

Aplikasi Sistem Antar Jemput Berbasis Mobile

Yudi Hadi Saputra, Lily Puspa Dewi, Anita Nathania Purbowo
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp. (031) - 2983455, Fax (031) - 8417658
yudihadihadi@gmail.com, lily@petra.ac.id, anitaforpetra@gmail.com

ABSTRAK

Antar jemput sekolah dan universitas banyak digunakan, tetapi keberadaan antar jemput saat ini tidak bisa dilacak lokasinya. Dilihat dari aspek ini pengguna membutuhkan sebuah sistem yang dapat melihat lokasi kendaraan antar jemput dan sebuah sistem yang mampu menghubungkan antara pemilik, supir dan pengguna. Dilihat dari aspek supir, supir juga membutuhkan sistem yang mampu memberikan panduan navigasi mengenai rute yang bisa digunakan untuk sampai ke suatu tempat atau tujuan.

Aplikasi sistem antar jemput berbasis Android ini ditujukan agar membantu pemilik, supir dan pengguna dapat meningkatkan kualitas sistem antar jemput. Aplikasi ini akan dibuat dengan menggunakan bahasa Java dengan menggunakan Android Studio. Aplikasi berbasis *mobile* ini membantu owner untuk menentukan jadwal setiap minggunya dan setiap harinya, memantau lokasi setiap supir. Selain itu dapat membantu supir untuk mendapatkan navigasi mengenai rute untuk mencapai ke suatu lokasi atau tujuan.

Hasil pengujian menunjukkan bahwa aplikasi ini berjalan dengan baik pada *device* seperti Android 6.0.1 (Marshmallow). Aplikasi ini menjalankan fitur pengaturan jadwal mingguan dan harian, fitur notifikasi, menampilkan informasi supir, fitur obrolan, dan fitur navigasi sehingga memudahkan supir menggunakan rute terbaik.

KATA KUNCI: android, antar jemput, aplikasi antar jemput, *firebase cloud messaging*.

ABSTRACT

School and university transfers are widely used, but the location of the shuttle at this time cannot be tracked. Viewed from this aspect the user needs a system that can see the location of the shuttle vehicle and a system that is able to connect between the owner, driver and user. Viewed from the aspect of the driver, the driver also needs a system that is able to provide navigation guidance on routes that can be used to get to a place or destination.

This Android-based shuttle system application is intended to help owners, drivers and users to improve the quality of the shuttle system. This application will be made using the Java language using Android Studio. This mobile based application helps the owner to determine the schedule every week and every day, monitor the location of each driver. In addition, it can help the driver to get navigation about the route to reach a location or destination.

The test results show that this application runs well on devices such as Android 6.0.1 (Marshmallow). This application runs weekly and daily schedule settings, notification features, displays driver information, chat features, and navigation features making it easier for drivers to use the best routes.

Keywords: *android, firebase cloud messaging, shuttle, shuttle application.*

1. LATAR BELAKANG MASALAH

Usaha antar jemput anak sekolah memiliki prospek luar biasa dan akan terus berkembang. Pasalnya, selama orang masih membutuhkan pendidikan dan selama sekolah masih beroperasi, maka usaha macam ini begitu diminati [9]. Mobilitas sekarang merupakan hal yang penting, apalagi di daerah perkotaan yang padat. Salah satunya adalah adanya jasa antar jemput. Sistem antar jemput sekarang yang masih cenderung konvensional, yaitu dengan adanya sistem sms maupun telepon untuk penghubung antara penumpang dengan pemilik. Sedangkan pemilik sendiri perlu menghubungi setiap supir yang tersedia untuk menjemput penumpang itu di alamatnya masing-masing. Sehingga seringkali terjadi miskomunikasi antar ketiga pihak, yaitu pemilik, supir dan penumpang jasa akibat jadwal penjemputan yang cukup berubah – ubah.

Sistem antar jemput yang masih konvensional pastinya menggunakan sistem antar jemput yang lama sehingga kemungkinan terjadinya miskomunikasi sangat tinggi karena setiap komunikasi tersebut terhubung dengan telepon dan sms yang kadang susah dipahami. Kemungkinan terjadinya ketinggalan juga cukup besar karena jadwal penjemputan yang cukup berubah-ubah ditambah lagi jika penumpang yang memilih jasa antar jemput tersebut sangat banyak.

