

# MoBI, Sistem Informasi berbasis Android

Baktiar Krisna Mukti Nugroho<sup>1</sup>, Handoko<sup>2</sup>, Hartanto Kusuma Wardana<sup>3</sup>

Fakultas Teknik Elektronika dan Komputer,  
Universitas Kristen Satya Wacana, Salatiga

<sup>1</sup>bakrimuno@gmail.com, <sup>2</sup>handoko@uksw.edu, <sup>3</sup>hartanto.kusuma@staff.uksw.edu

## Ringkasan

PT Tripilar Betonmas merupakan perusahaan yang bergerak di bidang industri manufaktur atap gelombang asbes dan *silica board*. Pelaporan data yang kurang fleksibel menjadi masalah tersendiri pada perusahaan ini, di antaranya untuk melihat laporan masih menggunakan aplikasi *desktop*. Hal ini mengurangi fleksibilitas dan mobilitas untuk melihat laporan jika berada di luar lingkungan perusahaan. Di sisi lain, Android adalah sistem operasi yang paling banyak digunakan oleh pengguna *smartphone* seluruh dunia karena memiliki beberapa kelebihan di antaranya, memiliki banyak aplikasi yang dapat digunakan secara gratis, banyak *library* yang disediakan dan bersifat *open source* sehingga mudah untuk dikembangkan. Aplikasi Android dirancang dengan tujuan untuk mengatasi kekurangan aplikasi perusahaan dalam hal fleksibilitas dan mobilitas untuk melihat laporan. Dengan aplikasi Android ini, *user* dapat melihat informasi perusahaan walaupun berada di luar lingkungan perusahaan. Aplikasi Android ini dilengkapi dengan autentikasi *user* dengan Firebase, dapat menampilkan informasi penjualan, produksi dan barang rusak berupa grafik. Pada hari kerja pengguna akan menerima notifikasi penjualan per hari yang sudah dicapai setiap jam. Aplikasi ini dapat digunakan pada sistem operasi Jelly Bean hingga Oreo. Selain itu aplikasi ini dilengkapi dengan 3 autentikasi untuk menjaga keamanan data perusahaan yang terdiri dari autentikasi aplikasi *mobile*, *webservice* dan *database*.

**Kata kunci:** Android, *Webservice*, *Database* dan Aplikasi *Mobile*

## 1. Pendahuluan

PT Tripilar Betonmas merupakan perusahaan yang bergerak di bidang industri manufaktur atap gelombang Asbes dan *Silica Board*. Pelaporan data yang kurang fleksibel menjadi masalah tersendiri dalam perusahaan tersebut karena menggunakan aplikasi *desktop*. Aplikasi *desktop* memiliki keterbatasan yaitu mobilitas untuk melihat laporan jika berada di luar lingkungan kantor.

Berdasarkan permasalahan di atas, dirancang sebuah sistem untuk menampilkan laporan omzet, penjualan dan produksi. Aplikasi *mobile* berbasis Android dirancang untuk dapat berkomunikasi dengan *database* melalui *webservice*. Aplikasi *mobile* ini sebagai pelengkap aplikasi *desktop* dan pengganti aplikasi *web* yang sudah ada serta menjadi solusi dari masalah keterbatasan akses yang banyak dipengaruhi oleh faktor jarak maupun kemudahan akses *database*, sehingga dapat meningkatkan produktivitas perusahaan.

*Business Intelligence* (BI) adalah sekumpulan teknik dan alat untuk mentransformasi data mentah menjadi informasi yang berguna dan bermakna untuk tujuan analisis bisnis. Teknologi BI dapat menangani data yang tak terstruktur dalam jumlah yang sangat besar untuk membantu mengidentifikasi, mengembangkan, dan selain itu membuat kesempatan

strategi bisnis yang baru. Tujuan dari BI yaitu untuk memudahkan interpretasi dari jumlah data yang besar tersebut.

