



## Implementasi Model *View Controller* pada Aplikasi Pemesanan Makanan Platform Sistem Operasi Android

Cynthia Hayat  
Universitas Kristen Krida  
Wacana  
[cynthia.hayat@ukrida.ac.id](mailto:cynthia.hayat@ukrida.ac.id)

Enggalwiguno Rahardja  
Universitas Kristen Krida  
Wacana  
[enggalwiguno.2014si020@civitas.ukrida.ac.id](mailto:enggalwiguno.2014si020@civitas.ukrida.ac.id)

Frans Pasamboan  
Universitas Kristen Krida  
Wacana  
[frans.2014si016@civitas.ukrida.ac.id](mailto:frans.2014si016@civitas.ukrida.ac.id)

### Abstrak

Sudah menjadi rahasia umum di kota-kota besar seperti Jakarta saat ini bahwa untuk memulai usaha kecil di bidang penjualan makanan dapat dilakukan dengan cukup mudah. Tetapi untuk dapat bertahan hidup, bahkan tumbuh di tengah persaingan pasar, eksekusi inovasi dan strategi superior perlu dilakukan. Makalah ini membahas pembuatan aplikasi pemesanan makanan menggunakan metode *Object-oriented programming (OOP)* berbasis mobile menggunakan platform android yang dirancang secara sistematis khusus untuk karakteristik objek penelitian. Persyaratan penemuan dilakukan dengan melakukan wawancara, membuat skenario, dan studi etnografi. Pemodelan Sistem diterapkan dengan metode UML melalui diagram aktivitas, diagram use-case, diagram kelas, dan diagram aliran data. Lebih jauh, menambahkan beberapa fitur pada kebutuhan dan tuntutan pemangku kepentingan dipenuhi dengan menggabungkan database MySQL dengan Firebase - platform berbasis Cloud dari Google inc. untuk mengintegrasikan akun pengguna seluler yang ada yang sudah dimiliki oleh pengguna pada perangkat, serta pemanfaatan fasilitas layanan lainnya yang memiliki fitur penting untuk pengembangan aplikasi di masa depan. Hasil akhir dari penelitian ini adalah 2 (dua) aplikasi yang terintegrasi dengan platform android, yang terdiri dari (1) aplikasi pemesanan makanan untuk klien (pelanggan) yang diharapkan dapat memudahkan pelanggan dalam mendapatkan informasi produk, yang selanjutnya memudahkan proses pembelian, dan (2) Manajemen

aplikasi dari proses penjualan makanan untuk admin (penjual / pemilik restoran) dan untuk melakukan kegiatan promosi tambahan pada kegiatan bisnisnya.

Kata kunci: OOP, MVC, Android, Pemesanan Makanan

### Abstract

It is common knowledge in big cities like Jakarta today that to start a small business in the field of food sales can be done quite easily. But to be able to survive, even growing in the midst of market competition, execution of innovations and superior strategies needs to be done. This paper discusses the making of a food ordering application using the mobile-based *Object-oriented programming (OOP)* method using an android platform that is systematically designed specifically to the characteristics of the object of study. Discovery requirements were carried out by conducting interviews, making scenarios, and ethnographic studies. System Modeling is applied with the UML method through activity diagrams, use-case diagrams, class diagrams and data flow diagrams. Furthermore, adding several features to the needs and demands of stakeholders was fulfilled by combining the MySQL database with Firebase - a Cloud-based platform from Google inc. to integrate the existing mobile user account that is already owned by the user on the device, as well as the utilization of other service facilities that have essential features for future application development. The final results of the study are 2 (two) applications integrated with the android platform, which consist of (1) Food ordering applications for

clients (customers) that are expected to facilitate customers in obtaining product information, which further facilitates the purchasing process, and (2) Application management of the food sales process for the admin (seller / owner of the restaurant) and for carrying out additional promotional activities on his business activities.

Keywords : OOP, MVC, Android, Food Ordering

## 1. Pendahuluan

Peluang bisnis makanan atau usaha kuliner di Indonesia dari tahun ke tahun semakin meningkat. Data dari Kementerian Perindustrian (KemPrin) pada tahun 2017, sektor makanan dan minuman tumbuh di kisaran 7.5%-7.8%. Sedangkan data dari Gabungan Pengusaha Makan dan Minuman Indonesia (Gapmmi) menunjukkan pertumbuhan bisnis usaha makanan minuman berada di kisaran 8.5%.

