JOURNAL OF INFORMATION SYSTEM, GRAPHICS, HOSPITALITY AND TECHNOLOGY

Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang

Gerry Surya Chandra, *Departemen Informatika*, *Institut Sains dan Teknologi Terpadu Surabaya*, dan Suhatati Tjandra, *Departemen Informatika*, *Institut Sains dan Teknologi Terpadu Surabaya*

***Abstrak*—Pemesanan air minum dalam kemasan sekarang masih menggunakan telepon, yaitu customer menelepon toko untuk membuat pesanan. Customer perlu menyebutkan alamat pengiriman, dan barang yang akan dipesan. Cara ini sungguh tidak efisien, karena banyak customer yang memesan setiap hari dengan alamat dan barang yang sama. Dengan adanya aplikasi pengiriman dan inventori yang dibuat ini, pengguna dapat mempermudah membuat pesanan terhadap pemilik distributor air mineral dalam kemasan.**

Flutter adalah SDK untuk pengembangan aplikasi mobile dengan kinerja tinggi, aplikasi untuk iOS dan Android, dari satu codebase (basis kode) yang di buat oleh Google dengan lisensi open source. Electron adalah framework untuk membuat aplikasi desktop dengan JavaScript murni dengan menyediakan API dari OS. Pada aplikasi ini terdapat empat aktor yaitu, admin, manajer pengiriman, kurir, dan customer. Yang mendasar aplikasi ini adalah back-end, yaitu sebuah API yang bertugas untuk mengelola data. Back-end pada aplikasi ini menggunakan framework CodeIgniter dan MySQL sebagai database, kemudian aplikasi-aplikasi yang dipakai oleh aktor baru bisa berjalan dengan baik. Kurir menggunakan aplikasi pada Android yang dibuat menggunakan framework Flutter. Customer menggunakan aplikasi pada Android yang dibuat menggunakan framework Flutter. Admin menggunakan aplikasi yang dibuat menggunakan framework Electron dan Angular pada komputer. Manajer pengiriman menggunakan aplikasi yang dibuat menggunakan framework Electron dan Angular, manajer pengiriman menggunakan Raspberry Pi untuk menjalankan aplikasi, dan menggunakan balenaCloud untuk mengelola perangkat.

***Kata Kunci*—Electron, Flutter, Inventori, Pengiriman.**

I. PENDAHULUAN

Saat ini banyak distributor air minum dalam kemasan masih menggunakan pencatatan pesanan kiriman menggunakan metode manual, yaitu mencatat semua pesanan di nota dengan format tanggal, jenis barang, kuantitas, dan harga. Sistem lama ini masih bisa terkendali bila pesanan tidak melebihi 20 nota, ketika melebihi itu, pemilik kesulitan untuk mengikuti kecepatan dari cepatnya pesanan masuk. Dan akhirnya customer yang menjadi korban dari sistem ini.

Inventori pada banyak distributor air minum dalam kemasan tidak tercatat dengan baik. Meskipun, sudah ada sistem untuk menghitung stok barang, tetapi tidak berfungsi dengan baik, karena banyak stok yang opname, contohnya ada karton yang rusak dan harus menjual dengan cara eceran sehingga stok menjadi tidak cocok. Pada beberapa kasus, distributor air minum dalam kemasan juga sering mengalami kehabisan stok barang karena tidak mengetahui sisa stok. Sistem ini sangat tidak mendukung operasional.

Berdasarkan perkembangan teknologi dan masalah di atas, memungkinkan untuk pemilik toko bisa memiliki sistem yang efisien dengan menggunakan teknologi komputer. Untuk pengiriman, pembeli melakukan pesanan dengan cara menelepon nomor telepon distributor dengan memberi tahu alamat, dan barang yang dipesan. Sistem ini tidak efisien karena pembeli biasanya adalah pelanggan yang melakukan pesanan setiap hari, jadi pembeli harus memberitahu alamat setiap kali menelepon. Pada proses berikutnya, toko menulis nota yang berisi alamat dari pembeli dan barang yang dipesan. Lalu nota di berikan kepada kurir untuk dikirim. Kurir memikirkan rute yang tercepat, dan menyiapkan barang yang akan dikirim. Lalu kurir akan menggunakan motor gerobak untuk mengirim.