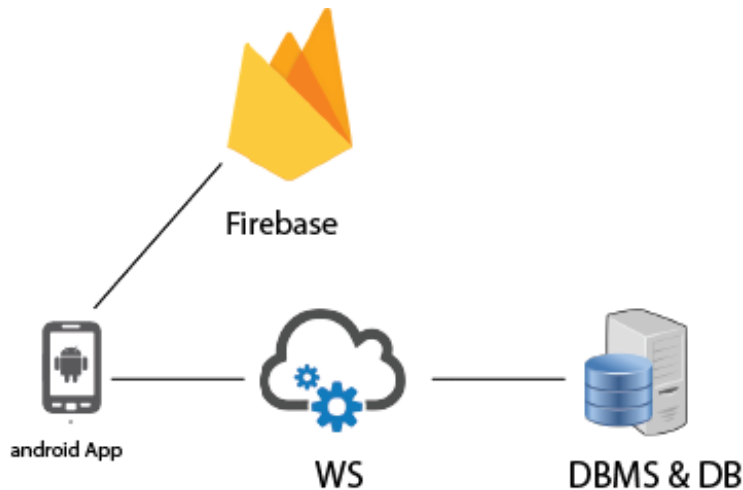
Aplikasi *mobile* berbasis Android ini memiliki keunggulan dibandingkan aplikasi sejenis :

Tabel 1. Perbandingan dengan aplikasi sejenis

Keterangan	Aplikasi yang dirancang	Aplikasi sejenis ( <i>FastBudget</i> )
Data	Data rill berdasarkan database yang ada di perusahaan	Data masih bersifat lokal pada aplikasi
Keamanan	Memerlukan 3 autentikasi untuk mengakses data yang tersimpan (aplikasi, webservice dan database)	Tidak ada autentikasi untuk mengakses data yang tersimpan

## 2. Perancangan Sistem

Pada bagian ini dijelaskan mengenai perancangan sistem yang secara umum dapat dilihat pada Gambar 1.

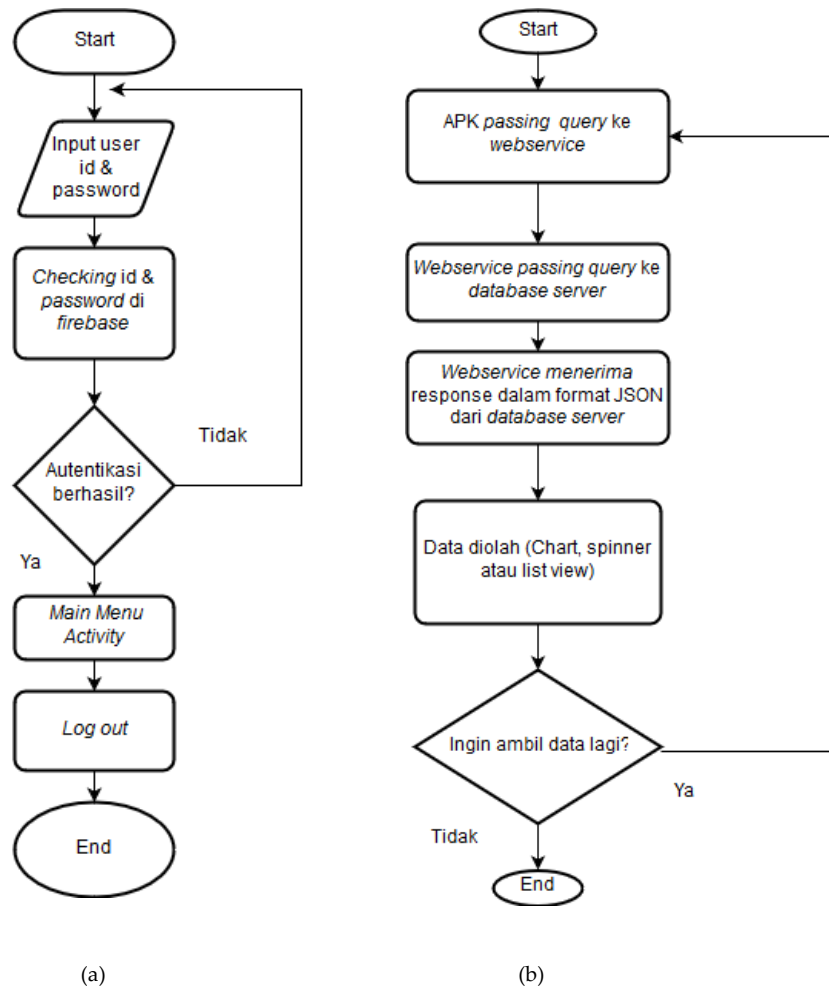


Gambar 1. Diagram Sistem

Aplikasi *mobile* dirancang menggunakan *framework* Android Studio. Aplikasi *mobile* berbasis Android bekerja melalui Internet untuk mengakses data pada Firebase untuk proses login. Setelah berhasil login maka aplikasi akan mengakses *web service* ketika menampilkan *report*. Selanjutnya *web service* akan mengakses database dengan mengirimkan *query*. Database Management System (DBMS) akan mengirim *dataset*, hasilnya akan diubah dalam format JSON pada *webservice* dan dikirimkan ke aplikasi.

