

# Aplikasi *Reservasi* Tiket Bus pada *Handphone* Android menggunakan *Web Service* (Studi Kasus: PO. Rosalia Indah)

1) Fajar Fani Hartono, 2)Hendry, 3)Ramos Somya

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Jl. Diponegoro 52-60, Salatiga 50711, Indonesia

Email: <sup>1)</sup>[fajar8091@gmail.com](mailto:fajar8091@gmail.com), <sup>2)</sup>[hendry@staff.uksw.edu](mailto:hendry@staff.uksw.edu),  
<sup>3)</sup>[ramos.6005@gmail.com](mailto:ramos.6005@gmail.com)

## Abstract

*Presentation of ticket availability information is still a manual one problems faced by the travel services for passengers have come to the ticket agent to check availability. Therefore the need for a reservation system applications. Bus ticket reservation system used for passengers booking a trip without having to come to the agency. To facilitate passengers to access this system, the bus ticket reservation application built Android-based mobile. Android is a collection of software intended for mobile devices includes an operating system. This application runs on the Android platform and integrate with applications residing on the server. Server is built to take advantage of web-based .web service. Web Service provides a standardized communication among different software applications are different, and can run on various platforms and frameworks. Through the application of this reservation, the reservation transaction is expected to be done anywhere and anytime without being bound by time and place.*

**Keyword :** *Reservation, Android, web service*

## Abstrak

Penyajian informasi ketersediaan tiket yang masih manual menjadi salah satu permasalahan yang dialami oleh jasa travel karena calon penumpang harus datang ke agen untuk mengetahui ketersediaan tiket. Oleh karena itu perlu adanya suatu aplikasi sistem *reservasi*. Sistem *Reservasi* tiket bus digunakan calon penumpang untuk melakukan pemesanan tiket perjalanan tanpa harus datang ke agen. Untuk memudahkan calon penumpang dalam mengakses sistem ini, maka dibangun aplikasi *reservasi* tiket bus berbasis Android *mobile*. Android adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi. Aplikasi ini berjalan pada *platform* Android dan berintegrasi dengan aplikasi yang berada di *server*. *Server* dibangun berbasis *web* yang memanfaatkan *web service*. *Web Service* menyediakan standar komunikasi di antara berbagai aplikasi *software* yang berbeda-beda, dan dapat berjalan di berbagai *platform* maupun *framework*. Penelitian ini membahas tentang perancangan dan implementasi aplikasi sistem *reservasi* tiket bus berbasis Android *mobile*. Melalui aplikasi *reservasi* ini, diharapkan transaksi *reservasi* dapat dilakukan dimana saja dan kapan saja tanpa terikat dengan waktu dan tempat.

**Kata Kunci :** *Reservasi, Android, web service*

## 1. Pendahuluan

Perkembangan teknologi informasi yang sangat pesat memungkinkan terjadinya pemenuhan permintaan baik yang berupa informasi, jasa, atau barang secara lebih cepat dan cepat. Terlebih lagi perkembangan yang pesat di bidang *mobile device*, di mana kini perangkat *mobile* sudah di dukung oleh *fitur* internet. Kondisi ini yang akhirnya dimanfaatkan oleh perusahaan jasa travel untuk mengatasi permasalahan penyajian informasi khususnya dalam ketersediaan tiket dan proses *reservasi* yang masih manual. Keberadaan aplikasi *reservasi* pada perangkat *mobile* mampu mengatasi permasalahan yang terjadi di perusahaan jasa travel, karena dengan aplikasi tersebut konsumen dapat melakukan transaksi *reservasi* dimanapun dan kapan saja tanpa terikat dengan waktu.