Meningkatnya usaha makanan dan minuman merupakan reaksi atas tingginya permintaan konsumen. Fenomena ini menunjukkan persaingan yang semakin ketat diantara usaha yang ada. Ditengah persaingan dengan puluhan usaha makanan lain yang ada di sekitarnya, mengharuskan pihak pemilik untuk terus mencari strategi-strategi khusus untuk pengembangan usahanya dengan melihat faktor-faktor eksternal maupun internal agar tetap bertahan bahkan untuk maju dan berkembang.

Opsi dalam melakukan pengembangan penjualan secara online saat ini secara dominan mengarah kepada aplikasi pada perangkat bergerak (mobile). Laju pertumbuhan ponsel pintar di seluruh dunia, menegaskan bahwa aplikasi mobile adalah instrumen vital untuk menjangkau dan merebut segmen pasar secara signifikan. Tingkat keterlibatan aplikasi mobile berkali lipat dibandingkan dengan website berbasis desktop.

Pemrograman berorientasi objek bukanlah merupakan hal yang baru, Dalam dua dekade terakhir hampir semua pengembang sudah dapat menerapkan konsep dari evolusi teknik pemrograman ini dengan baik dan telah menjadi paradigma pemrograman yang dominan digunakan oleh para pengembang (Onu, Nov-Dec 2015).

Analisis terhadap model bisnis yang ada, batasan masalah dengan merancang aplikasi pemesanan makanan yang disesuaikan dengan karakteristik usaha (*business customized mobile-application* (Ricky, 2014) (Fan, 2014)).

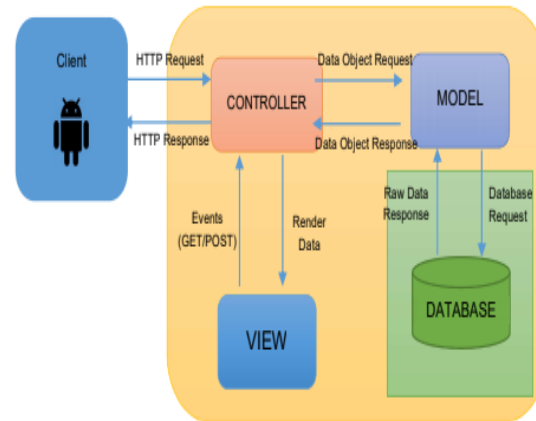
## 2. Metode Penelitian

### 2.1 Model View Controller (MVC)

MVC adalah pola perancangan perangkat lunak untuk pengembangan aplikasi web, yang merupakan satlah satu metode penerapan teknik OOP. MVC sangat populer karena mengisolasi logika aplikasi dari lapisan

antarmuka pengguna dan mendukung pemisahan masalah.

Pada MVC, *Controller* menerima semua permintaan untuk aplikasi dan kemudian bekerja dengan *Model* untuk mempersiapkan data yang dibutuhkan oleh *View*, kemudian *View* menggunakan data yang disiapkan oleh *Controller* untuk menghasilkan respon final yang layak. Abstraksi MVC yang akan dipergunakan pada aplikasi dapat digambarkan secara grafis pada gambar 1 di bawah ini: (Cahyawan, 2012) (Singh, Sharma, & Singh, 2016)



Gambar 1. Model View Controller

Pada penerapannya di setiap fungsi Aplikasi Pemesanan Makanan *Authentication*, untuk mengintegrasikan *user account* yang pada umumnya sudah dimiliki oleh pelanggan di perangkat selulernya, yang biasanya akun Google+ dan akun media sosial lainnya.

1. *Crash Reporting*, untuk mendapatkan feedback atas masalah-masalah (*error crash*) yang terjadi pada aplikasi pemesanan makanan di setiap perangkat seluler yang menggunakannya. Hal ini sangat berguna untuk meningkatkan kualitas aplikasi serta mengembangkannya dengan lebih baik lagi di masa mendatang.
2. *Cloud Storage*, untuk mengembangkan fitur perpesanan (obrolan / *chat*), lapisan arsitektur MVC akan diinteraksikan secara terpisah dengan database (MySQL) yang berfungsi sebagai model, yang pada akhirnya akan ditampilkan (view) pada *user interface platform* Android, dengan JSON *Web service* sebagai pengontrol (*controller*).
3. Untuk mendukung *controller* pada konsep MVC ini juga akan dikembangkan pemrograman *server side* untuk mengakses *database*, dalam hal ini memilih bahasa pemrograman PHP yang dapat mengeksekusi fungsi *encode* dan *decode* data dalam bentuk JSON, dan kemudian mentransmisikannya dari server ke client.

