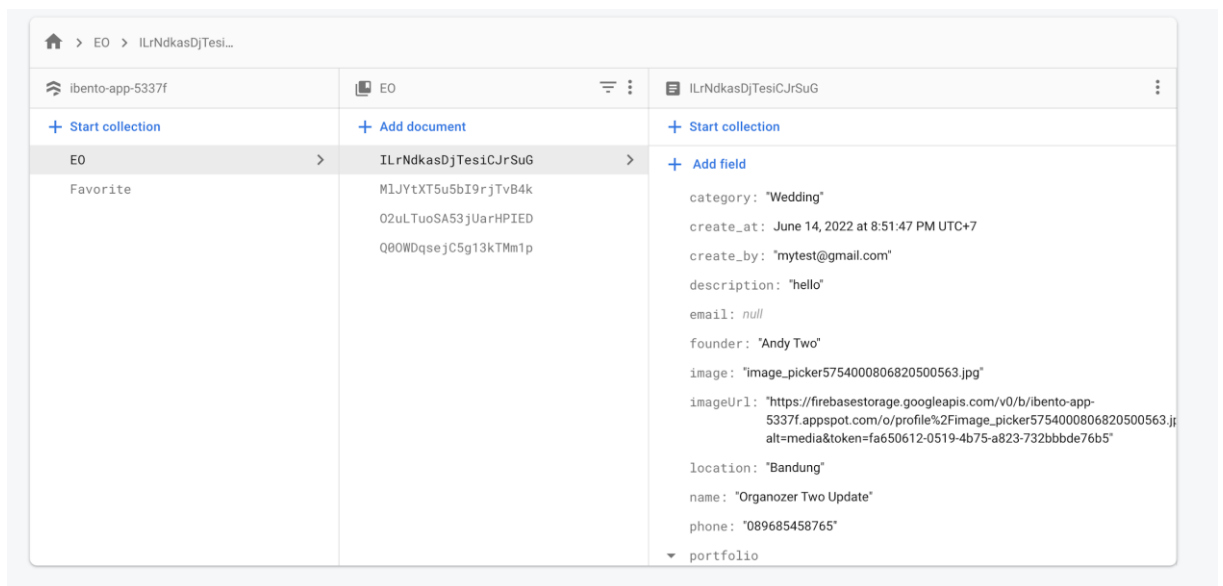
Gambar 3.10 Proses Sistem Add Review dan Rating

### 3.3.5 Perancangan dan Implementasi Database

Seperti yang telah dijelaskan sebelumnya, *database* yang digunakan dalam penelitian ini berbasis *cloud*. *Database* digunakan untuk menyimpan data dari *Event Organizer*. *Database* yang digunakan dalam penelitian ini ialah *Firebase Realtime Database* yang merupakan *database NoSQL* yang digunakan untuk menyimpan, membaca, menulis dan mengambil data secara *Realtime* yang disinkronkan kepada klien yang terhubung.

Data yang diambil dari aplikasi Marketplace *Event Organizer* yaitu nama *Event Organizer*, kategori *Event Organizer*, deksripsi, email, *founder*, gambar *Event Organizer*, lokasi *Event Organizer*, nomor handphone, dan juga portofolio.

*Database* akan menyimpan seluruh history dari data yang telah diterima seperti *create at*, *create by*, *url* dari gambar-gambar yang diterima, lalu jika pengguna meng-*update* data, maka *database* juga akan menyimpan data waktu *update* dan juga *user* yang meng-*update* data tersebut. Struktur *database* dapat dilihat pada Gambar 3.11.