Ada beberapa jasa layanan antar jemput juga sudah mulai berhenti akibat persaingan dari taksi online yang lebih mudah digunakan daripada antar jemput konvensional yang menggunakan telepon dan sms. Ada juga yang berhenti karena banyaknya penumpang yang ada sedangkan pemilik yang tidak mumpuni untuk mengatur jadwal sebanyak itu sehingga jadwal menjadi kacau. Tidak hanya itu, terdapat banyak masalah yang dihadapi baik pemilik, pengguna maupun supir. Misalnya pada pemilik, susahnya mencari lokasi pasti para supir, pencatatan murid yang ikut dan yang tidak ikut. Pada pengguna, susahnya berkomunikasi dengan pemilik maupun supir yang bisa menyebabkan kesalahan-kesalahan sehingga terjadi salahnya penjemputan atau lainnya. Pada supir, mengetahui pastinya murid yang ikut ataupun tidak sehingga penjemputan dan pengantaran sedikit terhambat.

Oleh karena itu, kami ingin membuat aplikasi yang akan membantu efisiensi antar jemput, membantu pengguna antar jemput untuk merasa lebih nyaman, dan mengurangi miskomunikasi antara pengguna dan pemilik antar jemput tersebut.

2. LANDASAN TEORI

2.1. Layanan Antar Jemput

Berdasarkan informasi dari beberapa antar jemput yang ada di Petra, sistem antar jemput yang digunakan di Petra cukup maju, hampir semuanya memiliki sistem yang sama, yaitu penggunaan shift pada penjemputan dan juga pengantaran. Dibalik sistem yang cukup maju tersebut, terdapat beberapa kekurangan yang bukan

terletak pada sistem shift tetapi terletak pada komunikasi antar ketiga pihak, yaitu pengguna, supir dan pemilik. Sering terjadi miscommunication antar ketiga pihak tersebut yang menyebabkan kurangnya keefektifitas sistem tersebut yang mampu menyebabkan tidak dijemput atau terlambat.

2.2. Google Maps Platform

Google Maps Platform adalah satu set API dan SDK yang memungkinkan pengembang untuk menyematkan Google Maps ke aplikasi seluler dan halaman web, atau untuk mengambil data dari Google Maps [3].

Google Maps Platform menyediakan API yang dapat digunakan untuk menjalankan fitur yang berhubungan dengan Google Maps, sesuai dengan kebutuhan aplikasi. API yang disediakan Google Maps Platform antara lain, Directions API, Distance Matrix API dan Maps SDK for Android.

Untuk menggunakan API pada Google Maps Platform, kita harus mengaktifkan API yang akan digunakan pada Google Maps Platform. Kemudian, gunakan API key yang disediakan masing – masing API, dan masukkan API key tersebut pada aplikasi.

2.3. Directions API

Directions API adalah layanan yang menghitung arah antara lokasi menggunakan permintaan HTTP [1]. Dalam menggunakan Directions API dibutuhkan beberapa parameter yang harus ada misalnya, origin dan destination dalam bentuk alamat ataupun koordinat dan membutuhkan kunci API aplikasi. Terdapat parameter opsional lainnya seperti, mode untuk bagaimana kita akan melakukan perjalanan misalnya walking, driving dan bicycling, waypoint untuk mengubah rute melalui lokasi yang ditentukan, dan masih banyak opsional lainnya yang bisa digunakan.

2.4. Distance Matrix API

Distance Matrix API adalah layanan yang menyediakan jarak dan waktu perjalanan untuk matriks asal dan tujuan. API mengembalikan informasi berdasarkan rute yang direkomendasikan antara titik awal dan akhir, sebagaimana dihitung oleh Google Maps API, dan terdiri dari baris yang berisi nilai durasi dan jarak untuk setiap pasangan [2].

Dalam penggunaan Distance Matrix API, dibutuhkan origin dan destination dan juga key untuk memperoleh data berupa jarak dan waktu yang dibutuhkan untuk sampai.