Akan dibuat sistem inventori dan aplikasi pengiriman air minum dalam kemasan secara online berbasis aplikasi. Aplikasi tersebut dibuat untuk mempermudah pemilik dalam masalah di atas. Aplikasi ini juga akan memberikan laporan penjualan secara berkala dan menampilkan barang yang stoknya sudah sisa sedikit. Sehingga manfaatnya pemilik tidak menghabiskan waktu untuk mengecek inventori dan menghitung setiap barang.

II. TEORI DASAR

Dalam pengembangan aplikasi ini digunakan software, framework, library dan bahasa pemrograman. Berikut teori-teori dasar yang digunakan

A. Flutter

Flutter adalah SDK untuk pengembangan aplikasi mobile dengan kinerja tinggi, aplikasi untuk iOS dan Android, dari satu codebase (basis kode) yang di buat oleh Google dengan lisensi open source. Tujuannya adalah memungkinkan pengembang untuk menghadirkan aplikasi berkinerja tinggi

Oktober 2020
Gerry Surya Chandra, Departemen Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: suryagerry@gmail.com)
Suhatati Tjandra, Departemen Informatika dan Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: tati@stts.edu)

yang terasa alami pada platform yang berbeda.

Untuk membuat aplikasi Flutter, diperlukan untuk mengerti bahasa Dart. Dart merupakan bahasa pemrograman yang dibuat oleh Google untuk menggantikan Javascript. Dart menggunakan static typing yang berarti sebelum memakai variabel, variabel perlu didefinisikan terlebih dahulu. Dart bisa berjalan pada semua perangkat juga, pada web Dart memakai dart2js yang artinya Dart diubah ke Javascript agar bisa dimengerti browser. Pada perangkat desktop Dart memakai dart2aot yang mengubah Dart menjadi bahasa mesin. Pada perangkat mobile Dart memakai Flutter. Syntax pada bahasa Dart sangat mudah dipelajari, karena Dart memiliki kemiripan dengan syntax-syntax pemrograman lain, seperti Javascript dan Java.

Dart bisa melakukan pemrograman Asynchronous, yaitu fungsi yang non-blocking yaitu memungkinkan program untuk menjalankan kode lain sambil menunggu fungsi asynchronous selesai. Contoh penggunaan sehari-hari adalah pada saat aplikasi sedang mengambil data dari web, aplikasi juga menampilkan halaman loading. Dart juga bisa melakukan pemrograman Reactive, yaitu pemrograman asynchronous dengan stream data. Stream data artinya datanya bisa di terima lebih dari sekali, kebalikan dengan asynchronous yang datanya hanya sekali terima saja. Pada Dart asynchronous menggunakan kata Future, reactive menggunakan Stream. Pada Flutter ada 2 cara dalam menjalankan aplikasi [2], yaitu:

- Debug
  Flutter akan berjalan secara JIT (Just in Time), aplikasi akan dicompile pada saat aplikasi berjalan. Dengan menggunakan debug, dapat beberapa fitur yang sangat diperlukan oleh pengembang aplikasi yaitu, assert dinyalakan, observatory dinyalan berguna untuk debug. Service extension dinyalakan, kompilasi dioptimisasi untuk pengembangan (sehingga tidak dioptimisasi untuk kecepatan, ukuran aplikasi). Pada debug Flutter juga mempunya fitur hot reload dan hot restart. Hot reload adalah melakukan penerapan ulang pada aplikasi tanpa membuang state. Hot restart adalah seperti menutup aplikasi dan membuka lagi, tetapi hanya bagian Flutternya saja, yang pastinya akan mereset state.
- Release
  Flutter akan berjalan secara AOT (Ahead of Time), sehingga aplikasi perlu dicompile terlebih dahulu. Dengan menggunakan release, Flutter akan mematikan fungsi assert, informasi debug dihapus, debug dimatikan, kompilasi dioptimisasi untuk kecepatan dan ukuran aplikasi, service extension dimatikan.