Aplikasi Android ini memiliki *user interface* yang terdiri dari beberapa bagian utama di antaranya, *login activity* dan *function menu activity*

1. *Login Activity* berfungsi untuk keamanan aplikasi supaya tidak semua *user* dapat mengakses aplikasi tersebut, hanya *user* yang memiliki hak akses yang dapat mengakses aplikasi tersebut
2. *Function Menu Activity* berfungsi untuk pengambilan data ke *database*.

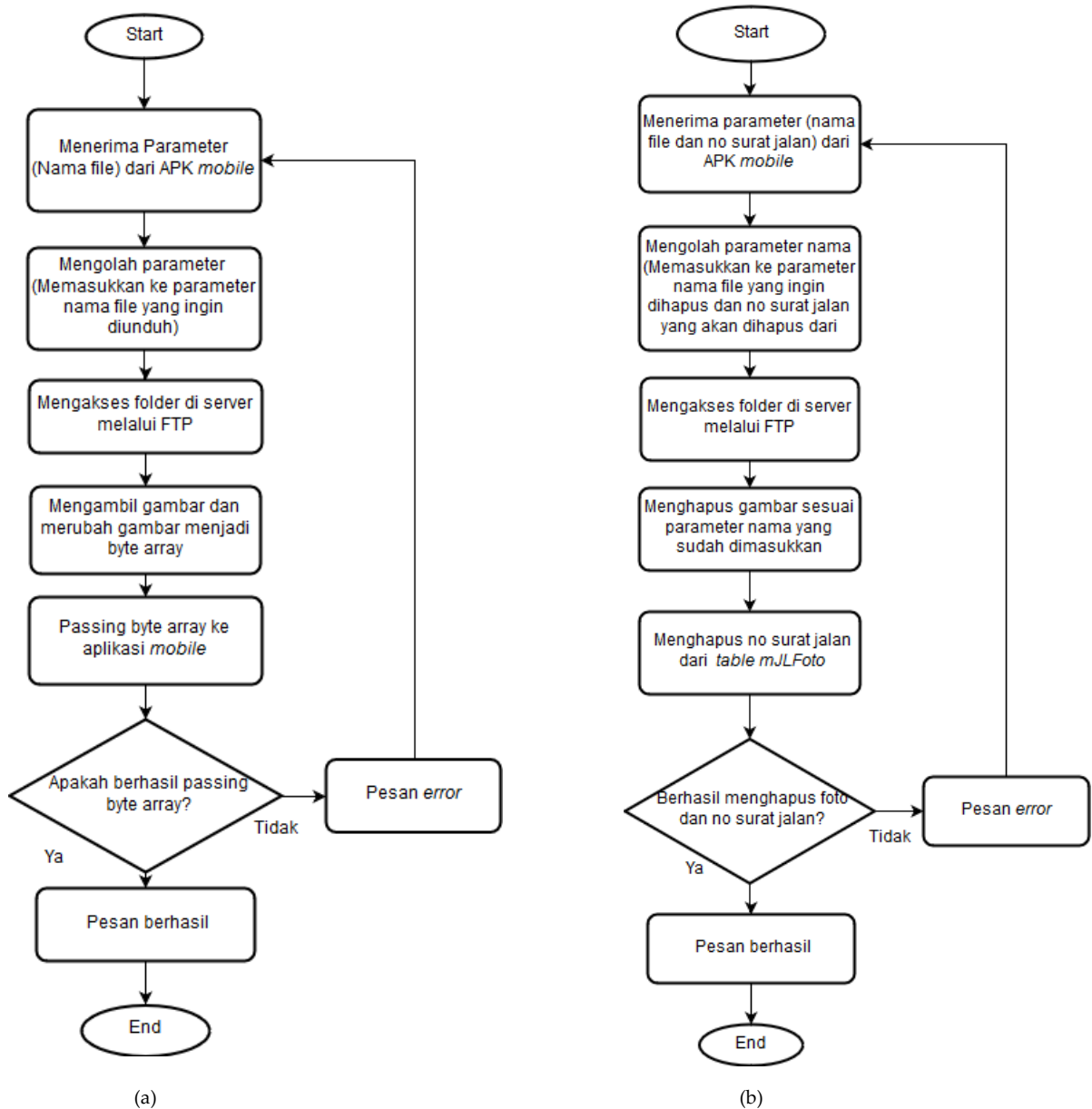


Gambar 2. a. Diagram Alir Login Activity b. Diagram Alir Pengambilan Data

Login activity akan muncul ketika pertama kali aplikasi mobile dijalankan. User memasukkan autentikasi id dan password pada activity ini. Selanjutnya id dan password yang sudah dimasukkan akan diperiksa di Firebase. Setelah berhasil diperiksa aplikasi mobile akan berpindah ke halaman menu activity (Gambar 2.a).

Sedangkan pada proses pengambilan data, aplikasi mobile passing query sesuai kebutuhan ke webservice. Selanjutnya webservice passing query ke database server. Setelah passing query, webservice akan menerima respon dari database server dan respon tersebut dipassing kembali ke aplikasi mobile untuk diolah sesuai fungsinya (Gambar 2.b).

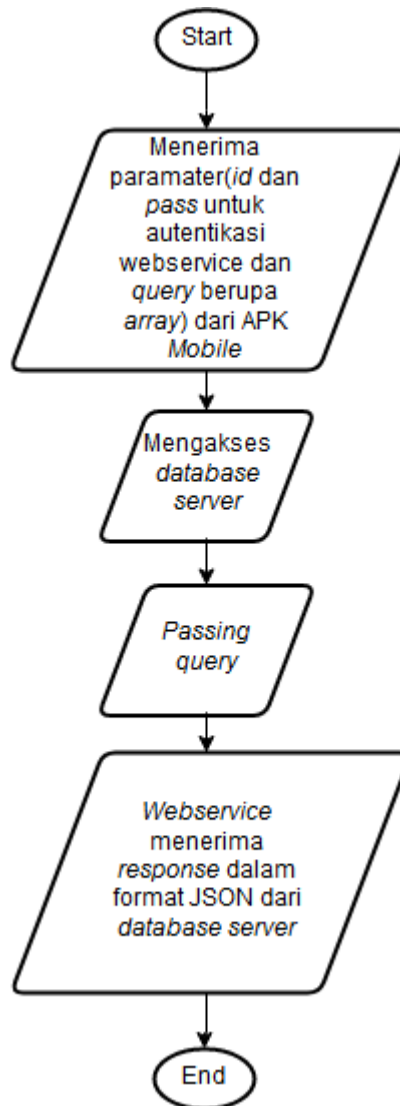
*Web service* digunakan untuk melakukan pertukaran data antar aplikasi atau sistem dan melakukan pertukaran data antara database dengan *mobile app*. *Web Service* yang digunakan pada makalah ini dibagi menjadi tiga fungsi yaitu mengunduh gambar, menghapus gambar dan mengambil data.