2.5. Firebase

Firebase Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis [4].

Untuk menggunakan Firebase pada Android Studio, kita harus menghubungkan Android Studio dengan Firebase. Terdapat 2 cara untuk menghubungkan Android Studio dengan Firebase yaitu, dihubungkan menggunakan tools yang disediakan Android Studio dan menghubungkan secara manual dengan cara, buat Firebase Project pada Firebase Console, masukkan id aplikasi yang akan digunakan pada Firebase, tambahkan file konfigurasi Firebase pada aplikasi, tambahkan Firebase SDK pada aplikasi.

2.6. Firebase Cloud Messaging

Firebase Cloud Messaging adalah salah satu fitur yang disediakan oleh Firebase. Firebase Cloud Messaging (FCM) adalah solusi perpesanan lintas platform yang memungkinkan Anda mengirimkan pesan secara andal tanpa biaya [5].

Untuk menggunakan Firebase Cloud Messaging, kita harus menggunakan Server Key yang disediakan oleh Firebase Cloud Messaging pada Project tersebut. Untuk melakukan pengiriman notifikasi, dibutuhkan token yang berisikan id perangkat.

2.7. NoSQL

NoSQL adalah pendekatan untuk desain basis data yang dapat mengakomodasi berbagai macam model data, termasuk nilai kunci, dokumen, kolom dan format grafik. NoSQL, yang merupakan singkatan dari "tidak hanya SQL," adalah sebuah alternatif untuk database relasional tradisional di mana data ditempatkan dalam tabel dan skema data dirancang dengan cermat sebelum database dibangun. Database NoSQL sangat berguna untuk bekerja dengan set besar data terdistribusi [7].

2.8. AsyncTask

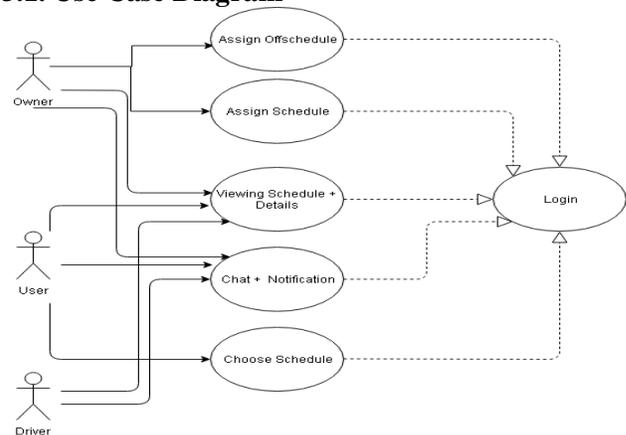
Gunakan kelas AsyncTask untuk mengimplementasikan tugas asinkron yang berjalan lama di Worker Thread/Thread Pekerja. Worker Thread adalah Thread yang bukan Thread UI/Main Thread. AsyncTask memungkinkan anda menjalankan operasi latar belakang dan mempublikasikan hasil di Thread UI tanpa memanipulasi thread [8].

AsyncTask memiliki beberapa langkah penting apabila dieksekusi:

- OnPreExecute(), dipanggil di Thread sebelum tugas dieksekusi.
- doInBackground(Params...), dipanggil pada Background Thread setelah OnPreExecute() selesai. Langkah ini dapat menjalankan komputasi pada Background, mengembalikan hasil, dan meneruskan hasil ke OnPreExecute().
- onProgressUpdate(Progress...), berjalan pada Thread UI setelah publishProgress(Progress...) dipanggil. Menggunakan onProgressUpdate() untuk memberitahu suatu bentuk kemajuan kepada Thread UI sewaktu komputasi Background dieksekusi.
- onPostExecute(Result), berjalan di Thread UI setelah komputasi pada Background selesai.

3. DESAIN SISTEM

3.1. Use Case Diagram



Gambar 1. Use Case Diagram

Diagram digunakan untuk menjabarkan interaksi pengguna terhadap sistem dalam aplikasi Shutapp. Gambar 1 menunjukkan desain use case diagram pada aplikasi Shutapp ini. Aplikasi mempunyai beberapa fitur utama seperti mengatur jadwal, pemilihan jadwal, penambahana jadwal, chat, gps dan notifikasi.