## 2.2 Android Support Library

Bagi pengembang Android, ketika hendak membuat suatu aplikasi pada versi terbaru OS Android, mereka juga menginginkan aplikasi tersebut dapat berjalan di perangkat yang menjalankan versi android lebih lawas. Maka dari itu pengembang harus menambahkan kompatibilitas melalui *Android Support Library* yang menyediakan berbagai kelas untuk membangun aplikasi, mulai dari komponen aplikasi dasar, widget untuk antarmuka pengguna, penanganan media, hingga komponen aplikasi TV. Dimana pada pengembangan Aplikasi Pemesanan Makanan ini mengutilisasikan komponen *Android Support Library*, diantaranya sebagai berikut (Singh, Sharma, & Singh, 2016):

1. App Components
2. User Interface
3. Material Design
4. Graphics
5. Utilities

## 2.3 Firebase

*Firebase* adalah sekumpulan layanan produk teknologi yang kini dikembangkan oleh Google inc. yang memungkinkan pengembang membuat aplikasi web tanpa harus melakukan pemrograman pada sisi server sehingga proses pengembangannya menjadi lebih cepat dan mudah.

Produk-produk *Firebase* merupakan platform bagi aplikasi mobile dan web, dengan *tools* dan infrastrukturnya di desain sedemikian rupa agar pengembang bisa membuat aplikasi yang berkualitas tinggi.

Implementasi *Firebase* pada Aplikasi Pemesanan Makanan ini dilakukan dengan mempertimbangkan permintaan dari pihak pemilik dalam menggunakan beberapa fitur yang akan lebih mudah diaplikasikan dengan menggunakan layanan *Firebase*, diantaranya: (Singh, Sharma, & Singh, 2016)

## 2.4 Tinjauan Teknologi

Lingkup perancangan yang ditemukan adalah sebagai berikut:

1. Pembuatan aplikasi pemesanan makanan yang disesuaikan dengan karakteristik usaha (*business customized mobile-application*).
2. Aktifitas terhadap transaksi menu yang fleksibel dan interaktif, yang melingkupi produk:
  - a. Makanan
  - b. Minuman
  - c. Menu special atau menu masakan khusus lain yang diproduksi berbeda setiap harinya dan secara tidak teratur (*random*).
3. Penerapan fungsi-fungsi, sebagai berikut:
  - a. Pembagian, pemilihan dan pencarian produk

- b. Navigasi yang disusun sederhana sehingga tidak membingungkan.
- c. Promosi menu spesifik
- d. Notifikasi bagi penjual, sehingga pesanan baru tidak terlewatkan
- e. Interaksi / komunikasi antara customer dan penjual, serta antar customer.
- f. Metode Pembayaran
- g. Riwayat Penjualan (admin / penjual)
- h. Riwayat Pembelian (client / pembeli)
- i. Rekaman seluruh kegiatan / proses pada aplikasi
- j. Analisa penjualan sederhana
- k. Informasi lokasi usaha
- l. Berbagi aplikasi (*share app*)
- m. Menu favorit dan histori untuk memudahkan user untuk melakukan order yang telah dilakukan sebelumnya
- n. Login dan registrasi user yang mudah dan flexible, diharapkan dapat menggunakan alamat email atau akun sosial media yang sudah dimiliki sebelumnya oleh pengguna.

