

Pembuatan Sistem Pelayanan Taksi dengan Menggunakan *Android, Google Maps, dan Ruby on Rails*

Andi Wahyu Rahardjo Emanuel, Anthony Salim

Program Studi S1 Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri no. 65, Bandung

email: awreman@gmail.com, tenebralis90@yahoo.com

Abstract

With the increase of the number of population and economy especially in big cities, the need for any modes of public transportation is also increases. One of the mode of transportation with increasing demand by the medium-to-high income population is taxi. The fact that the current operation of taxies is still using conventional and inefficient ways of communication such as two-way radios, and phone calls, raises the possibility of using more efficient tools such as using Android devices, Google Map API, and Ruby of Rails framework as proposed in this paper. It can be shown that the application of these technologies can provide an integrated taxi service system which is more informative, effective and efficient.

Keywords: Android, Google Maps, Ruby on Rails, GPS, taxi service

1. Pendahuluan

Saat ini transportasi telah berkembang menjadi salah satu kebutuhan yang semakin berperan penting di segala lapisan masyarakat, khususnya di kota – kota besar. Berbagai jenis moda transportasi telah tumbuh dan berkembang seiring dengan tingkat perkembangan ekonomi dan penduduk di suatu daerah. Salah satu bentuk moda transportasi yang berkembang dengan pesat seiring dengan perkembangan penduduk kalangan menengah dan kalangan atas adalah taksi.

Meskipun taksi sebagai moda transportasi publik sudah berkembang dengan pesat, namun pada kenyataannya proses pengoperasiannya masih sangat konvensional dan minim sentuhan teknologi informasi. Proses pemesanan taksi dari calon konsumen masih mengandalkan panggilan telepon, dan juga proses distribusi pemesanan taksi dari pusat layanan ke masing – masing armada taksi juga masih menggunakan sarana komunikasi radio. Proses pelaporan dari armada taksi tentang proses pengantaran konsumen ke tujuan juga masih berupa komunikasi radio. Sistem pelayanan diatas tentu saja kurang efisien sehingga

dipandang perlu untuk merumuskan sebuah sistem baru yang dapat membuat sistem pemesanan taksi lebih efisien.

Pada tulisan ini dikemukakan salah satu solusi peningkatan efisiensi pelayanan taksi dengan memanfaatkan Teknologi Informasi dalam pengoperasian armada taksi, khususnya dengan menggunakan teknologi Internet dan perangkat bergerak berbasis Android. Aplikasi ini memungkinkan pelayanan pesanan oleh calon konsumen dengan menggunakan penjelajah Internet maupun dengan panggilan telepon, operator taksi juga bisa memantau armada taksi yang sedang beroperasi, sedangkan sopir taksi akan dibekali perangkat bergerak Android untuk menanggapi permintaan layanan dari operator atau pelaporan penyelesaian hantaran konsumen.

2. Kajian Teori

Perangkat – perangkat teknologi yang dipergunakan dalam sistem ini meliputi perangkat bergerak yang memiliki sistem operasi Android, layanan peta Internet *Google Maps*, teknologi pelacak lokasi GPS (*Global Positioning System*), framework aplikasi web *Ruby on Rails*, dan basis data *PostgreSQL*.

2.1. Android

Pada sistem pelayanan taksi ini, para sopir taksi akan dibekali sebuah perangkat mobile yang berbasis sistem operasi *Android*. Di perangkat tersebut akan dipasang suatu aplikasi khusus yang memungkinkan seorang sopir taksi untuk menanggapi permintaan untuk pelayanan konsumen oleh operator taksi. *Android* memiliki empat elemen yang menjadi fondasi sebuah aplikasi, yaitu *Activity*, *Intent*, *Service* dan *Content Provider*. Tidak semua dari elemen selalu dibutuhkan dalam membangun sebuah aplikasi, namun sistem ini umumnya merupakan kombinasi dari keempat hal tersebut. Setelah menentukan komponen apa saja yang akan dibutuhkan untuk aplikasi, pengembang perlu mencatat mereka pada sebuah berkas bernama *AndroidManifest.xml*. Berkas ini adalah berkas yang memiliki format XML (*Extensible Markup Language*) di mana pengembang dapat mendeklarasikan komponen yang akan digunakan pada aplikasi dan apa saja kemampuan yang dapat mereka kembangkan dan juga menyatakan *requirement* dari kemampuan tersebut [2].