3.2. Analisa Sistem Serupa

Sebagai referensi, aplikasi yang akan dibuat akan dibandingkan dengan aplikasi serupa yang sudah ada, yaitu aplikasi XYZ.

Aplikasi layanan antar jemput XYZ berbasis Android ini ditujukan agar membantu user dapat langsung memesan layanan antar jemput dengan lebih cepat. Aplikasi ini akan dibuat dengan menggunakan bahasa HTML, Javascript, PHP yang kemudian akan diubah menjadi aplikasi mobile dengan menggunakan Eclipse. Aplikasi berbasis mobile ini juga akan membantu user untuk dapat mengetahui sopir yang akan menjemput sehingga akan membantu memudahkan dalam proses penjemputan. Selain dapat membantu dalam layanan pemesanan antar jemput, aplikasi ini juga membantu perusahaan antar jemput agar lebih cepat dan akurat dalam mencatat dan memproses pemesanan antar jemput sehingga dapat membantu dalam proses penjemputan user. Proses analisis sistem yang lama dengan melakukan survey ke layanan antar jemput [6].

Tabel 3.1. Perbandingan Aplikasi dengan Aplikasi Serupa

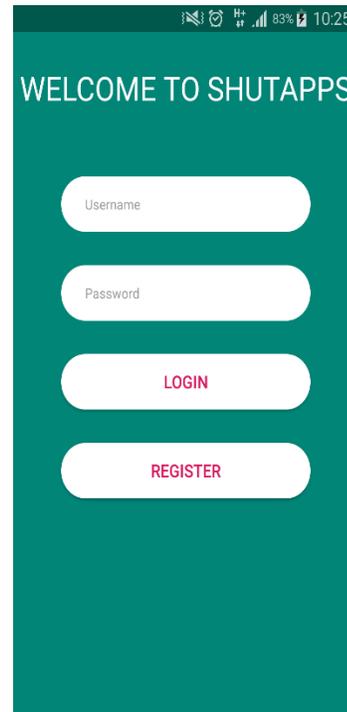
	Aplikasi XYZ	Aplikasi yang akan dibuat
Verifikasi melalui Email	x	✓
Melihat informasi penjemput	✓	✓
Dapat digunakan oleh jasa layanan antar jemput konvensional lainnya	x	✓
Pengaturan jadwal antar dan jemput setiap minggu dan setiap hari nya	x	✓
Navigasi ke lokasi pengguna / tujuan	x	✓
Jarak ke lokasi tujuan	✓	✓
Estimasi waktu	x	✓
Estimasi biaya	✓	x
Chat	x	✓
Notifikasi	✓	✓

4. PENGUJIAN SISTEM

Situasi pengujian adalah satu orang *owner* yang bernama Willy memiliki sebuah jasa antar jemput, memiliki seorang supir bernama Munir, dan dua orang user bernama Doni dan Lili. Willy memiliki

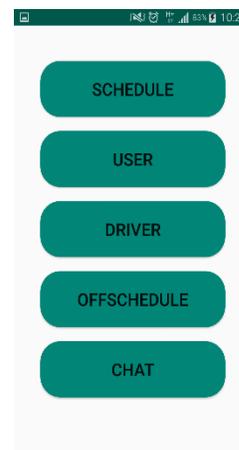
jadwal penjemputan dan pengantaran setiap harinya, tetapi pada tanggal 10 juni jadwal yang tersedia hanya penjemputan jam 6, 8 dan pengantaran jam 1 dan 3. Doni dan Lili memiliki jadwal yang di nonaktifkan sehingga harus memilih jadwal yang baru untuk penjemputan dan pengantaran.

Untuk dapat masuk ke halaman utama, pengguna harus memasukkan email dan password yang telah terdaftar pada form yang disediakan. Apabila ada form yang kosong, maka aplikasi akan meminta pengguna mengisi kembali form tersebut. Apabila Login berhasil, maka pengguna akan masuk ke dalam halaman utama. Pada Gambar 2 dapat dilihat Tampilan Login.