Pada pengembangan tahap awal dari Aplikasi ini, penulis memutuskan untuk mencoba memenuhi permintaan dari pemilik / penjual terhadap aplikasi yang hendak dikembangkan, dengan mengambil pokok penting pengembangan, mengacu kepada poin berikut:

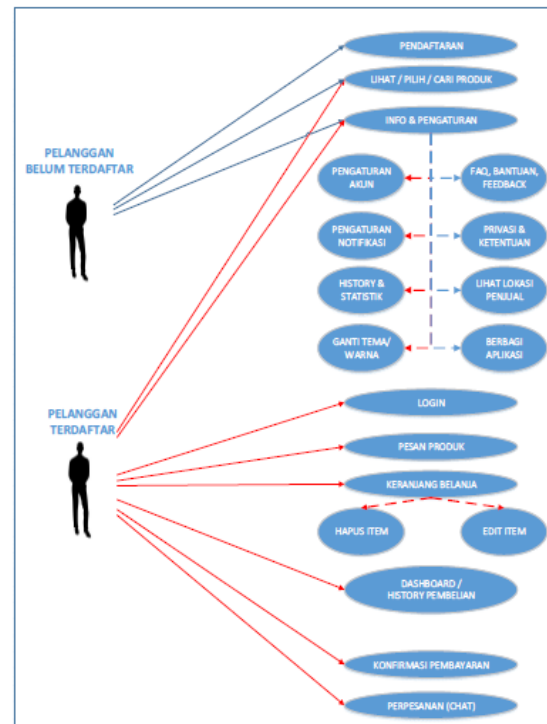
1. Fungsi Umum:
  - a. Berbasis Sistem Operasi Android dengan mengikuti standarisasi Google *design support libraries* (Soomerville, 2011) dan *repositories* terkini.
  - b. Membuat dua buah aplikasi terpisah namun terintegrasi, dengan tujuan mengurangi kompleksitas baik penjual (admin) maupun pelanggan (client) dalam menggunakan aplikasi ini, juga dimaksudkan agar pihak admin dapat melakukan pengelolaan tanpa harus mengandalkan perangkat komputer lainnya.
  - c. Navigasi dibuat sederhana sehingga penjual (admin) maupun pelanggan (client) mudah melakukan pencarian produk ataupun fitur lain.
  - d. Membuat *user interface* yang informatif, atraktif & menarik sebagai nilai tambah dari aplikasi yang dapat mempengaruhi minat pelanggan dalam menggunakan aplikasinya.
  - e. Membuat fitur obrolan yang terintegrasi didalam aplikasi sehingga mempermudah penjual memberikan tanggapan terhadap keinginan pelanggannya, juga bisa sebagai media sosial antar pelanggan.
  - f. Terdapat fitur notifikasi bagi penjual maupun pembeli.
2. Fungsi Admin:

- a. Membuat sistem informasi yang memudahkan penjual untuk mengelola proses penjualan yang terintegrasi dengan system informasi kepada pelanggan.
  - b. Membuat fitur laporan dan/atau analisa transaksi sederhana, dengan periode waktu yang dapat diatur / dipilih sesuai dengan kebutuhan penjual.
  - c. Membuat fitur otorisasi status kegiatan secara terstruktur terhadap langkah-langkah mulai dari proses penerimaan pesanan hingga selesainya transaksi penjualan.
  - d. Membuat fitur untuk melakukan pengurutan (*sorting*), pemilihan (*selection*) terhadap transaksi serta status dari kegiatan-kegiatan yang terjadi.
  - e. Membuat fitur pengaturan terhadap menu produk, seperti menambah (*add*), mengubah, (*edit*) dan menghapus (*delete*).
  - f. Membuat fitur untuk mengaktifkan / mendeaktivasi produk
  - g. Membuat fitur untuk promosi menu makanan dan minuman (*product promotion*).
  - h. Membuat fitur obrolan (*chat / messaging*) yang terintegrasi didalam aplikasi.
3. Fungsi Client:
- a. Proses transaksi pemesanan makanan yang efektif (proses pemesanan bisa dilakukan dengan cepat oleh pelanggan, tidak berbelit dan tidak membingungkan).
  - b. Membuat fitur bantuan atau fitur yang dapat memandu pelanggan yang baru atau belum berpengalaman dalam menggunakan aplikasi.
  - c. Menggunakan metode keranjang belanja (*shopping cart*) pada kegiatan transaksi untuk mengantisipasi aktifitas dan masalah yang umum terjadi dalam proses pemesanan seperti: dapat mengubah jumlah pesanan (menambah mengurangi, atau menghapus pesanan).
  - d. Metode registrasi user yang mudah dan flexibel: dapat menggunakan alamat email user, atau akun media sosial yang sudah dimiliki oleh user.
  - e. Membuat fitur yang memudahkan user untuk melakukan order yang telah dilakukan sebelumnya (menu favorit & histori pembelian).
  - f. Membuat fitur konfirmasi pesanan dengan mencantumkan metode pembayaran pada tahap akhir pemesanan.
  - g. Membuat fitur obrolan (*chat / messaging*) yang terintegrasi didalam aplikasi, sehingga client dapat berkomunikasi dengan admin, serta memungkinkan berkomunikasi dengan client lainnya.