Perkembangan teknologi tersebut yang akhirnya dimanfaatkan oleh perusahaan jasa travel PO. Rosalia Indah. PO. Rosalia Indah adalah sebuah perusahaan yang sedang berkembang pesat dan merupakan perusahaan pribumi yang diperhitungkan dalam percaturan bisnis jasa angkutan darat di Indonesia. Perusahaan yang selalu mengedepankan pelayanan, keamanan, kenyamanan serta kebersihan armadanya ini memiliki jalur trayek Jawa-Sumatra dengan dua sistem *reservasi* tiket yaitu sistem manual untuk pemesanan tiket yang dilakukan oleh konsumen melalui agen dan

pemesanan via SMS (*Short Message Service*) oleh pelanggan. Namun keberadaan dua alternatif tersebut belum mampu memenuhi tingginya permintaan tiket yang berkisar 1959 tiket per hari.[1]. Berdasarkan angka permintaan tiket yang tinggi, penyajian informasi khususnya dalam ketersediaan tiket sering kali menjadi salah satu permasalahan karena konsumen harus datang ke agen untuk mengetahui ketersediaan tiket. Untuk itu perlu dibuat sistem *reservasi* yang dapat memberikan informasi yang lebih akurat dalam pemesanan tiket dan jadwal keberangkatan berbasis *mobile*. Melalui teknologi tersebut, *user* dapat melakukan transaksi *reservasi* tiket kapan saja dan dimana saja.

Perkembangan teknologi *mobile* yang sedang berkembang saat ini adalah Android. Android merupakan sistem operasi bergerak (*mobile*) yang menggunakan versi modifikasi dari kernel Linux. Sistem ini memiliki berbagai keunggulan sebagai *software* berbasis kode komputer yang bisa didistribusikan secara terbuka (*open source*), sehingga *programmer* bisa membuat aplikasi baru di dalamnya. Hal tersebut menjadikan android cocok untuk mendukung aplikasi *reservasi* berbasis *mobile*.

Berdasarkan permasalahan tersebut, maka dibutuhkan sebuah layanan yang mampu mendukung sebuah aplikasi *client*. *Web service* adalah layanan yang dirasa mampu mengatasi permasalahan tersebut. *Web service* yang berbasis XML sangat memungkinkan untuk diimplementasikan sehingga menjanjikan banyak kemudahan dan perbaikan dalam mendukung integrasi berbagai platform *system* dan aplikasi, baik melalui *infrastruktur Intranet* maupun *Internet/Exsternet*[2]. Keberadaan *web service* yang mampu mengelola administrasi dari penjualan tiket bus ini diharapkan dapat memudahkan agen dalam melakukan pekerjaannya, karena para agen cukup menyediakan *device* dan sebuah *aplikasi client* yang dapat mengolah transaksi yang dikirim oleh *web service* itu. Oleh karena itu muncullah keinginan untuk membangun sebuah aplikasi pemesanan tiket di PO. Rosalia Indah berbasis android *mobile* yang memanfaatkan *web service* guna memberikan pelayanan penyajian informasi yang lebih baik.

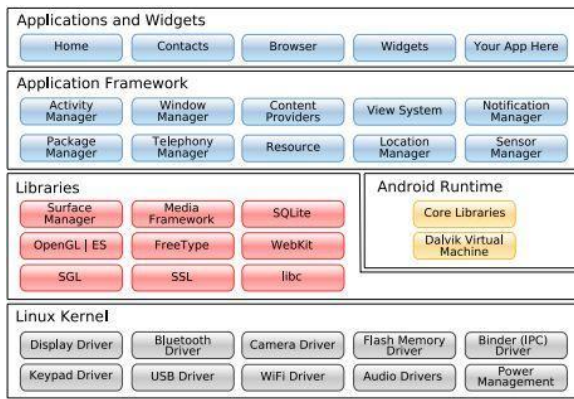
## 2. Tinjauan Pustaka

Penelitian sebelumnya adalah “perancangan aplikasi pemesanan tiket berbasis web” (study kasus: pemesanan tiket di PO. Maju Lancar) di mana dalam penelitian ini *user* dapat melakukan *reservasi* melalui sebuah aplikasi berbasis *web*. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman PHP dan MySQL dan dirancang dengan berbasis *web* yang memberi fasilitas pemesanan tiket bus sesuai dengan jadwal keberangkatan, dan pemberian alternatif pemilihan bus lain apabila tiket bus yang diinginkan sudah habis[3].