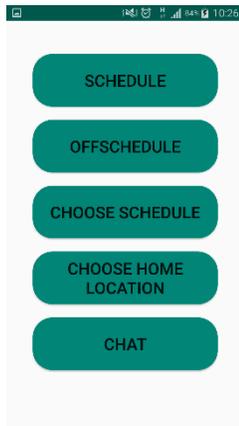


Gambar 2. Tampilan Login

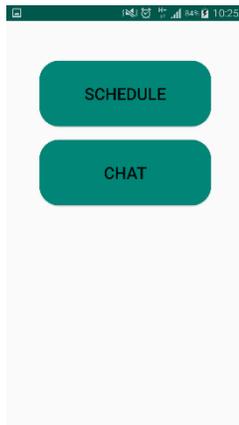
Sesuai dengan tipenya halaman utama terbagi menjadi tiga yaitu, Homepage Owner (Gambar 3), Homepage User (Gambar 4) dan Homepage Driver (Gambar 5).



Gambar 3. Tampilan Homepage Owner

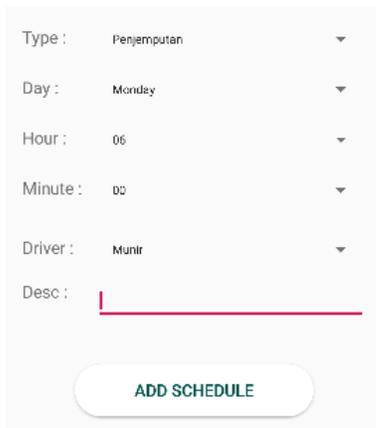


Gambar 4. Tampilan Homepage User



Gambar 5. Tampilan Homepage Driver

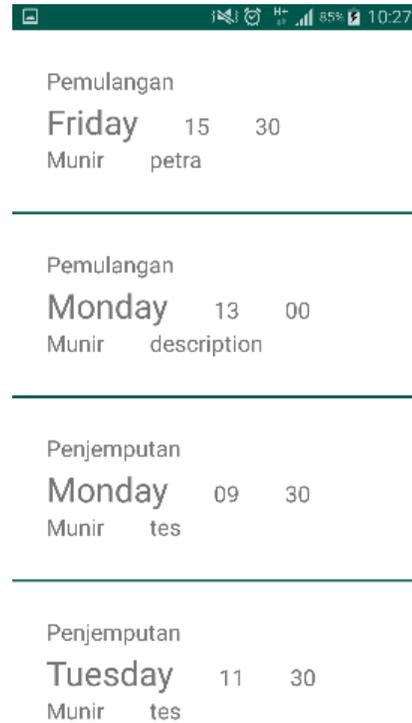
Owner dapat menambah jadwal yang ada dengan cara masuk ke Halaman Schedule, lalu menekan tombol tambah. Di Halaman Tambah Jadwal (Gambar 6.), owner memasukkan data dari jadwal yang akan dibuat. Apabila form tersebut sudah terisi dan menekan tombol tambah, maka jadwal akan tersimpan.



Gambar 6. Tampilan Halaman Tambah Jadwal

User dapat menambahkan jadwal yang ingin diikuti dengan cara masuk ke Halaman Choose Schedule (Gambar 7.), lalu sesuai

dengan cerita, jadwal yang tersedia pada tanggal 10 untuk penjemputan jam 6 dan 8 dan untuk pengantaran jam 13 dan jam 15. Oleh karena itu Doni dan Lili harus memilih salah satu dari kedua pilihan tersebut. Tampilan Halaman Add Schedule dapat dilihat pada Gambar 8.