Gambar 3. a. Diagram Alir Mengunduh Gambar b. Diagram Alir Menghapus Gambar

Pada proses mengunduh gambar melalui *webservice* (Gambar 3.a), *webservice* menerima parameter nama *file* yang ingin diunduh. Selanjutnya *webservice* mengakses folder di server melalui FTP dan mengambil gambar sesuai parameter. Setelah itu gambar diubah menjadi *byte array* dan dikirimkan ke aplikasi *mobile*. Sedangkan pada Gambar 3.b proses menghapus gambar pada *webservice*, *webservice* menerima parameter nama *file* yang ingin dihapus. Selanjutnya *webservice* mengakses folder melalui FTP dan menghapus gambar sesuai

parameter. Setelah itu *webservice passing query* ke *database server* untuk menghapus data sesuai parameter. Aplikasi *mobile* akan menerima respon setelah semua fungsi dijalankan dari *webservice*.



Gambar 4. Diagram Alir Mengambil Data

Pada Gambar 4 proses mengambil data pada *webservice*, *webservice* akan menerima parameter *id* dan *password* untuk proses autentikasi pada *webservice* dan *query* berupa array. Selanjutnya *webservice* akan mengakses *database server* dan *passing query* sesuai parameter yang diterima. Setelah itu *webservice* akan menerima respon dari database dan mengirimkannya ke aplikasi *mobile*.