2.2. Google Maps

Pada saat pendistribusian permintaan layanan dari konsumen yang diterima oleh operator, seorang operator akan menawarkan layanan ini pada taksi – taksi yang letaknya dekat dengan konsumen tersebut. Pengukuran jarak terdekat antara supir taksi dan pelanggan menggunakan salah satu fitur *Google Maps* yang disebut dengan *Distance Matrix*. *Distance Matrix* adalah sebuah layanan

yang menyediakan perhitungan jarak dan waktu dari *array* (lintang dan bujur lokasi) tempat awal dan *array* tempat tujuan. Berdasarkan informasi yang didapat, diperoleh dari rute rekomendasi dari titik awal dan titik akhir, seperti yang sudah dihitung oleh *Google Maps API*, dan terdiri dari baris-baris yang berisi nilai durasi dan jarak dari tiap pasangan *array* [2].

2.3. Global Positioning System

Untuk mendapatkan lokasi dari armada taksi, perangkat bergerak *Android* yang dioperasikan oleh sopir taksi akan mengirimkan lokasi keberadaan taksi dengan memanfaatkan teknologi GPS (*Global Positioning System*). *Global Positioning System* adalah sistem navigasi berbasis satelit yang mengirim dan menerima sinyal radio. Sebuah penerima GPS mendapatkan sinyal tersebut dan menyampaikan informasi pada pengguna. Dengan menggunakan teknologi GPS, pengguna dapat mengetahui lokasi, kecepatan, dan waktu selama 24 jam, pada kondisi cuaca apapun dimanapun di dunia secara gratis. Untuk perangkat bergerak *Android*, lokasi pengguna dapat diperoleh melalui *callback*. Pengembang dapat mengindikasikan untuk mendapatkan perubahan dari *LocationManager* dengan memanggil *requestLocationUpdates()*, memberikan sebuah *LocationListener*. *LocationListener* harus mengimplementasikan beberapa metode *callback* yang *LocationManager* panggil ketika lokasi pengguna berubah atau ketika status layanan berubah [3].

2.4. Ruby on Rails

Operator dari perusahaan taksi akan dibantu oleh sebuah aplikasi yang dibangun dengan teknologi *Ruby on Rails*. *Rails* adalah sebuah *framework* aplikasi web untuk bahasa pemrograman *Ruby*. *Rails* dapat menolong pengembang untuk membangun website secara cepat, dengan kode yang bersih dan mudah di maintain. Pengembang dapat menulis lebih sedikit kode untuk mencapai hasil yang sama dibandingkan dengan bahasa pemrograman lain. Filosofi *Rails* menyertakan beberapa prinsip berikut:

1. DRY – “*Don’t Repeat Yourself*” – menyarankan bahwa menulis kode yang sama berulang kali adalah hal yang buruk
2. *Convention Over Configuration* – berarti bahwa *Rails* membuat asumsi apa yang ingin pengembang ingin lakukan, daripada mengharuskan pengembang untuk melakukan spesifikasi terhadap setiap hal-hal kecil pada file configuration
3. REST (*Representational state transfer*) adalah pola terbaik untuk aplikasi web [1].

2.5. PostgreSQL

Penyimpanan data pada sistem pelayanan taksi ini menggunakan PostgreSQL. *PostgreSQL* adalah sistem manajemen *object-relational database* (ORDBMS) berbasis *POSTGRES*. *PostgreSQL* adalah turunan *open-source* dari kode *Berkeley* yang menyediakan sebagian besar standar SQL dan menawarkan beberapa fitur modern lainnya:

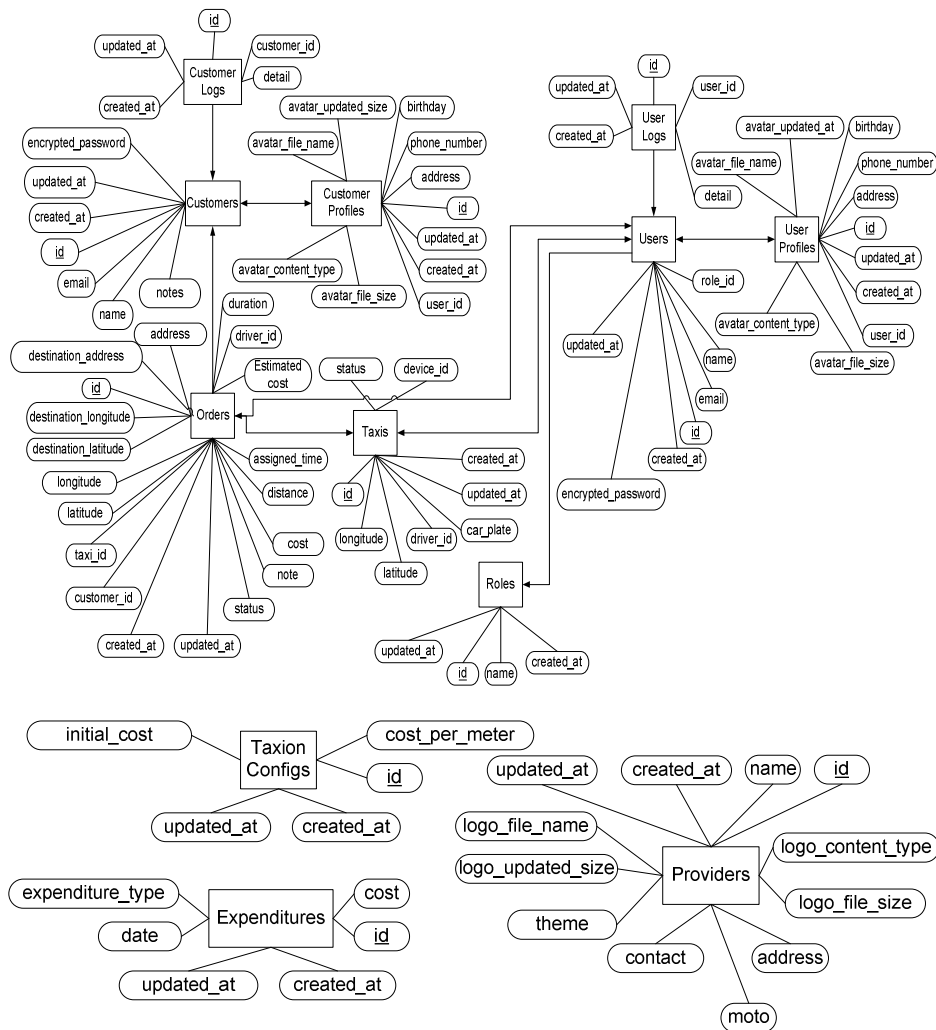
1. *Query* yang kompleks
2. *Foreign key*
3. *Trigger*
4. *View*
5. *Transactional Integrity*
6. *Multiversion concurrency control*.

3. Analisis dan Rancangan Sistem

Konsep perancangan dari sistem pelayanan taksi ini dengan menerapkan pola desain MVC (*Model – View – Controller*). Pada pola MVC, model merepresentasikan data, *view* merepresentasikan tampilan, dan *controller* adalah penyalur segala aksi. Dalam *Ruby on Rails*, kode harus ditempatkan di lokasi yang benar dan mengikuti konvensi penamaan agar *Rails* berjalan dengan normal. Setiap bagian dari pola MVC adalah entitas yang terpisah, yang dapat dibangun dan diuji coba secara terpisah. Perubahan pada *model*, tidak perlu mempengaruhi tampilan; sebaliknya, perubahan pada tampilan (*view*) tidak dapat mempengaruhi *model*. Ini artinya perubahan pada aplikasi MVC bersikap lokal dan hanya mempunyai *low impact*, sehingga memudahkan *maintenance* (perawatan) dan meningkatkan *reusability* (penggunaan kembali) secara signifikan antar komponen [1].

3.1. Rancangan Entity Relationship Diagram (ERD)

Rancangan ERD pada aplikasi pemesanan taksi dapat dilihat pada gambar 1.