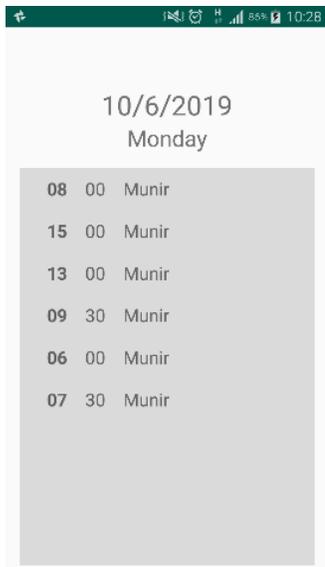
Gambar 7. Tampilan Halaman Choose Schedule



Gambar 8. Tampilan Halaman Add Schedule

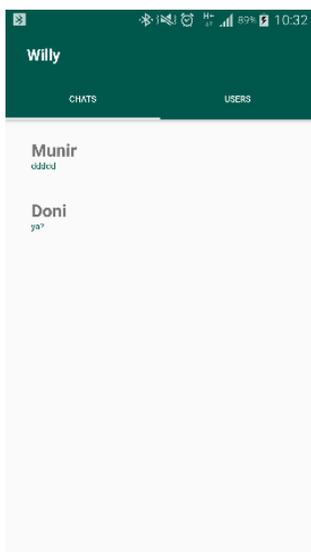
Owner dapat mengatur jadwal yang aktif pada tanggal tertentu dengan cara masuk ke Halaman Offschedule (Gambar 9.), lalu

sesuai dengan cerita, jadwal yang tersedia pada tanggal 10 untuk penjemputan jam 6 dan 8 dan untuk pengantaran jam 13 dan 15. Oleh karena itu selain jadwal yang tersebut akan di nonaktifkan untuk hari tersebut. Tampilan Halaman Offschedule pada tanggal 10 dapat dilihat pada Gambar 9. Nantinya pada Halaman Schedulesupir, jadwal yang dinonaktifkan pada hari tersebut akan tidak ditampilkan.

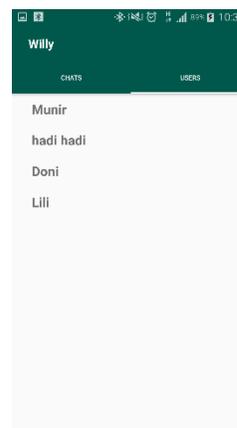


Gambar 9. Tampilan Halaman Add OffSchedule

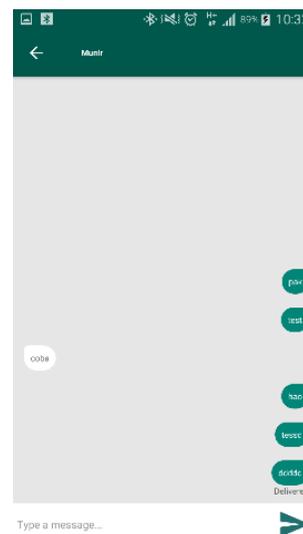
Untuk melakukan Chat, pengguna harus masuk ke dalam Halaman Chat yang terbagi menjadi 2 yaitu, Halaman Chat Chats (Gambar 10.) dan Halaman Chat Users (Gambar 11.). Untuk bisa melakukan Chat, pengguna harus memilih siapa yang akan di chat. Apabila sudah maka aplikasi akan melanjutkan ke Halaman Message Chat (Gambar 12.). Dan ketika mengirim sebuah pesan, akan mengirim notifikasi kepada orang yang dituju. Tampilan Notifikasi dapat dilihat pada Gambar 13.



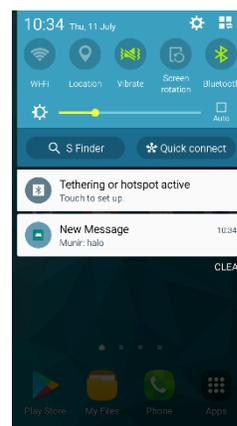
Gambar 10. Halaman Chat Chats



Gambar 11. Halaman Chat Users



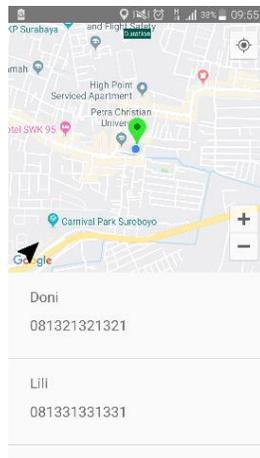
Gambar 12. Tampilan Halaman Message Chat