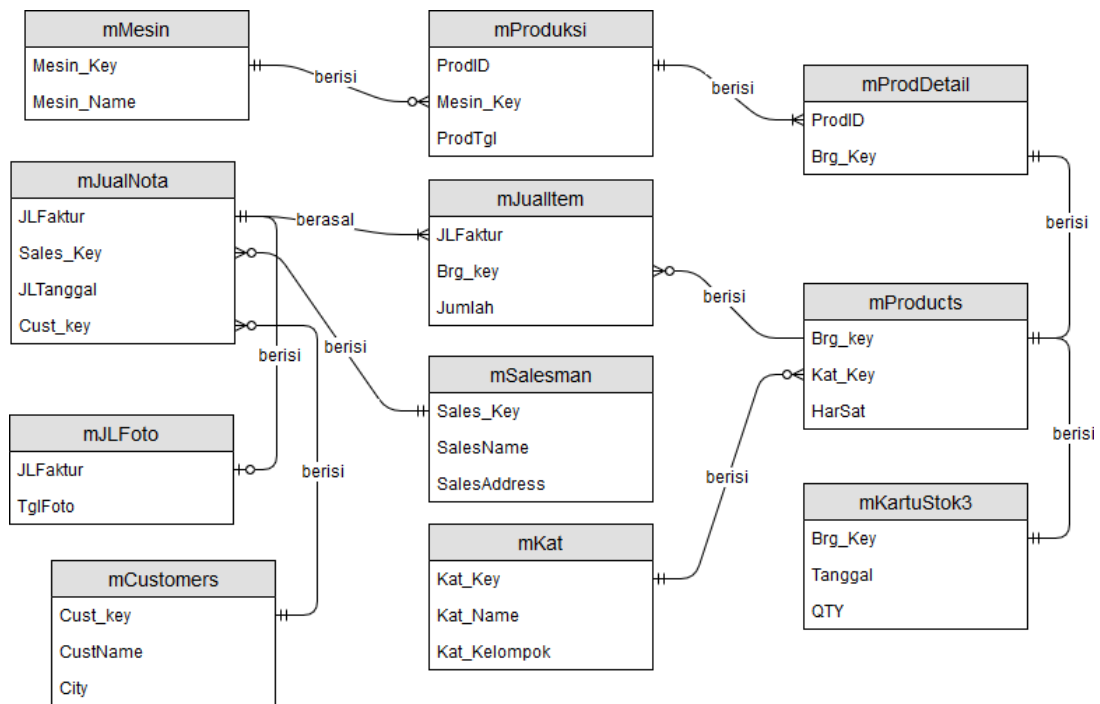
*Database* yang digunakan makalah ini dibagi menjadi dua bagian yaitu dengan *Firestore* dan *Microsoft SQL Server*. *Firestore* digunakan untuk menyimpan data *user* yang digunakan untuk proses autentikasi ketika masuk aplikasi Android sedangkan *Microsoft SQL Server* digunakan untuk menyimpan informasi perusahaan. Gambar 5.a menunjukkan struktur data

pada Firebase, sedangkan Gambar 5.b menunjukkan rancangan *database user* dalam bentuk tabel.



Gambar 5. a. Rancangan pada Firebase b. Rancangan Database User

Database Microsoft SQL Server digunakan untuk menyimpan data perusahaan yang meliputi penjualan, retur, produksi dan barang *reject*. Gambar 6 merupakan ERD (*Entity Relationship Diagram*) database dan tidak semua atribut ditulis, hanya atribut yang digunakan pada makalah ini saja.