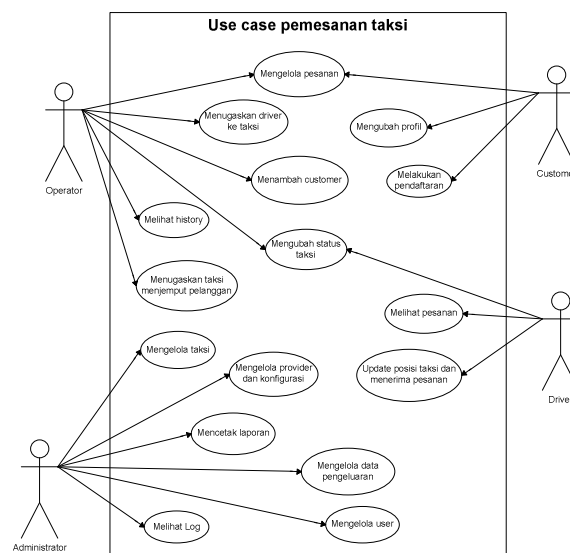
Gambar 1. Entity Relationship Diagram

Rancangan ERD terdiri dari entitas pengguna yang mencatat data akun supir, operator dan juga administrator aplikasi yang dibedakan melalui *primary key* ID pengguna. Pencatatan data profil pengguna dicatat pada entitas profil pengguna yang mempunyai *foreign key* pengguna. Setiap kegiatan operator dan administrator dicatat pada entitas log pengguna yang memiliki *foreign key* pengguna. Entitas taksi mencatat plat nomor taksi, koordinat taksi, dan juga kode sopir melalui *foreign key* pengguna. Pencatatan akun dan profil dilakukan pada entitas profil pelanggan dan entitas pelanggan. Setiap kegiatan pemesanan pelanggan akan dicatat pada entitas log pelanggan yang memiliki *foreign key*

pelanggan. Entitas pesanan mencatat seluruh pesanan yang dilakukan pelanggan. Pada entitas ini, data supir yang ditugaskan untuk menjemput pelanggan dicatat dalam bentuk ID supir. Entitas pengaturan mencatat pengaturan harga yang akan dikalkulasikan untuk estimasi ongkos pada pesanan pelanggan. Entitas *provider* digunakan untuk mencatat nama dan profil perusahaan yang menggunakan aplikasi pemesanan. Entitas pengeluaran digunakan untuk mencatat pengeluaran perusahaan untuk kepentingan laporan keuangan.

3.2. Rancangan Use Case Diagram

Rancangan *Use Case Diagram* pada aplikasi pemesanan taksi dapat dilihat pada gambar 2.

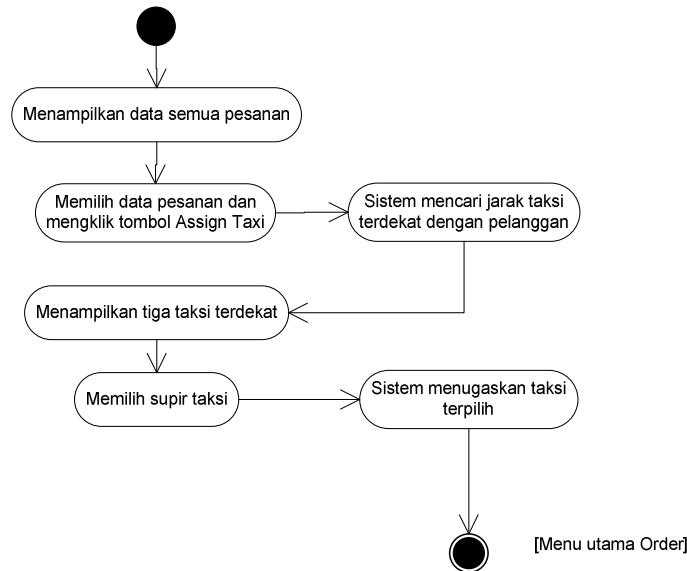


Gambar 2. Use Case Diagram

Secara umum, *use case diagram* yang digunakan adalah mengelola pesanan, menugaskan taksi untuk menjemput pelanggan, pendaftaran pelanggan, mengubah posisi taksi, dan menerima pesanan. Aktor yang berperan adalah administrator, operator, sopir taksi, dan pelanggan.

3.3. Rancangan Activity Diagram

Activity diagram penugasan supir taksi menjemput pelanggan pada aplikasi pemesanan taksi dapat dilihat pada gambar 3.