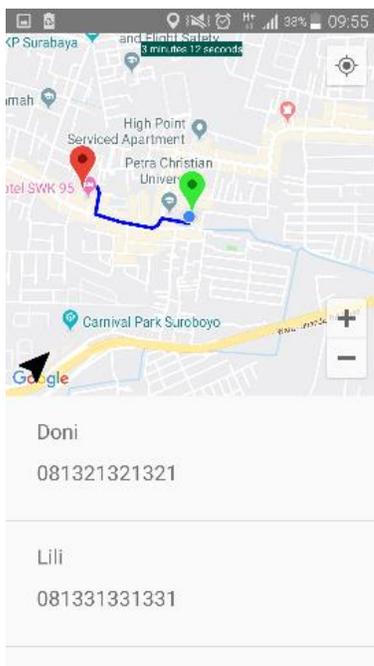
Gambar 13. Tampilan Notifikasi

Fitur Navigasi terbagi menjadi 3 yaitu, Navigasi untuk supir dan Navigasi untuk user dan owner. Untuk bisa menggunakan Navigasi, baik supir atau user dan owner harus masuk ke Halaman

Schedule dan memilih salah satu dari jadwal tersebut. Pada Navigasi untuk owner, owner dapat melihat lokasi supir, lalu bisa melihat siapa saja yang mengikuti jadwal tersebut, mampu melihat navigasi arah dari lokasi supir ke user yang akan dijemput beserta waktu yang akan ditempuh dan juga dapat melakukan telepon terhadap user yang mengikuti jadwal tersebut. Tampilan Halaman Navigasi Owner dapat dilihat pada Gambar 14. dan Gambar 15.

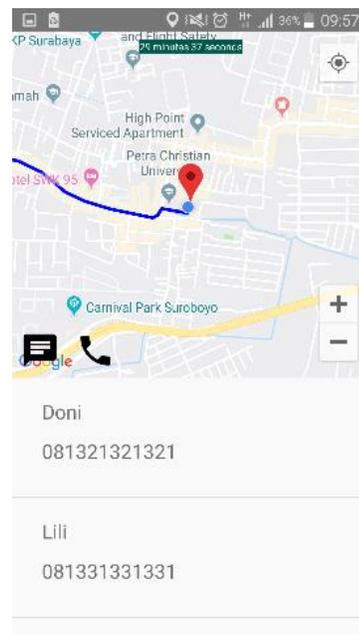


Gambar 14. Tampilan Halaman Navigasi Owner



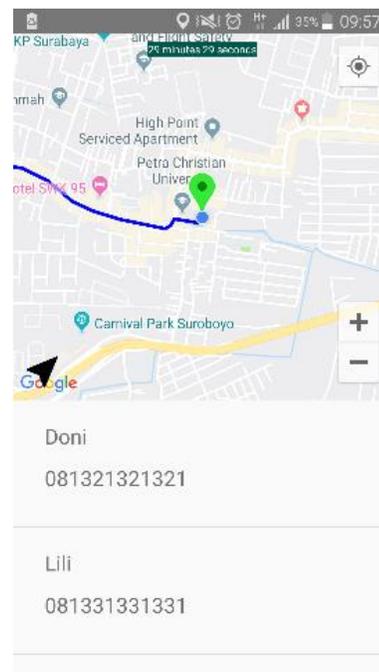
Gambar 15. Tampilan Halaman Navigasi Owner Setelah klik

Pada Navigasi untuk user, user dapat melihat lokasi supir dan waktu penjemputan yang akan ditempuh beserta navigasinya. Pada Tampilan Navigasi User terdapat tombol *chat* dan *call* yang ditujukan terhadap supir pada jadwal tersebut. Tampilan Halaman Navigasi User dapat dilihat pada Gambar 16.