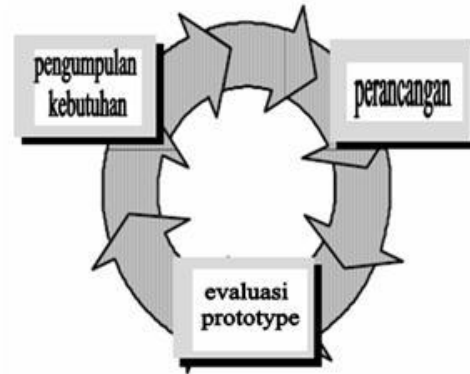
Perkembangan teknologi *smartphone* Android dimanfaatkan dalam pengembangan di sektor kuliner. Di mana teknologi ini digunakan dalam pembuatan aplikasi *delivery order* di restoran. Aplikasi tersebut dibangun untuk memudahkan para kostumer dalam memesan makanan dan melihat promo makanan yang ada di dalam restoran. Sehingga cukup dengan sentuhan pada menu yang terdapat dalam aplikasi kostumer dapat memesan makanan yang diinginkan[4].

Pada penelitian ini akan dibahas mengenai pembuatan sistem *reservasi* tiket bus akan diterapkan pada Android *mobile* yang akan menampilkan jadwal keberangkatan bus, ketersediaan tiket bus dan tempat duduk, serta konfirmasi pembayaran tiket. Aplikasi yang akan dibangun ini merupakan aplikasi berbasis *Android Mobile*, dimana aplikasi ini akan memanfaatkan *web service* yang akan menyediakan *method database*. Dimana *method-method* tersebut akan memudahkan dalam mengakses *database* sehingga pengolahan data lebih cepat.

**Android** adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. Android merupakan sistem operasi yang dikembangkan dengan memanfaatkan linux kernel untuk *smartphone* yang dibuat oleh Google corporation. Platform pada Android terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak[5]. Arsitektur Android ditunjukkan pada Gambar 1.



Gambar 1 Arsitektur Android[6]



Gambar 2 Metode Prototyping [9].

Arsitektur Android seperti yang ditunjukkan pada Gambar 1 terdiri dari 5 bagian utama, yaitu *Application* dan *Widgets*, *Application Framework*, *Libraries*, *Android Runtime*, dan *Linux Kernel*.

*Web Service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses dengan melalui jaringan, misalnya internet dalam bentuk pesan XML (*eXtensible Markup Language*)[7]. *Web Service* menyediakan standar komunikasi di antara berbagai aplikasi *software* yang berbeda-beda, dan dapat berjalan di berbagai *platform* maupun *framework*. *Web service* dapat ditemukan, diakses, diuraikan berdasarkan pada XML dan protocol standart Web pada intranet, externet, dan internet. Karena *web service* menggunakan format data yang universal yaitu XML, maka *web service* juga mewariskan sifat *multi-tier* dari XML. Sehingga memungkinkan terjadi integrasi antara *web service* atau aplikasi[8].

### 3. Metode Penelitian

Perancangan sistem ini menggunakan metode *prototyping*. *Prototyping* adalah proses yang digunakan untuk membantu pengembangan perangkat lunak dalam membentuk model dari perangkat lunak yang harus dibuat. Metode ini dilakukan secara bertahap, yaitu dengan mengembangkan suatu *prototype* yang sederhana terlebih dahulu baru kemudian dikembangkan dari waktu ke waktu sampai perangkat lunak selesai dikembangkan. *Prototype* merupakan bentuk dasar atau model awal dari suatu sistem atau subsistem [9].