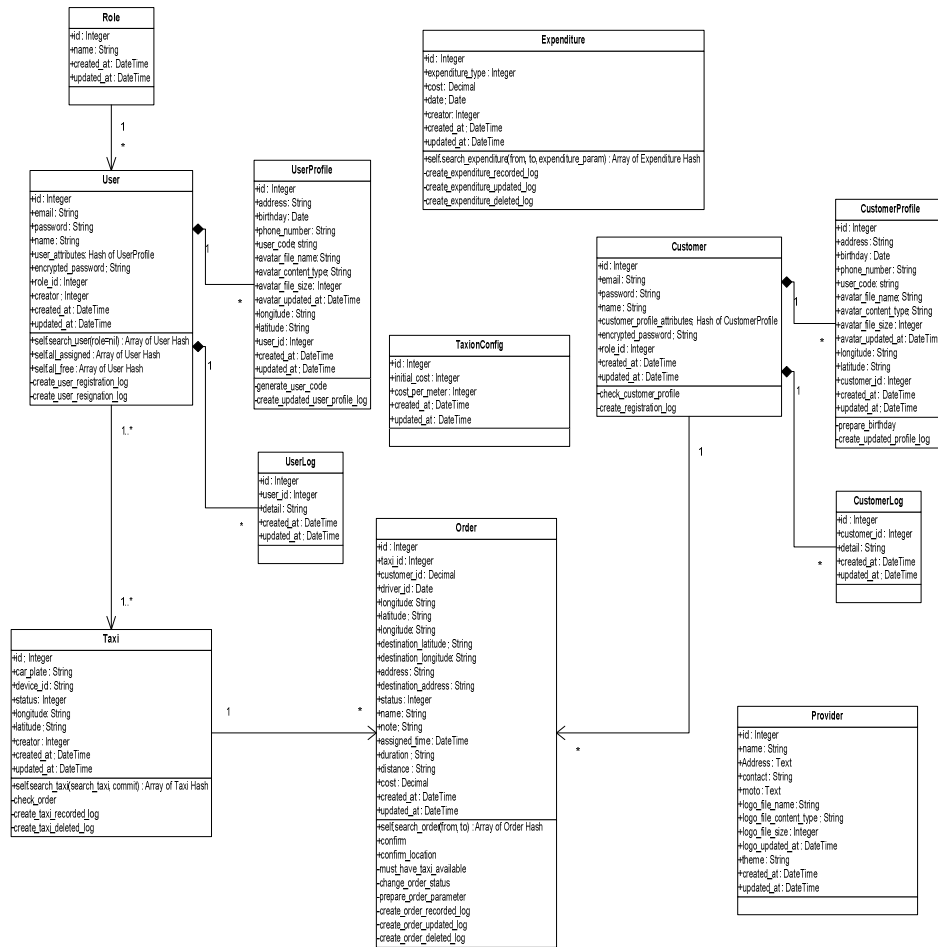


Gambar 3. Activity diagram penugasan supir taksi menjemput pelanggan

Activity diagram pada gambar 3 merupakan proses penugasan supir taksi menjemput pelanggan oleh operator. Pemilihan supir taksi yang akan menjemput pelanggan dipertimbangkan dengan faktor jarak terdekat dan frekuensi supir taksi telah melakukan pesanan pada periode tertentu. Pilihan supir taksi oleh operator akan dicatat pada database.

3.4. Perancangan Class Diagram

Class diagram pada aplikasi ini dapat dilihat pada gambar 4.

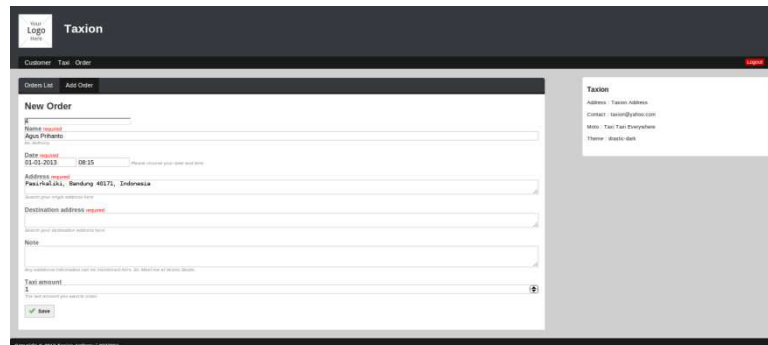


Gambar 4. Class Diagram

Class diagram ini merupakan model data yang dipetakan oleh Active Record berdasarkan entitas pada database.