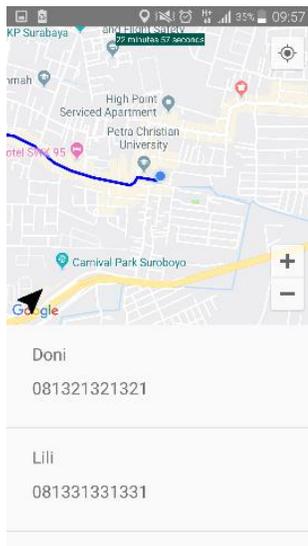


Gambar 16. Tampilan Halaman Navigasi User

Pada Navigasi untuk supir, supir dapat melihat lokasi dan mendapatkan arah untuk menjemput atau mengantarkan user yang ada pada jadwal tersebut. Apabila user yang akan dijemput menghasilkan waktu kurang dari 5 menit, aplikasi akan memberi user yang akan dijemput sebuah notifikasi. Supir juga memperoleh navigasi arah ke tujuan. Tampilan Halaman Navigasi Driver dapat dilihat pada Gambar 17 dan Gambar 18.



Gambar 17. Tampilan Halaman Navigasi Driver



Gambar 18. Tampilan Halaman Navigasi Driver ketika Destinasi dipilih

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari seluruh proses pembuatan dan proses pengujian program, dapat diambil beberapa kesimpulan, antara lain :

- Pengujian mampu menampilkan lokasi, navigasi dan durasi untuk melakukan penjemputan atau pengantaran menggunakan *Direction API* dan *Matrix Distance API* serta penggunaan *Firebase Cloud Messaging* untuk mengirim notifikasi.
- Data pada tabel user mengenai lokasi penjemputan harus diatur terlebih dahulu agar tidak terjadi kesalahan.
- Data pada tabel jadwal mengenai lokasi tujuan harus diatur terlebih dahulu agar tidak terjadi kesalahan.
- Program atau aplikasi yang dibuat mampu menjawab rumusan masalah yang ada
- Untuk tampilan aplikasi sistem antar jemput cukup baik.
- Berdasarkan kuisisioner yang dibagikan, didapatkan bahwa aplikasi cukup mudah untuk digunakan

- Aplikasi sangat bermanfaat dalam berlangsungnya proses sistem antar jemput

5.2. Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan aplikasi ini lebih lanjut antara lain :

- Tampilan diperbagus agar memudahkan pengguna.
- Pembuatan panduan penggunaan aplikasi dan penjelasan akan kosakata yang tidak umum, sehingga pengguna lebih memahami fitur-fitur yang ada.
- Penambahan status pada setiap user untuk setiap penjemputan atau pengantaran sehingga menjelaskan status penjemputan user tersebut.
- Perbaikan pada update location
- Tampilan kurang bagus sewaktu di *rotate*
- Tampilan kurang bagus pada *smartphone* yang berukuran kecil

6. DAFTAR PUSTAKA

- [1] *Directions API*. 2019, April 26. Retrieved from Google Maps Platform: <https://developers.google.com/maps/documentation/directions/intro>
- [2] *Distance Matrix API*. 2019, June 20. Retrieved from Google Maps Platform: <https://developers.google.com/maps/documentation/distance-matrix/intro>
- [3] *FAQ*. 2019, July 3. Retrieved from Google Maps Platform: <https://developers.google.com/maps/faq>
- [4] *Firebase | Firebase*. (n.d.). Retrieved November 29, 2018, from <https://firebase.google.com/products/>
- [5] *Firebase Cloud Messaging*. (n.d.). Retrieved from Firebase: <https://firebase.google.com/docs/cloud-messaging?authuser=0>
- [6] Gosal, R. 2016. *Aplikasi Layanan Antarjemput XYZ Berbasis Android*. Retrieved November 29, 2018, from <https://dewey.petra.ac.id/catalog/digital/detail?id=36756>
- [7] *NoSQL (Not Only SQL Database)*. 2017, March. Retrieved from TechTarget: <https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>
- [8] Ridwansyah, M. I. 2018, January 26. *Konsep AsyncTask dan AsyncTaskLoader pada Android - Easyread*. Retrieved from <https://medium.com/easyread/konsep-async-task-dan-async-task-loader-pada-android-b5ba2744dcd8>
- [9] Syaikhon, A. 2013, September 7. *Menjemput Rupiah dari Bisnis Antar Jemput Sekolah*. Retrieved November 29, 2018, from <http://www.neraca.co.id/article/32560/menjemput-rupiah-dari-bisnis-antar-jemput-sekolah>