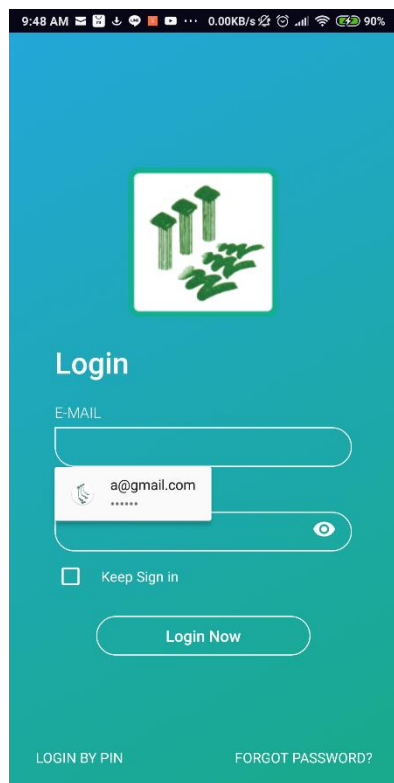


Gambar 6. Entity Relationship Diagram pada MSSQL

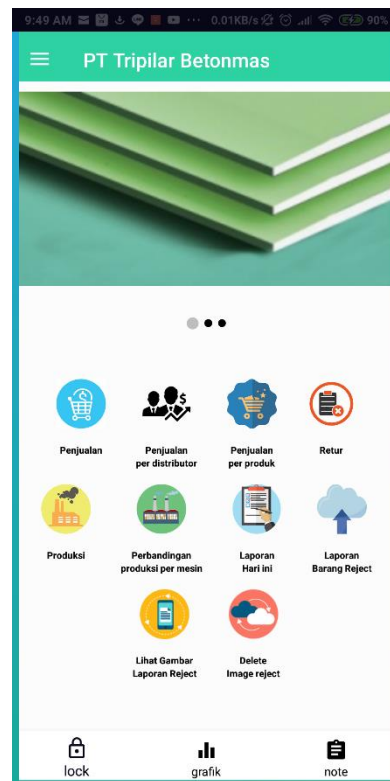
### 3. Hasil dan Analisis

Pengujian dilakukan dengan cara menguji fungsi-fungsi pada aplikasi dengan OS (Operating System) yang berbeda dan menguji aplikasi menggunakan jaringan internet.

User Interface pada aplikasi yang dirancang terbagi menjadi 4 bagian, yaitu *Login Activity*, *MainMenu Activity*, *SubMenu Activity* dan *Function Menu Activity*. *Login Activity* digunakan untuk keamanan aplikasi *mobile* supaya hanya bisa diakses oleh *user* yang memiliki *id* dan *password*. *MainMenu Activity* berfungsi untuk menghubungkan antar *Activity* yang ada pada aplikasi *mobile*. *SubMenu Activity* berfungsi untuk memberikan filter pada fungsi yang ingin dijalankan. Sedangkan *Function Menu Activity* berfungsi untuk melihat data yang melalui grafik, melihat data bukti retur berupa foto.

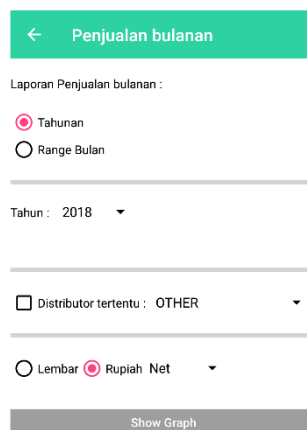


(a)



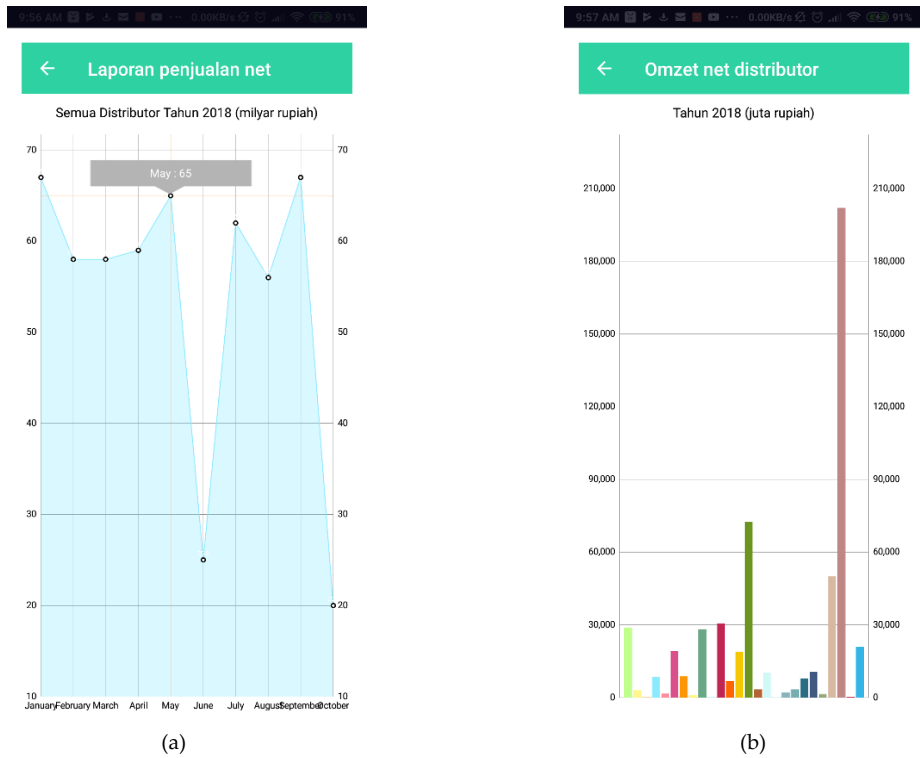
(b)