Secara garis besar terdapat tiga tahapan dalam metode *prototyping* ini, seperti terlihat pada Gambar 2.

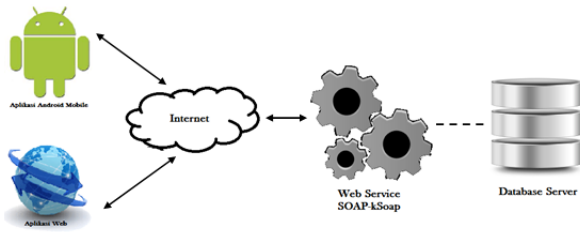
Metode *prototyping* seperti yang terlihat pada Gambar 2 terdiri dari tiga bagian utama yaitu:

- Pengumpulan kebutuhan

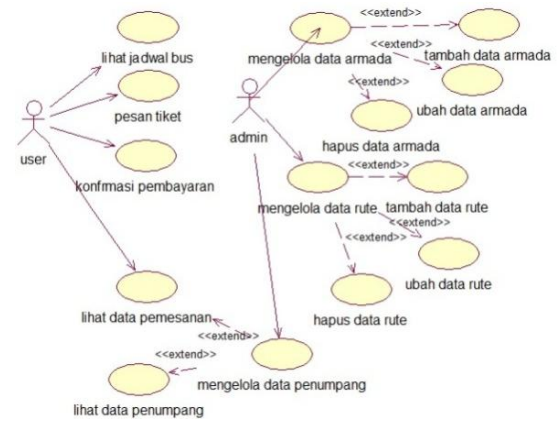
Tahap pertama yang dilakukan pada metode *prototyping* adalah pengumpulan kebutuhan. Pada tahap ini dilakukan dengan metode wawancara pada PO. Rosalia Indah. Dari wawancara yang telah dilakukan, dihasilkan data-data harga tiket, sebagai *sample* untuk aplikasi *reservasi* ini, akan memakai data harga tiket bulan Agustus.

- Perancangan Sistem

Langkah berikutnya dari metode ini adalah membuat rancangan sistem aplikasi, yaitu membangun sistem berdasarkan hasil dari tahap sebelumnya. Pada tahap ini perancangan sistem dilakukan dengan menggunakan UML (*Unified Model Language*) yang memanfaatkan *tools rational rose*, pada tahap ini pula dilakukan desain terhadap *User Interface*. Kemudian dilakukan perancangan tabel yang dibutuhkan dalam pembuatan aplikasi. Setelah semuanya selesai maka pada tahap ini aplikasi dibuat sampai selesai sesuai dengan *user requirement* yang diperoleh pada tahap awal. Arsitektur sistem dari sistem yang dibangun terlihat pada Gambar 3.



Gambar 3. Arsitektur Sistem



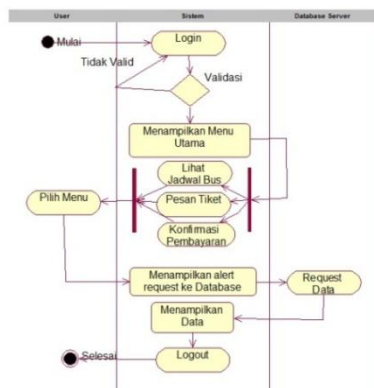
Gambar 4 Use Case Diagram

Gambar 3 merupakan rancangan arsitektur aplikasi reservasi tiket bus. Aplikasi mobile diimplementasikan ke perangkat Android mobile. Aplikasi web diimplementasikan pada laptop/PC. Semua perangkat harus terhubung dengan internet. Web service menghubungkan antara aplikasi mobile dan web. Library KSOAP digunakan untuk mengakses data pada database server.