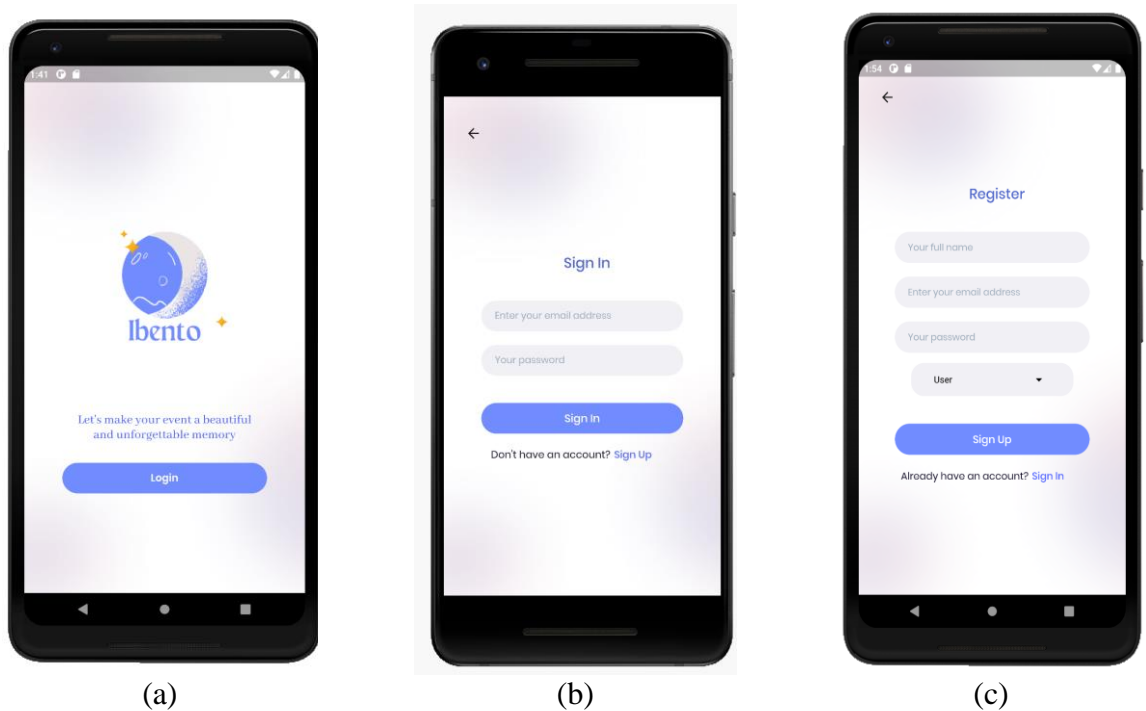
Gambar 3.11 Struktur Database pada Firebase