### 3. Hasil dan Pembahasan

#### 3.1 Analisis Proses Bisnis

Alur proses bisnis dari sisi *client app* bisa dilihat pada gambar 2 di bawah ini.

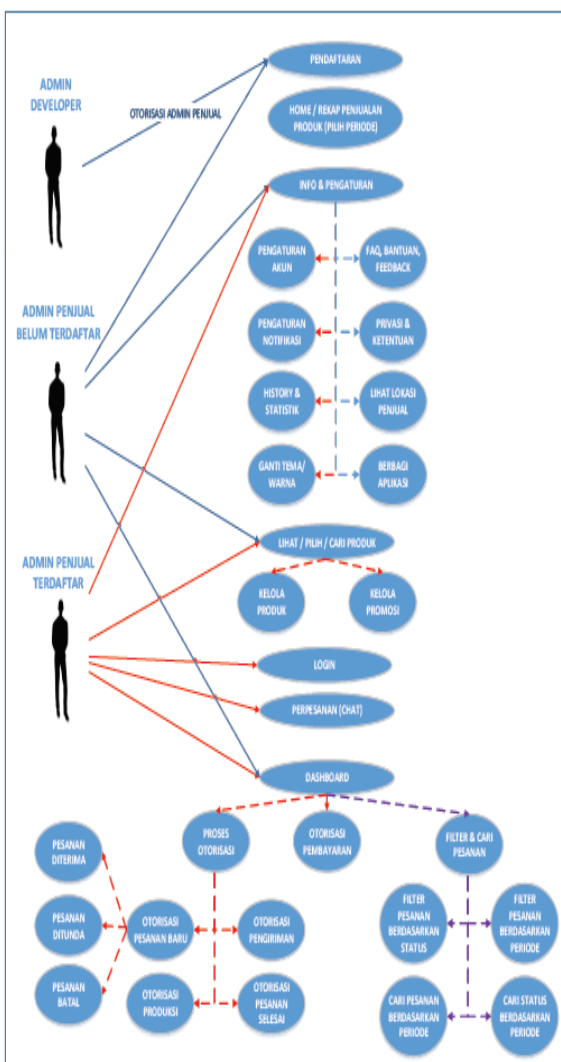


Gambar 2. Use Case Diagram – Client App (Aplikasi Pelanggan)

Pelanggan yang menggunakan aplikasi terdiri dari pelanggan yang belum terdaftar dan pelanggan yang sudah terdaftar. Jika pelanggan belum terdaftar, diharuskan melakukan pendaftaran terlebih dahulu melalui dua cara yaitu melalui akun Gmail maupun lewat akun Facebook yang mungkin sudah dimiliki oleh pelanggan. Setelah itu, pelanggan dapat melakukan pengaturan akun, pengaturan notifikasi, melihat lokasi penjual, maupun melihat history dan statistik pembelian yang telah dilakukan.

Jika pelanggan sudah terdaftar dan memiliki akun pada aplikasi, pelanggan bisa langsung melakukan login dan kemudian melakukan pemesanan produk. Produk yang telah dipesan kemudian akan masuk ke dalam keranjang belanja. Apabila pelanggan ragu-ragu dalam melakukan pemesanan, masih bisa melakukan proses hapus item dan edit item pada keranjang belanja tersebut. Pelanggan juga bisa melakukan chat apabila ingin melakukan pesanan secara manual. Jika pemesanan sudah tidak ada perubahan, akan muncul history pembelian dan kemudian menu untuk melakukan konfirmasi pemesanan.