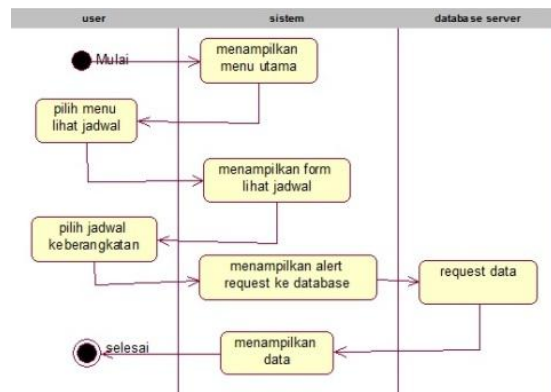
Use Case Diagram menggambarkan interaksi antara aktor-aktor dengan sistem yang dibangun[10], serta menggambarkan fungsionalitas yang dapat diberikan sistem kepada user. Dari dalam sistem yang akan dibangun terdapat dua aktor yang masing-masing memiliki hak akses yang berbeda-beda yaitu user, dan admin. User adalah masyarakat umum yang bisa mengakses sistem informasi dan memesan tiket bus di PO. Rosalia Indah Admin adalah pegawai di PO. Rosalia Indah yang khusus menangani ticketing.

Gambar 4 menjelaskan interaksi antara aktor dalam sistem serta fungsionalitas yang diberikan kepada user, dan admin. Mengenai bagian-bagian yang tersedia untuk hak akses user yaitu dapat melakukan registrasi sebagai pelanggan sehingga dapat melakukan reservasi tiket kepada pihak PO. Rosalia Indah. Selain itu admin dapat melakukan pengelolaan data armada yaitu menambah, edit dan delete ketersediaan armada, mengolah data rute bus yaitu tambah rute, edit rute, delete rute beserta jadwal keberangkatan, juga dapat mengolah data penumpang yaitu melihat data penumpang, mengubah data pesan tiket.

Activity diagram menggambarkan aliran aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir[10].



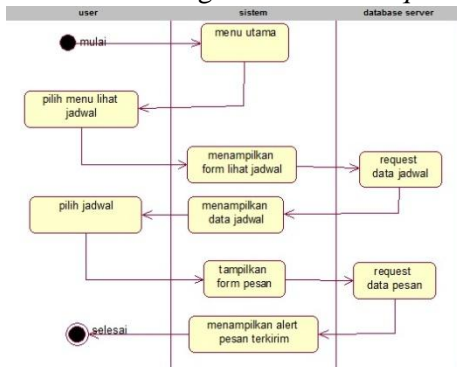
Gambar 5 Activity Diagram user



Gambar 6 Activity Diagram Lihat Jadwal

Gambar5 menggambarkan aktifitas yang dapat dilakukan oleh *user*. Pertama yang dilakukan oleh *user* adalah melakukan *login*, jika berhasil maka *user* dapat masuk ke *form* utama. Pada *form* utama terdapat beberapa *user*, diantaranya lihat jadwal bus, pesan tiket dan konfirmasi pembayaran. Setelah *user* memilih salah satu menu secara otomatis sistem akan mengirimkan *alert request* ke *database* dan *database* akan menampilkan data. Setelah itu *user* dapat *logout* dari sistem.

Gambar 6 menggambarkan aktifitas yang dapat dilakukan oleh *user* untuk melihat jadwal keberangkatan bus. Pertama yang dilakukan oleh *user* adalah memilih menu lihat jadwal, sistem akan menampilkan *form* lihat jadwal. Pada *form* lihat jadwal *user* harus memilih tanggal keberangkatan serta tujuan dari perjalanan. Setelah *user* memilih salah satu jadwal secara otomatis sistem akan mengirimkan *alert request* ke *database* dan *database* akan menampilkan data.