Flutter menggunakan konsep widget untuk membuat UI nya. Semua UI adalah terdiri dari widget-widget. Contohnya adalah RaisedButton widget, ListView widget, DateTimePicker widget, TabBar widget, Text widget, Label widget, dan banyak widget lainnya. Karena konsep semua UI adalah widget, misalnya isi dari tombol bisa diberi widget lain, contohnya memberi gambar, tulisan dan memberi tombol lagi. Meskipun hasilnya akan aneh, tetapi Flutter tetap bisa mengambarnya. Widget pada Flutter mempunyai dua tipe yaitu:

- Stateful widget
  Stateful widget adalah widget bisa melakukan gambar ulang jika ada perubahan data. Menggambar ulang widget menggunakan method setState(). Cara ini mirip dengan konsep React.
- Stateless widget
  Stateless widget adalah widget yang tidak bisa melakukan gambar ulang.

Flutter mempunyai dua macam widget untuk pengembang aplikasi pakai, yaitu Material Design dan Cupertino. Material Design adalah bahasa design yang dibuat oleh Google, design ini sama dengan design yang dipakai pada Android. Cupertino atau dengan sebutan lain gaya iOS adalah bahasa design yang dipakai oleh iOS. Flutter mempunyai lebih banyak widget Material Design daripada widget Cupertino, tetapi tidak usah khawatir, pada OS perangkat yang berbeda, widget bisa dipakai secara cross-platform. Flutter pada digunakan untuk membuat aplikasi customer dan kurir.

B. Electron

Electron adalah framework untuk membuat aplikasi desktop dengan JavaScript murni dengan menyediakan API dari OS. Electron berhasil mencapai ini dengan cara menggabungkan Chromium dan Node.JS menjadi satu dengan kata lain Electron adalah browser yang bisa menjalankan Node.JS, sehingga Electron dapat berjalan pada macOS, Windows, dan Linux. Electron dikembangkan oleh GitHub, awalnya Electron adalah framework yang digunakan oleh Atom dengan nama Atom Shell, yang lalu berubah nama menjadi Electron. [1]

Pengembangkan Electron dimulai pada 2013, lalu pada 2014 Electron mulai menjadi open source. Dengan menggunakan Electron membuat aplikasi desktop hanya menggunakan HTML, Javascript, CSS saja, untuk mengakses level sistem OS sudah disediakan API oleh Electron. Aplikasi yang dibuat menggunakan Electron nantinya bisa dimasukkan ke Mac App Store dan Windows Store. Tetapi untuk memasukkan tempat pendistribusian aplikasi diperlukan syarat-syarat yang harus dipenuhi dari tempat pendistribusian aplikasi.

Arsitektur Electron terdiri dari dua yaitu main process (proses utama) dan renderer process. Main process digunakan untuk menampilkan GUI dengan membuat halaman web. Aplikasi Electron selalu memiliki satu proses utama, tetapi tidak pernah lebih. Karena Electron menggunakan Chromium untuk menampilkan halaman web, arsitektur multi-process Chromium juga digunakan. Setiap halaman web dalam Electron berjalan dalam prosesnya sendiri, yang disebut renderer process. Pada browser biasa, halaman web biasanya berjalan di sandbox dan tidak diizinkan mengakses sumber daya dari OS. Electron memiliki kekuatan untuk menggunakan API Node.js di halaman web yang memungkinkan interaksi level rendah OS.

Main process membuat halaman web dengan membuat instance BrowserWindow. Setiap instance BrowserWindow menjalankan halaman web dalam proses renderernya sendiri. Ketika instance BrowserWindow dihancurkan,

renderer process terkait juga dihentikan. Main process mengelola semua halaman web dan proses renderer yang sesuai. Setiap renderer process terisolasi dan hanya peduli tentang halaman web yang berjalan di dalamnya. Electron digunakan untuk membuat aplikasi admin dan manajer pengiriman.