4. Hasil Tercapai

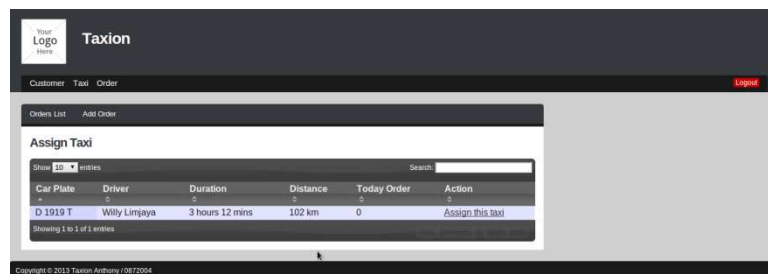
Aplikasi dapat digunakan oleh pelanggan apabila pelanggan sudah melakukan pendaftaran. Setelah pelanggan terdaftar dalam database, pelanggan dapat melakukan pemesanan melalui akun tersebut. Untuk pelanggan yang belum terdaftar pada database, dapat melakukan pemesanan dengan cara konvensional, yaitu menelepon operator taksi. Gambar 5 merupakan tampilan pengisian formulir pesanan.



Gambar 5. Halaman formulir pemesanan

Formulir pemesanan taksi digunakan untuk mengisi informasi mengenai pesanan berupa nama pelanggan, waktu pemesanan, tujuan asal dan tujuan akhir dari supir taksi. Selain itu, pelanggan juga dapat memesan lebih dari satu taksi dengan maksimal 5 unit taksi. *Field* catatan digunakan untuk melakukan catatan tambahan mengenai informasi tujuan awal.

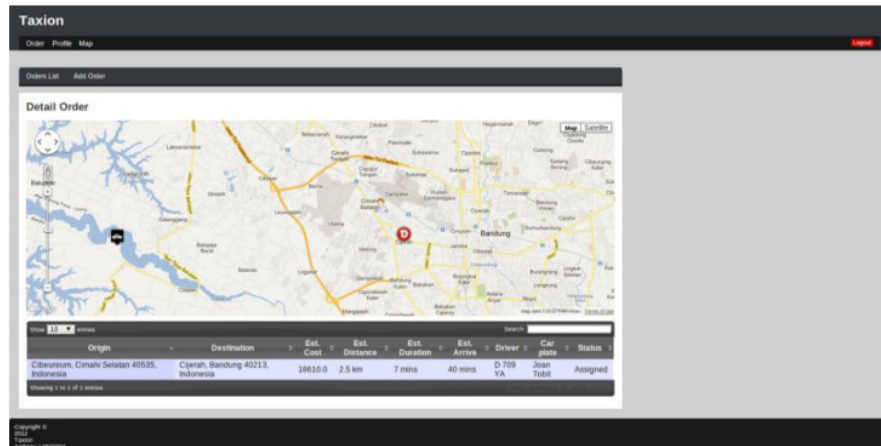
Setelah data pesanan masuk ke dalam database, operator dapat menugaskan supir taksi untuk menjemput pesanan tersebut. Supir taksi yang ditampilkan adalah tiga taksi yang terdekat dengan tujuan awal pesanan. Pemilihan taksi dipertimbangkan oleh jarak dan juga frekuensi pemesanan yang dilakukan supir taksi. Gambar 6 merupakan tampilan dari halaman pemilihan supir taksi.



Gambar 6. Halaman pemilihan supir taksi

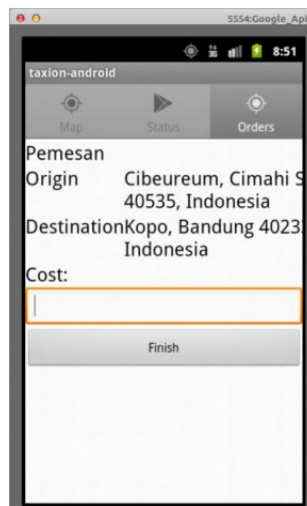
Operator juga bisa melihat letak dari seluruh armada taksi yang dimiliki dan juga juga lokasi asal dan tujuan dari suatu pesanan layanan taksi. Tampilan yang muncul di browser yang dilihat operator merupakan tampilan dari Google Maps yang dikombinasikan dengan ikon – ikon taksi, tujuan, asal, dll yang membantu seorang operator dalam mengatur pelayanan seluruh armada taksi.