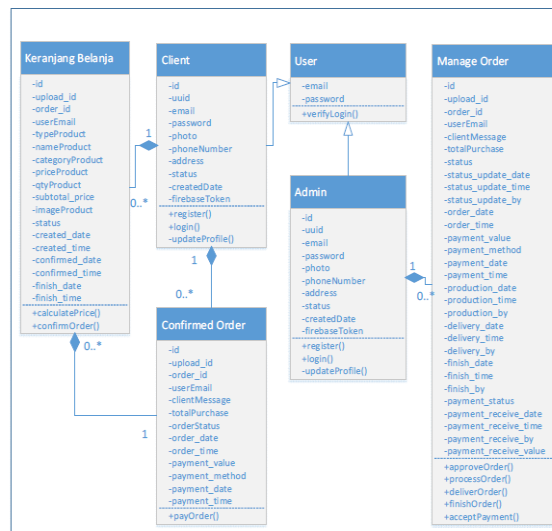
Sedangkan alur proses bisnis dari sisi *admin app* bisa terlihat pada gambar 3 di bawah ini.



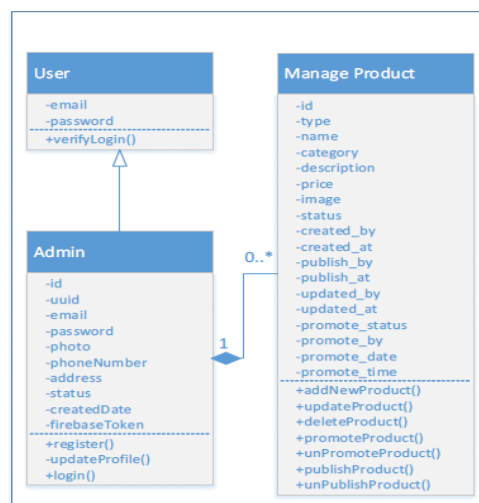
Gambar 3. Use Case Diagram – Admin App (aplikasi Penjual)

Admin terdiri atas tiga bagian yaitu admin developer, admin penjual yang belum terdaftar, dan admin penjual yang sudah terdaftar. Admin developer memiliki otoritas untuk melakukan penambahan admin penjual. Pada admin penjual belum terdaftar terlebih dahulu melakukan permintaan pendaftaran kepada admin developer. Jika sudah terdaftar, admin penjual bisa melakukan pengaturan akun, pengaturan notifikasi, maupun melihat history dan statistik pembelian yang telah dilakukan. Admin penjual yang sudah terdaftar selain bisa mengakses fitur di atas, juga dapat melakukan pengelolaan produk maupun pengelolaan promosi. Fitur-fitur yang bisa dilakukan oleh admin penjual yang sudah terdaftar berhubungan dengan interaksi dengan pembeli diantaranya melakukan chat, mengelola pesanan (menerima, menunda, dan menolak pesanan), melakukan otorisasi (otorisasi pesanan baru,

otorisasi pengiriman, otorisasi produksi, otorisasi pesanan selesai, dan otorisasi pembayaran), serta mengatur fitur untuk melakukan filter dan cari pesanan (berdasarkan status dan berdasarkan periode). Selain itu, alur proses bisnis dengan menggunakan class diagram terlihat pada gambar 4 dan gambar 5 di bawah ini.



Gambar 4. Class Diagram Transaksi Penjualan Pembelian



Gambar 5. Class Diagram Pengelolaan Produk – Admin App (Aplikasi Penjual)

Pada transaksi penjualan dan pembelian terdiri atas 6 class tabel yaitu keranjang belanja, client, user, manae order, admin dan confirmed order. Sedangkan pada class diagram, pengelolaan produk terdiri atas 3 class tabel yaitu user, admin, dan manage product.

### 3.2 Perancangan Database

Pada penerapan pengembangan aplikasi Pemesanan Makanan dilakukan kombinasi penyimpanan data pada basis data MySQL dan Firebase Realtime Database, dimana penggunaanya dibagi atas:

1. MySQL database; data produk dan data transaksi
2. Firebase Realtime Database; data akun (client & admin) dan data pemesanan / messaging (chat)

Yang secara grafis dapat digambarkan seperti gambar 6 berikut ini:



**Gambar 6. Arsitektur Database untuk Otentikasi dan Transaksi Pengguna**

Keterangan:

- a. Pada setiap kegiatan didalam Aplikasi Pemesanan Makanan ini, pengguna akan selalu terhubung dengan SDK pada Firebase yang telah dikonfigurasi untuk mengotentikasikan dengan akun yang sudah dimiliki pengguna sebelumnya. Terdapat tiga jenis provider untuk metode autentikasi, yaitu: Google, Facebook dan alamat email resmi yang dimiliki oleh user lainnya.
- b. Apabila proses otentikasi sukses, Firebase akan menginformasikan kepada perangkat mengenai detail yang diperlukan untuk dapat melanjutkan kegiatan. Apabila tidak sukses maka sistem tidak memperkenankan untuk melanjutkan kegiatan yang memiliki akses terbatas.
- c. Apabila otorisasi sukses dilakukan, maka pengguna dapat berinteraksi dengan aplikasi yang diatur dengan akses terbatas, misalnya untuk client / pelanggan dalam melakukan transaksi pembelian, atau untuk admin / penjual dalam mengelola proses penjualan, data dikirim dalam format JSON.
- d. PHP pada web server selaku controller akan mengeksekusi query data yang diminta oleh

pengguna, dengan menterjemahkan format JSON menjadi query /bahasa SQL.

- e. Database MySQL akan memberikan respon data yang diminta untuk selanjutnya controller merubah data yang didapat menjadi Format JSON kembali.
- f. Web server akan meneruskan data dalam format JSON tersebut kepada perangkat pengguna, yang kemudian akan dirubah menjadi tampilan pada antarmuka sistem operasi Android.

Pada desain database MySQL, didesain struktur data untuk menghindari kompleksitas dalam menulis program, dengan cara menyusun tabel tidak banyak berbeda dengan class diagram yang digambarkan sebelumnya pada paper ini, yaitu tanpa memisahkan data setiap entity yang bisa membentuk banyak relasi. Konsekuensi dari penggabungan semua entity ini adalah data yang direkam sudah berbentuk normalisasi. Selanjutnya pada database MySQL, penulis menambahkan satu buah fungsi Trigger pada tabel\_confirm untuk fungsi penomoran pesanan otomatis (sales order numbering). Entity Relationship Diagram bisa terlihat pada gambar 7 di bawah ini.

| tblfamily.tabel_product    | tblfamily.tabel_confirm        | tblfamily.tabel_order            |
|----------------------------|--------------------------------|----------------------------------|
| id : int(10)               | id : int(10)                   | id : int(10)                     |
| type : varchar(20)         | order_id : varchar(25)         | upload_id : varchar(50)          |
| name : varchar(50)         | upload_id : varchar(50)        | order_id : varchar(15)           |
| category : varchar(50)     | email : varchar(50)            | email : varchar(50)              |
| description : varchar(255) | message : varchar(200)         | type : varchar(25)               |
| price : int(15)            | purchase : int(15)             | name : varchar(50)               |
| status : varchar(20)       | status : varchar(25)           | category : varchar(50)           |
| created_by : varchar(100)  | status_update_date : date      | price : int(15)                  |
| created_at : datetime      | status_update_time : time      | newQty : int(15)                 |
| publish_at : datetime      | status_update_by : varchar(50) | subtotal : int(15)               |
| publish_by : varchar(50)   | order_date : date              | imageurl : text                  |
| updated_by : varchar(100)  | order_time : time              | status : varchar(15)             |
| updated_at : datetime      | payment_value : int(11)        | created_date : date              |
| promote : varchar(10)      | payment_method : varchar(20)   | confirmed_time : time            |
| promote_by : varchar(50)   | payment_date : date            | confirmed_date : date            |
| promote_date : date        | payment_time : time            | confirmed_time : time            |
| promote_time : time        | production_date : date         | finish_date : date               |
| qty_forecast : int(10)     | production_time : time         | finish_time : time               |
| qty_inhand : int(10)       | production_by : varchar(50)    | finish_by : varchar(50)          |
|                            | delivery_date : date           | payment_status : varchar(50)     |
|                            | delivery_time : time           | payment_receive_date : date      |
|                            | delivery_by : varchar(50)      | payment_receive_time : time      |
|                            | finish_date : date             | payment_receive_by : varchar(50) |
|                            | finish_time : time             | payment_receive_value : int(20)  |
|                            | tblfamily.confirm_seq          |                                  |
|                            | id : int(10)                   |                                  |

Gambar 7. Entity Relationship Diagram MySQL

### 3.3 Implementasi

Proses dalam memulai aplikasi mengharuskan pengguna terkoneksi dengan jaringan internet dan web server, jika persyaratan utama ini tidak terpenuhi, maka aplikasi tidak bisa dipergunakan, karena semua fungsi dan fitur yang terdapat pada aplikasi mengandalkan layanan ini. Maka dari itu dibuatkan fungsi berikut:

- Fungsi memeriksa koneksi internet perangkat
- Fungsi memeriksa koneksi ke database MySQL pada Web Server

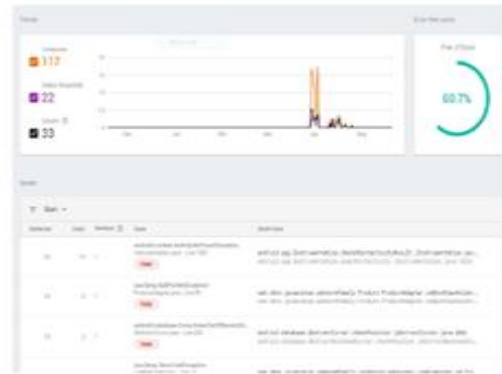
Jika salah satu persyaratan jaringan tidak tersedia, maka aplikasi terhenti dan mengarahkan (redirect) ke halaman informasi. Tampilan halaman awal (home) dengan mengimplementasikan 3<sup>rd</sup> party library amlucurran/ShowcaseView untuk memandu pengguna serta memberikan informasi lainnya. Bantuan ini hanya ditampilkan satu kali saja, yaitu pertama kali user menggunakan aplikasi. Tampilan halaman awal aplikasi bisa dilihat pada gambar 8 di bawah ini.



**Gambar 8. Halaman Awal Aplikasi dengan Promo Banner Bergerak Mengimplementasikan 3<sup>rd</sup> Party Library**

Pada setiap pengembangan perangkat lunak sebelum mencapai kestabilan serta berhentinya aplikasi secara tak terduga (system crash), maka dari itu diterapkan penanganan terhadap kesalahan (error handling) yang dapat menginformasikan secara mendetail mengenai kesalahan yang terjadi. Dengan menerapkan Firebase Crash Reporting, sumber penyebab kesalahan/ kegagalan pada aplikasi dapat diketahui untuk selanjutnya diperbaiki dan disempurnakan. Pengelolaan data pada Firebase realtime dapat dilihat pada gambar 9 di bawah ini.

Gambar 9. Pengelolaan Data pada Firebase



### Realtime Database

## 4. Simpulan

Konsep Model *View Controller* yang umum dipergunakan untuk aplikasi Website, dapat juga diaplikasikan dalam pemrograman berbasis Android yang menerapkan penggunaan database pada web server. Untuk penggunaan bahasa pemrograman PHP dan basis data MySQL dipergunakan sebagai Controller dan Model bagi aplikasi, untuk dapat melayani organisasi data yang secara dinamis dan terpusat. Sedangkan penerapan Android 3<sup>rd</sup> party library dapat meningkatkan kualitas dari aplikasi, juga mempermudah serta mempercepat proses pengembangannya. Penggunaan Firebase SDK dan Google API sangat menguntungkan baik untuk pengembang maupun pengguna aplikasi.

## 5. Referensi

- Cahyawan, N. L. (2012). Implementation of MVC (Model-View-Controller) Architectural to Academic Management Information System with Android Platform Base. *International Journal of Computer Applications*, 57(8), 1-6.
- Fan, Y. (2014). *Mobile Food Ordering Application*. Finlandia: VAASAN AMMATTIKORKEAKOULU University of Applied Sciences.
- Onu, F. I. (Nov-Dec 2015). The Importance of Object-Oriented Programming in This Era of. *IOSR Journal of Computer Engineering (IOSR-JCE)*, Volume 17( Issue 6, Ver. 1 ), 30-40.
- Pressman, R. S. (2010). *Software Engineering a Practitioner's Approach, Seventh Edition*. United State of America: McGraw-Hill.
- Ricky, M. Y. (2014). Mobile Food Ordering Application using Android OS Platform . *The European Physical Journal Conferences*.

Singh, A., Sharma, S., & Singh, S. (2016). Android Application Development using Android Studio and PHP Framework. *IJCA Proceedings on Recent Trends in Future Prospective in Engineering and Management Technology, RTFEM 2016* (1), 5-8.

Soomerville, I. (2011). *Software Engineering Ninth Edition* . Pearson Education Inc.