C. *Google Maps Platform*

Google Maps Platform adalah sekumpulan API dan SDK yang dikelola dari Google Cloud Platform Console. Google Maps banyak disokong oleh banyak perangkat, contohnya web browser, android, iOS, dan lain-lain. Untuk mulai penggunaan Google Maps Platform, diperlukan akun billing, API/SDK Google Maps Platform, dan minimal satu API key. Google Maps Platform digunakan untuk menampilkan peta, mencari rute tercepat, dan menampilkan alamat.

D. *Moota*

Moota adalah layanan untuk pengecekan mutasi dan saldo rekening, di mana mutasi rekening didapatkan dari akun internet banking. Moota tidak menjamin akun internet banking jika terblokir, karena cara kerja Moota adalah mengambil data yang ada di internet banking lalu disimpan di server Moota. Jadi penggunaan Moota bisa saja membuat akun internet banking terblokir. Moota bisa mengambil data dari internet banking setiap 15/30/45/60 menit sekali, yang bisa di atur pada pengaturan nomor rekening. Moota digunakan untuk mengecek customer sudah melakukan transfer secara otomatis.[3]

E. *Docker*

Docker adalah layanan untuk SaaS dan PaaS yang menggunakan container virtualisasi tingkat sistem operasi dalam mendeploy kode.[4] Docker awalnya bernama dotCloud yang berganti nama pada 29 Oktober 2013 . Container dalam docker adalah gabungan dari perangkat lunak, library, dan file konfigurasi yang terisolasi dari kontainer satu dengan yang lain. Meskipun terisolasi, container-container tetap bisa berkomunikasi dengan kontainer yang lain, biasanya melalui jaringan. Agar bisa menggunakan container, sebelumnya harus dibuat dahulu Docker imagenya. Docker image dibuat dari kode yang telah dibuat oleh pengembang aplikasi yang telah digabungkan dengan image-image lain. Docker digunakan untuk membuat image pada Balena Cloud.

F. *Balena Cloud*

Balena adalah seperangkat alat lengkap untuk membangun, menyebarkan, dan mengelola perangkat-perangkat Linux yang terhubung dengan internet. Balena menyediakan infrastruktur untuk pemilik perangkat-perangkat Linux sehingga pengembang dapat fokus pada pengembangan aplikasi dan mengembangkan perangkat dengan gesekan sesedikit mungkin. Sebelum Balena Cloud, dulu nama Balena Cloud adalah Resin. Resin dikenal sebagai alat untuk menyebarkan aplikasi untuk perangkat keras saja. Karena dirasa kurang tepat dengan nama resin, maka diganti menjadi Balena Cloud, yang juga melayani pengelolaan perangkat keras agar pengembang lebih mementingkan aplikasi. Balena Cloud digunakan untuk alat deploy kode pada aplikasi manajer.

III. METODE DAN TAHAPAN PENELITIAN

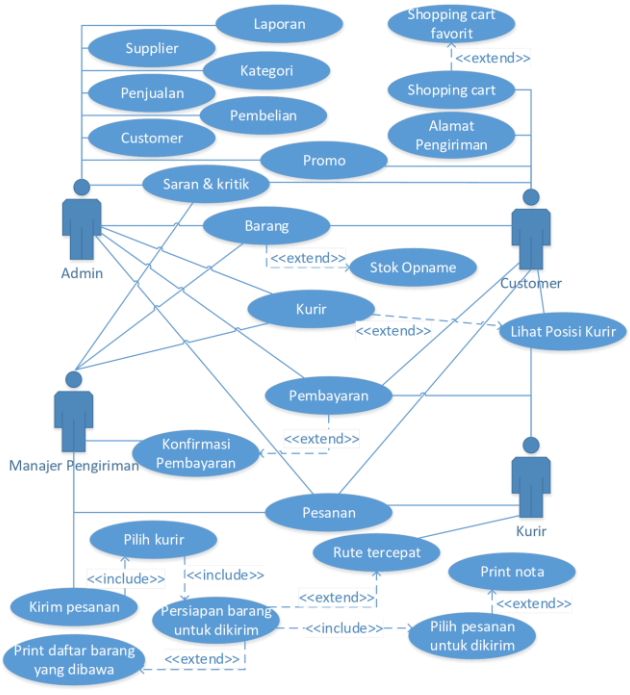
Adapun tahapan penyelesaian penelitian ini:

1) Melakukan uji coba untuk setiap API yang akan digunakan (Distance API, Moota, dll.).
2) Melakukan instalasi program-program yang dibutuhkan seperti PHP, Flutter, Codeigniter, dsb.
3) Membangun database.
4) Membangun web service.
5) Mendesain tampilan aplikasi customer dan admin kemudian menghubungkan dengan web service yang telah dibuat sebelumnya.
6) Melakukan uji coba ketika per modul digabung dan uji coba seluruh fitur ketika semua modul telah digabungkan.

Berikut ini akan dijelaskan model sistem dan desain interface. Penjelasan akan dibagi menjadi beberapa bagian yang terpisah.

A. *Model Sistem*

Model sistem digambarkan dengan use case diagram yang menjelaskan fitur-fitur apa saja yang dapat digunakan oleh masing-masing user. Pada aplikasi ini, terdapat 4 user yaitu admin, manajer pengiriman, customer, dan kurir.



Gambar. 1. Diagram use case

Pada gambar 1 merupakan usecase diagram yang menggambarkan fitur-fitur yang dimiliki setiap user. Fitur utama customer adalah membuat pesanan. Selain itu customer juga dapat mengatur alamat customer, mempunyai keranjang belanja favorit, dan melihat promo. Customer juga dapat melihat lokasi kurir pada saat kurir sedang perjalanan ke alamat customer.

Manajer pengiriman mengatur pesanan yang akan dikirim dan dibuatkan shift. Pertama, manajer pengiriman memilih

pesanan yang akan dikirim, kemudian mengurutkan pesanan. Mengurutkan pesanan bisa dengan cara otomatis urutan rute terbaik, atau bisa manual mengurutkan pesanan. Setelah selesai membuat shift, manajer pengiriman dapat mencetak nota pesanan dan cetak barang yang akan dibawa.

Fitur utama kurir adalah melakukan pengiriman. Awalnya kurir memilih shift yang sudah dibuat oleh manajer pengiriman, kemudian kurir dapat melihat semua pesanan. Kemudian kurir mengaktifkan shift dan lokasi tracking menyala. Kurir dapat melakukan navigasi pada peta ke alamat customer.

Admin dapat mengatur data-data inventori dan pengiriman seperti data barang, merek, tipe barang, customer, pembelian, penjualan, dan lain-lain. Selain itu admin juga dapat memasukkan data penjualan dari mesin kasir melalui kartu SD. Dan admin juga dapat memasukkan data barang dari mesin kasir melalui kartu SD juga.

B. API

API yang digunakan merupakan Direction API dan Moota. Direction API digunakan untuk mendapatkan rute tercepat pada saat mengurutkan pesanan. Contoh penggunaan Direction API pada library Google Maps pada Node.js adalah sebagai berikut:

```
googleMapsService.client.directions({                    (1)
   origin: Config.myLocation,
   destination: Config.myLocation,
   region: 'id',
   waypoints
})
```

Dimana:
origin = koordinat berangkat,
destination = koordinat tujuan, variabel
region = lokasi negara, supaya Google Maps lebih bagus untuk memberi arahan.
waypoints = variabel array yang berisi titik koordinat yang akan dituju.