Gambar 7 memperlihatkan tampilan sebuah taksi yang sedang melayani penumpang lengkap dengan tujuan dari perjalanan taksi tersebut.



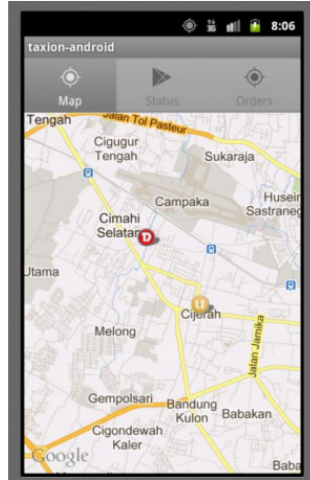
Gambar 7. Tampilan Detil Pesanan yang Dilihat Operator

Setelah supir taksi ditugaskan oleh operator, akan muncul pesan pesan baru pada perangkat bergerak *Android* yang dimiliki supir taksi yang ditugaskan tersebut. Dan apabila layanan pengantaran sudah dilaksanakan maka seorang sopir taksi juga diwajibkan memasukkan besaran biaya yang dibebankan ke penumpang. Gambar 8 Menunjukkan tampilan detil pesanan di perangkat *Android* yang dioperasikan oleh sopir taksi.



Gambar 8. Tampilan Detil Pesanan oleh Sopir

Sopir juga bisa melihat rute perjalanan yang akan ditempuh dengan bantuan *Google Maps* di perangkat Android-nya seperti yang diperlihatkan di gambar 9.



Gambar 9. Tampilan Rute Perjalanan

Keseluruhan data – data penumpang, biaya perjalanan yang dibayarkan penumpang, data aktifitas sopir taksi dan data – data lainnya juga direkam untuk keperluan administrasi dan dapat ditampilkan dalam bentuk berbagai macam format laporan yang bertujuan untuk memudahkan dalam evaluasi kinerja dari perusahaan taksi tersebut.

5. Simpulan dan Saran

5.1.Simpulan

Pada paper ini telah dipaparkan pembuatan sistem pelayanan taksi dengan menggunakan berbagai teknologi seperti *Android*, *Google Maps*, *Ruby on Rails*, dan *PostgreSQL* yang ditujukan untuk mengefisienkan pelayanan taksi yang selama ini masih konvensional. Beberapa kesimpulan yang dapat diperoleh dari pengembangan sistem ini adalah:

1. Berdasarkan hasil perancangan dan pengujian pada sistem yang telah dibuat, dapat disimpulkan bahwa teknologi Internet seperti *Google Maps*, *Ruby on Rails* yang dikombinasikan dengan perangkat bergerak berbasis *Android* dapat dipergunakan untuk membuat suatu layanan pengoperasian armada taksi.

2. Berdasarkan uji pengguna yang berbasis survey, dapat disimpulkan bahwa sistem ini dapat meningkatkan efisiensi dari pelayanan pengoperasian armada taksi.

5.2.Saran

Berikut ini adalah saran – saran pengembangan yang diperoleh selama masa uji coba dari sistem pelayanan taksi ini:

1. Cakupan aplikasi dapat diimplementasikan di luar area Bandung dan sekitarnya.
2. Diperlukan penyesuaian tampilan dengan resolusi layar, khususnya untuk mengakomodasi resolusi dan ukuran layar perangkat bergerak *Android* yang beraneka ragam.
3. Penunjukan posisi pesanan dan tujuan pesanan dapat diinput langsung melalui *Google Maps*

Daftar Pustaka

- [1] Carneiro, A.B.(2010). *Appress Beginning Rails 3*: Apress
- [2] Svennerberg, G. (2010). *Beginning Google Maps API 3*: Apress
- [3] Garmin Ltd. (2008). *GPS Guide for Beginners*