Gambar 7. a. User Interface Login Activity b. User Interface Main Menu Activity

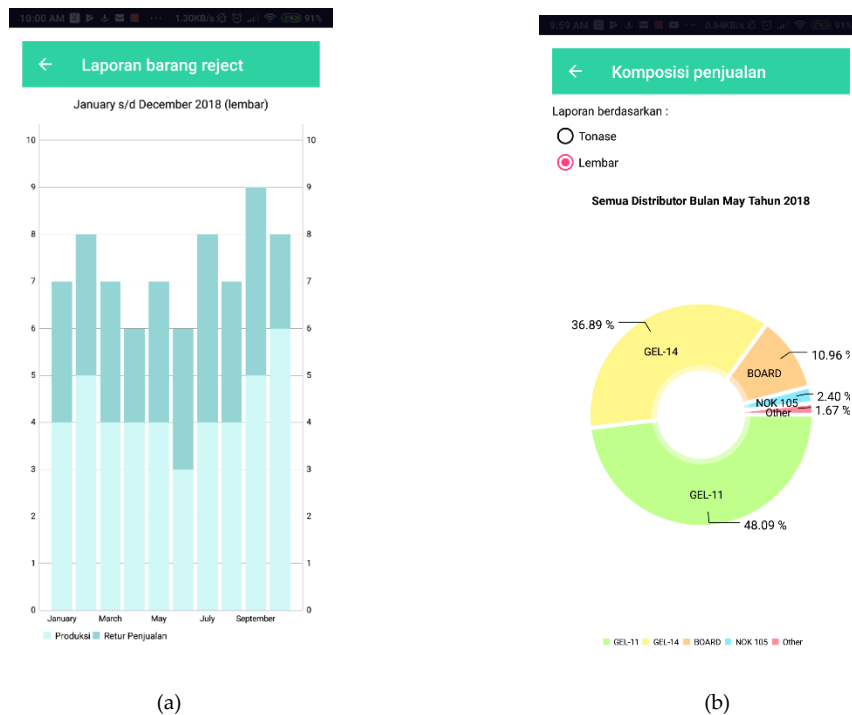


Gambar 8. User Interface Sub Menu Activity

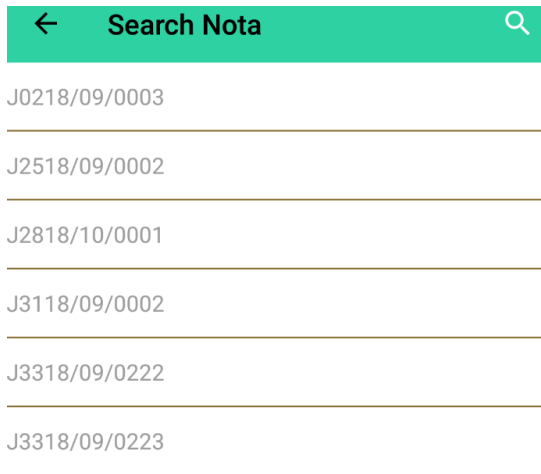




Gambar 9. a. User Interface Function Menu Activity Line Chart, b. User Interface Function Menu Activity Bar Chart



Gambar 10. a. User Interface Function Menu Activity Stackable Bar Chart, b. User Interface Function Menu Activity Pie Chart



(a)



(b)

Gambar 11. a. User Interface Sub Menu Activity Search Nota, b. User Interface Function Menu Activity Retour

Pada pengujian dibawah ini menggunakan metode pengujian *blackbox*, yang bertujuan untuk menemukan fungsi-fungsi yang tidak benar. Pada pengujian kompatibilitas, aplikasi dijalankan pada OS Jelly Bean - Oreo untuk dapat mengetahui kompatibilitas aplikasi. Pengujian dilakukan dengan menjalankan aplikasi pada OS Jelly Bean – Oreo dengan menjalankan semua fungsi pada aplikasi.