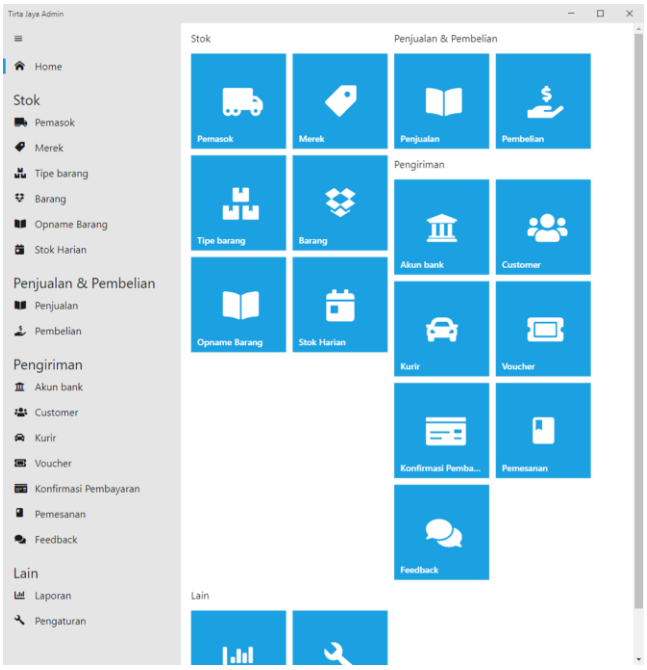
Moota digunakan untuk mengambil mutasi terbaru dari rekening bank. Contoh penggunaan Moota menggunakan GET request sebagai berikut:

```
https://app.moota.co/api/v1/bank/{bank_id}/muta      (2)
tion/recent/{jumlahMutasi}
```

Dimana:
bank_id = id bank yang telah diberikan oleh Moota pada saat mendaftarkan rekening bank.
jumlahMutasi = banyaknya mutasi terbaru yang ingin diambil.

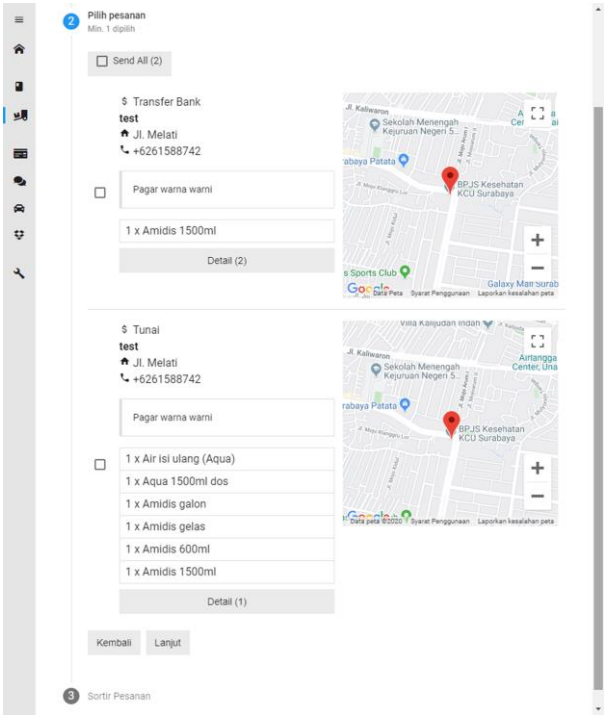
C. Desain Interface

Pada subbab ini akan dibahas desain interface yang ada pada aplikasi. Pada gambar 2 merupakan gambar yang menampilkan tampilan halaman home pada admin. Pada halaman ini admin dapat memilih menu yang akan ditekan dan halaman akan berpindah.



Gambar. 2. Tampilan halaman home pada admin

Pada gambar 3 merupakan gambar yang menampilkan manajer pengiriman sedang membuat shift dan memilih pesanan. Pada halaman ini manajer pengiriman dapat menekan kotak centang untuk memilih pesanan yang hendak dikirim.