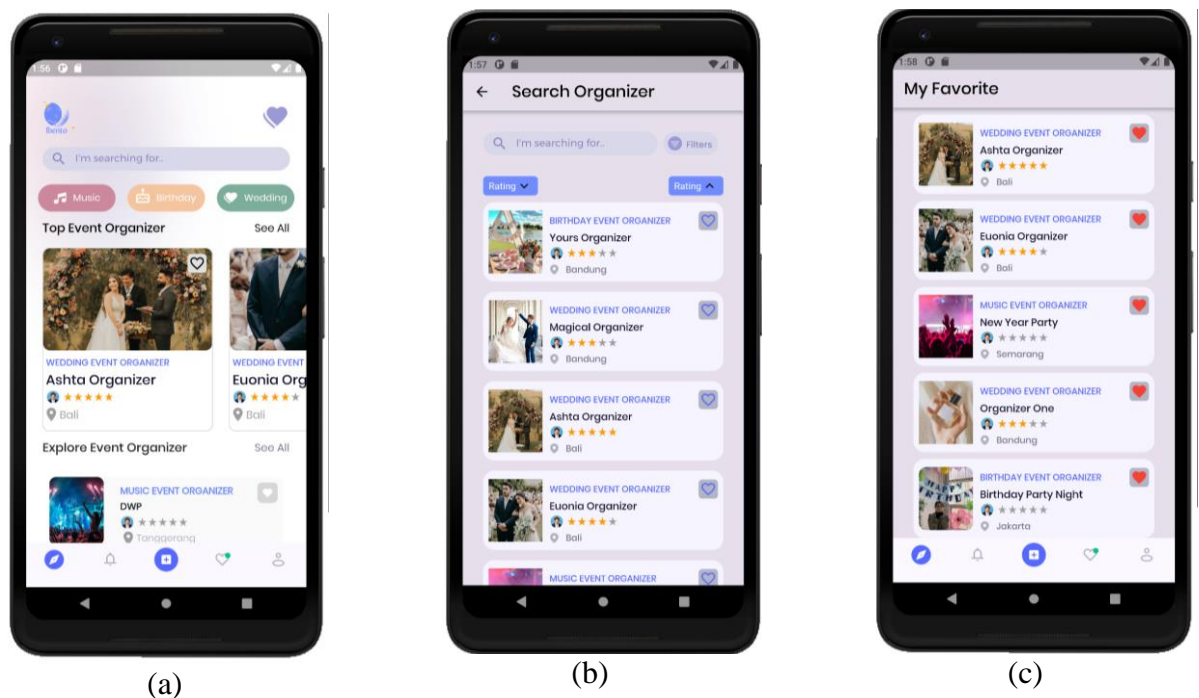


### 3.3.6 Perancangan dan Implementasi *User Interface* Aplikasi

Data dari *database* akan ditampilkan pada aplikasi. *Database* yang ada pada *cloud* akan menyimpan data yang di input oleh *Event Organizer* melalui aplikasi. Aplikasi yang dirancang menggunakan *framework Flutter*, yaitu sejenis *framework* yang digunakan untuk membantu *developer* membuat aplikasi *mobile multiplatform*.



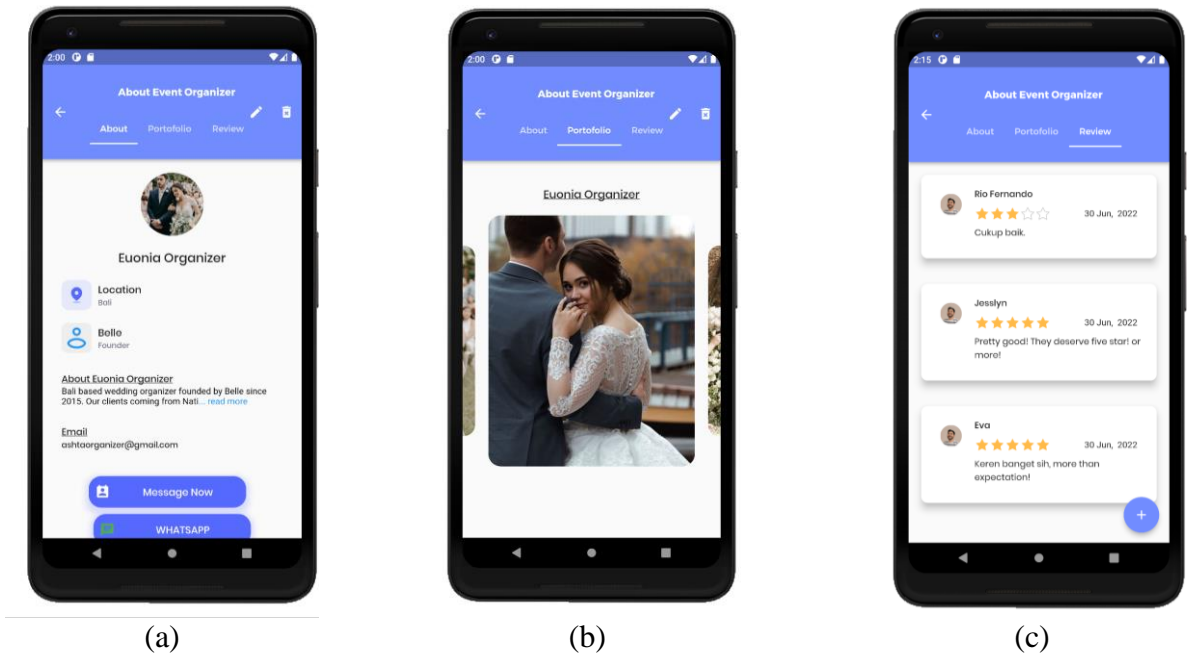
Gambar 3.12 User Interface Authentication