Tabel 2. Hasil Pengujian *Operating System*

Operating System	Error	Device
Jelly Bean	Fingerprint authentication & mengambil gambar untuk bukti retur menggunakan kamera	Emulator
KitKat	Fingerprint authentication & mengambil gambar untuk bukti retur menggunakan kamera	Emulator
Lollipop	Fingerprint authentication & mengambil gambar untuk bukti retur menggunakan kamera	Emulator
Marshmallow	-	Xiaomi Redminote 3
Nougat	Path penyimpanan data default URI file://	Redmi 5 plus
Oreo	Path penyimpanan data default URI file://	Xiaomi Redminote 5

Pengujian jaringan dilakukan untuk menguji jaringan yang digunakan untuk menjalankan aplikasi menggunakan Internet. Pengujian dilakukan dengan cara melakukan fungsi – fungsi yang ada pada aplikasi.

Tabel 3. Hasil Pengujian Pengambilan Data Berupa *Text*

Percobaan	Menggunakan jaringan <i>Wifi</i> (Indihome) 400Kb/s	Menggunakan Operator selular (Indosat)200Kb/s
1	4 s	7 s
2	3 s	9 s
3	5 s	4 s
4	4 s	5 s
5	2 s	6 s
6	5 s	7 s
7	4 s	9 s
8	3 s	5 s
9	5 s	4 s
10	2 s	7 s

Tabel 3 menunjukkan waktu yang dibutuhkan aplikasi untuk mengambil data penjualan. Waktu yang dibutuhkan berbeda-beda tergantung dari koneksi Internet yang digunakan, hal ini dipengaruhi konektivitas dari setiap operator penyedia layanan internet.

Tabel 4. Pengujian Jaringan Mengunggah dan Mengunduh Gambar

Ukuran Gambar	Waktu Mengunduh Gambar (s)		Waktu Mengunggah Gambar (s)	
	<i>Wifi</i> (indihome)	Indosat	<i>Wifi</i> (indihome)	Indosat
100kb	21	42	18	58
200kb	35	57	27	Gagal
300kb	47	Gagal	39	Gagal
400kb	58	Gagal	51	Gagal
500kb	Gagal	Gagal	Gagal	Gagal
600kb	Gagal	Gagal	Gagal	Gagal

Jaringan *Wifi* Indihome memiliki kecepatan *up to 400kb/s* dan jaringan seluler (Indosat) memiliki kecepatan *up to 200kb/s* pada Tabel 4. Semakin tinggi kecepatan jaringan yang digunakan akan semakin tinggi pula tingkat keberhasilan mengunduh ataupun mengunggah gambar.

#### 4. Kesimpulan

*Operating system* Jelly Bean hingga Lollipop belum memiliki fitur *fingerprint authentication*, sehingga untuk fitur *fingerprint authentication* hanya bisa dijalankan setelah *operating system* Marshmallow. Aplikasi yang dirancang dapat berjalan dari OS Jelly Bean hingga Oreo. Pengambilan data pada aplikasi ini berupa JSON dan aplikasi dapat menyimpan ataupun mengunduh gambar pada aplikasi yang telah dirancang. Pada proses pengujian, semakin besar data yang ingin diambil maka waktu yang dibutuhkan semakin lama untuk memprosesnya.

## Daftar Pustaka

- [1] Mobile Operating System Market Share Worldwide, [online]. <http://gs.statcounter.com/os-market-share/mobile/worldwide>. [Diakses pada 7 Oktober 2018]
- [2] Burnette, Ed, *Hello, Android(4<sup>th</sup> edition) Introducing Google's Mobile Development Platform*, 2015.
- [3] Petkovic, Dusan, *Microsoft SQL Server 2016: A Beginner's Guide, Sixth Edition*, 2016.
- [4] Developer Android, [online]. <https://developer.android.com/guide/components/activities/activity-lifecycle> [Diakses pada 3 Juni 2018]
- [5] Firebase, [online]. <https://firebase.google.com/?hl=id>. [Diakses pada 9 Juni 2018]
- [6] Firebase Realtime Database, [online]. <https://firebase.google.com/docs/database/?hl=id>. [Diakses pada 9 Juni 2018]
- [7] Firebase Authentication, [online]. <https://firebase.google.com/docs/auth/?hl=id>. [Diakses pada 9 Juni 2018]
- [8] Ari-Koponen, Pekka. *A Secure OAuth 2.0 Implementation Model*, 2016.
- [9] Sam Ruby, Leonard Richardson, *RESTful Web Services*, 2007.