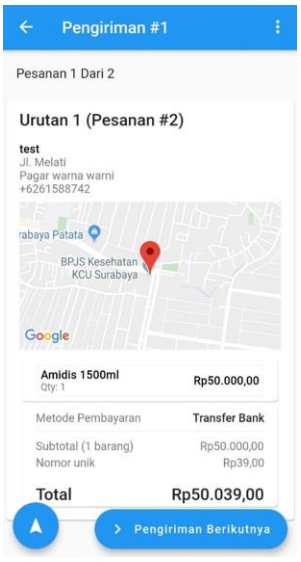


Gambar. 3. Tampilan halaman kirim pesanan pada manajer pengiriman

Pada gambar 4 merupakan gambar yang menampilkan customer pada halaman belanja. Customer dapat memilih barang yang akan dibeli dengan cara menekan tombol tambah untuk menambah kuantitas yang akan dibeli.

Gambar. 4. Tampilan halaman belanja pada customer

Pada gambar 5 merupakan gambar yang menampilkan kurir pada halaman pengiriman. Halaman ini berguna untuk melihat pengiriman yang sedang dikirim sekarang. Terdapat tombol navigasi untuk membuka peta dan melakukan navigasi ke alamat customer. Selain itu, terdapat tombol pengiriman berikutnya untuk melanjutnya pengiriman.



Gambar. 5. Tampilan halaman pengiriman pada kurir

IV. HASIL PENELITIAN

Pada bab ini akan dijelaskan hasil uji coba yang telah dilakukan. Uji coba dibagi menjadi dua metode yaitu blackbox dan kuesioner. Untuk admin, manajer pengiriman, dan kurir menggunakan metode blackbox. Untuk customer menggunakan metode kuesioner yaitu dengan cara memberikan kuesioner setelah pengguna memakai aplikasi.

A. Hasil uji coba admin

Uji coba ini dilakukan bersama dengan pemilik toko Tirta Jaya.

TABEL I
HASIL UJI COBA ADMIN

Modul	Hasil Uji Coba	Tanggapan
Pemasok	User tidak mengalami kesulitan	
Tipe Barang	User tidak mengalami kesulitan	
Barang	User tidak mengalami kesulitan	
Opname Barang	User tidak mengalami kesulitan	
Stok Harian	- User menginginkan stok harian mempunyai kuantitas masuk dan kuantitas keluar. - User menginginkan halaman stok barang - User menginginkan catatan pada kuantitas adalah tanggal pembelian/penjualan bukan id pembelian/penjualan - Terdapat bug pada saat sort kolom waktu, sehingga sort sesuai dengan alfabet.	- Penambahan kolom kuantitas masuk dan kuantitas keluar. - Stok harian akan diubah menjadi stok barang. - Stok harian dibuat untuk melihat stok keluar masuk pada satu hari itu. - Memperbaiki bug sort kolom waktu.
Penjualan	User tidak mengalami kesulitan	
Pembelian	User tidak mengalami kesulitan	
Akun Bank	User tidak mengalami kesulitan	
Customer	User tidak mengalami kesulitan	
Kurir	User tidak mengalami kesulitan	
Voucher	User tidak mengalami kesulitan	
Konfirmasi Pembayaran	User tidak mengalami kesulitan	
Pemesanan\	User tidak mengalami kesulitan	
Feedback	User tidak mengalami kesulitan	
Laporan	User tidak mengalami kesulitan	

B. Hasil uji coba manajer pengiriman

Uji coba ini dilakukan bersama dengan pemilik toko Tirta Jaya.

TABEL II
HASIL UJI COBA MANAJER PENGIRIMAN

Modul	Hasil Uji Coba	Tanggapan
Pesanan	User tidak mengalami kesulitan	
Kirim Pesanan	Terdapat bug pada saat ada text yang kosong, print akan gagal.	Memperbaiki bug print gagal.
Konfirmasi Pembayaran	User tidak mengalami kesulitan	
Feedback	User tidak mengalami kesulitan	
Kurir	User tidak mengalami kesulitan	
Barang	User tidak mengalami kesulitan	

C. *Hasil uji coba kurir*

Uji coba ini dilakukan bersama dengan pemilik toko Tirta Jaya. Aplikasi kurir bisa diunduh pada https://play.google.com/store/apps/details?id=id.co.tirtajaya.courier atau dengan melakukan pencarian pada Google Play Store dengan nama aplikasi "Tirta Jaya Kurir" dan nama pengembang "Gerry Surya".