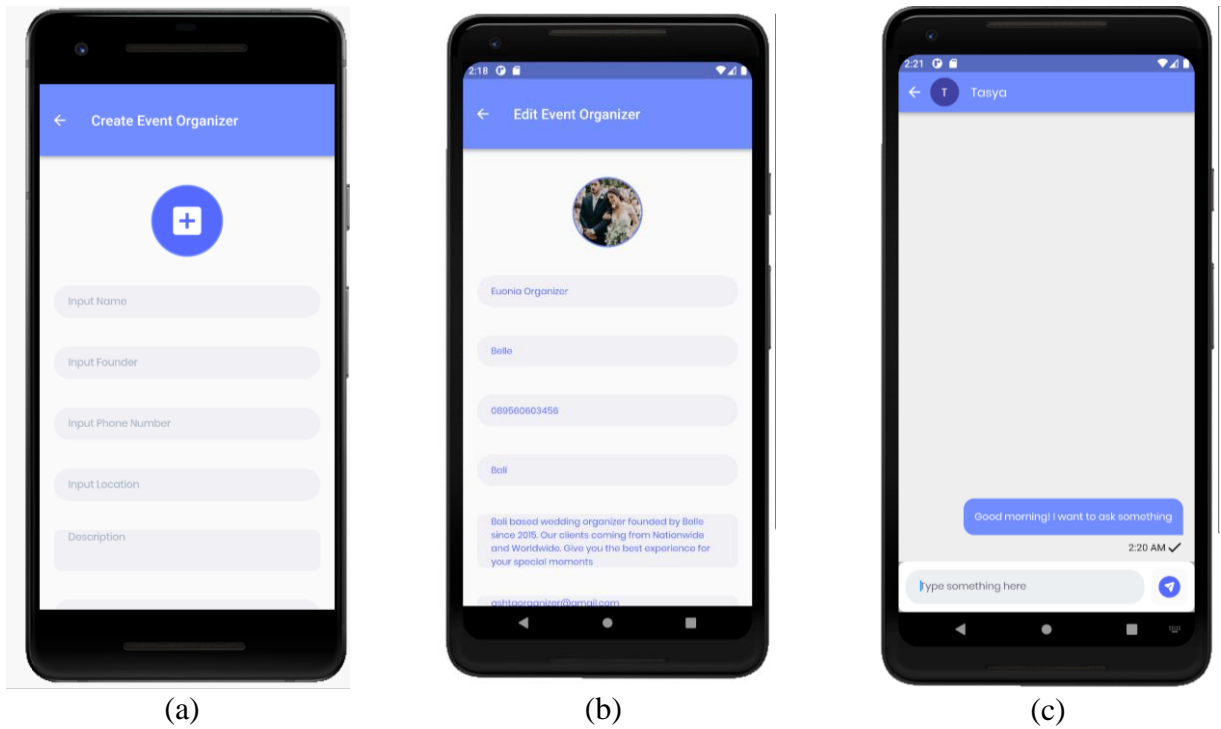
Gambar 3.13 User Interface Home, Search dan Favorite Page



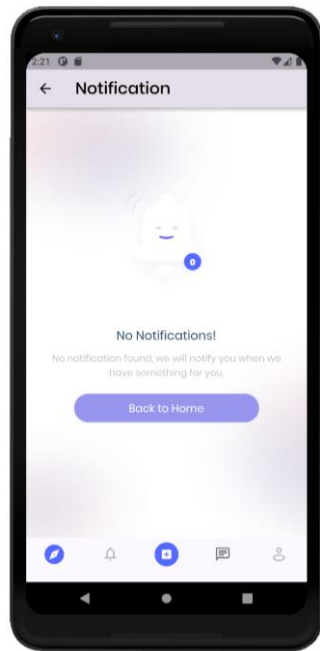
### BAB 3 PERANCANGAN DAN IMPLEMENTASI



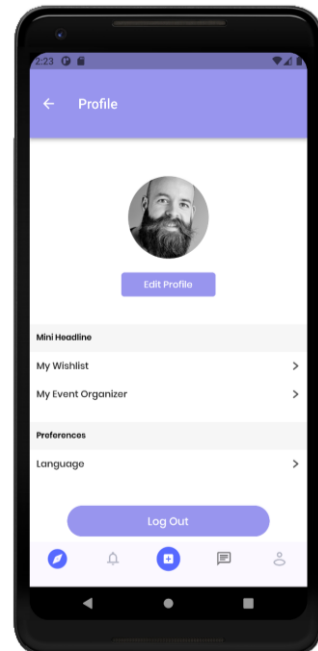
Gambar 3.14 User Interface Detail EO



Gambar 3.15 User Interface Detail Create, Edit dan Messaging



(a)



(b)

Gambar 3.16 User Interface Notification dan Profile Page