TABEL III
HASIL UJI COBA KURIR

Modul	Hasil Uji Coba	Tanggapan
Login	User tidak mengalami kesulitan	
Shift	User tidak mengalami kesulitan	
Delivery	User tidak mengalami kesulitan	

D. *Hasil kuesioner customer*

Kuesioner diisi oleh 19 responden, di mana responden merupakan teman-teman penulis. Aplikasi customer bisa diunduh pada url https://play.google.com/store/apps/details?id=id.co.tirtajaya.customer.

TABEL IV
HASIL UJI COBA CUSTOMER

No	Pertanyaan	Sangat Buruk	Buruk	Baik	Sangat Baik
1	Apakah aplikasi nyaman digunakan?	0%	10,5%	52,6%	36,8%
2	Apakah aplikasi mudah digunakan?	0%	5,3%	52,6%	42,1%
3	Apakah tampilan aplikasi mudah dimengerti?	0%	5,3%	52,6%	42,1%
4	Apakah Anda terbantu dengan adanya aplikasi ini?	0%	5,3%	52,6%	42,1%

V. KESIMPULAN

Pada bab ini akan dijelaskan mengenai kesimpulan yang didapatkan dari pembuatan aplikasi. Kesimpulan tersebut antara lain:

- Aplikasi sudah menjawab kebutuhan yang memerlukan aplikasi dengan manajemen stok dan pengiriman, tetapi ada beberapa fitur yang masih bisa dikembangkan lebih lanjut, seperti penyempurnaan pada aplikasi manajer pengiriman di modul pengiriman, supaya bisa dilakukan perubahan urutan pengiriman.
- Penggunaan Firebase sangat membantu dalam pengembangan yaitu untuk pembuatan autentikasi, manajemen data pengguna, pengamanan data pengguna. Keamanan dari Firebase tidak perlu diragukan karena Firebase merupakan perusahaan milik Google/Alphabet, kecuali ada bug pada aplikasi yang menyebabkan bisa mengakses data-data pada Firebase.
- PHP dengan framework CodeIgniter tidak memberikan static typing, sehingga pada saat penggunaan fungsi bisa terjadi salah ketik atau salah tipe data dan membuat eror pada aplikasi.
- Pengembangan fitur cek transfer otomatis menggunakan layanan Moota mengalami kesulitan karena Moota tidak menyediakan fitur sandbox, sehingga untuk melakukan uji coba sedikit mengalami kesulitan, uji coba hanya bisa dilakukan dengan cara melakukan transfer sebenarnya.
- Pada aplikasi customer, perlu ditambahkan sistem cache sehingga setiap kali aplikasi terbuka, sehingga tidak perlu mengambil data dari server pada saat aplikasi terbuka.

DAFTAR PUSTAKA

[1] Anon., 2019. Documentation | Electron. [Online] Available at: https://electronjs.org/docs/
[2] Anon., 2019. Flutter Documentation. [Online] Available at: https://flutter.dev/docs
[3] Anon., 2019. Moota API Reference. [Online] Available at: https://app.moota.co/developer/docs/
[4] Golub, B., 2013. dotCloud, Inc. is Becoming Docker, Inc. [Online] Available at: https://blog.docker.com/2013/10/dotcloud-is-becoming-docker-inc/

Gerry Surya Chandra menyelesaikan studi S1 di program studi Teknik Informatika STTS pada tahun 2020.

Suhatati Tjandra menyelesaikan studi S1 di program studi Teknik Informatika STTS pada tahun 1991 dan menyelesaikan studi masternya pada program studi Magister Teknologi Informasi STTS pada tahun 2005. Minat penelitiannya adalah bidang Sistem Informasi dan Software Engineering.