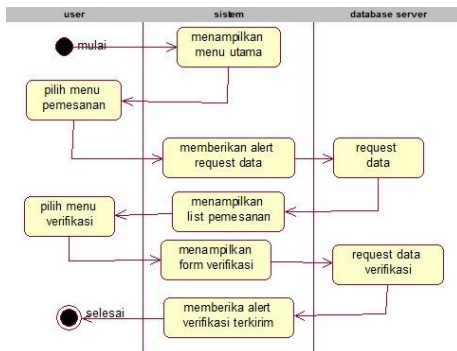
Gambar 7 Activity Diagram Pesan Tiket



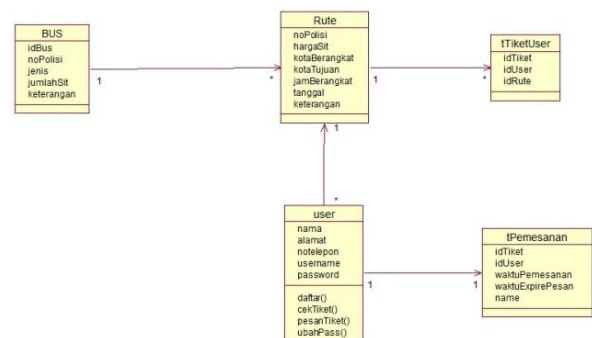
Gambar 8 Activity Diagram Lihat Pemesanan

Gambar 7 menggambarkan aktifitas yang dapat dilakukan oleh *user* untuk memesan tiket. Setelah sistem menampilkan data jadwal keberangkatan, *user* dapat memesan tiket sesuai jadwal keberangkatan. Setelah *user* memilih salah satu jadwal, agen keberangkatan, dan tempat duduk maka secara otomatis sistem akan mengirimkan *request* ke *database* dan sistem akan menampilkan *alert* bahwa pemesanan tiket telah berhasil.

Gambar 8 menggambarkan aktifitas lihat data pemesanan. Setelah *user* memesan tiket, *user* dapat melihat list data pemesanan tiket dengan memilih menu lihat pemesanan.



Gambar 9 Activity Diagram verifikasi Pembayaran



Gambar 10 Class Diagram Sistem

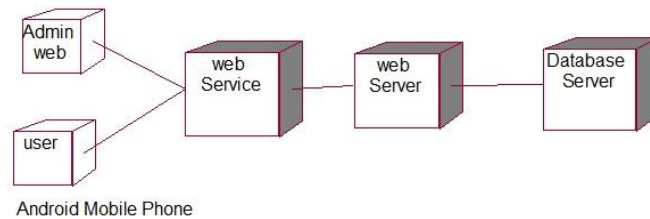
Gambar 9 menggambarkan aktifitas verifikasi pembayaran tiket bus. Pertama yang dilakukan oleh *user* adalah memilih menu lihat pemesanan. Pada list daftar pemesanan terdapat sub menu verifikasi, setelah *user* memilih menu verifikasi mana akan muncul *form* verifikasi. Setelah *user* mengisi *form* secara otomatis sistem akan mengirimkan *request* ke *database* dan sistem akan memberikan *alert* pengiriman verifikasi sukses.

**Class Diagram** adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. **Class** menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi

keadaan tersebut (metoda/fungsi) [11]. Gambar 10 merupakan diagram *class* dari sistem aplikasi ini.

Gambar 10 merupakan *class* diagram dari aplikasi yang dibuat. Terdiri dari beberapa *class* yaitu Bus, Rute, tTiketUser, *user*, dan tPemesanan. *ClassUser* merupakan *class* utama pada *class* diagram yang dirancang. Dalam *class* ini terdapat semua fungsi yang diperlukan untuk menjalankan aplikasi. *Class* yang lain hanya mengakses fungsi dari *class* utama tersebut. Sehingga setiap *class* yang ada pada *class* diagram tersebut memiliki hubungan dengan *classUser*.

**Deployment diagram** adalah susunan fisik sebuah sistem, menunjukkan tata letak bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*. Gambar 11 menunjukkan *Deployment diagram* untuk aplikasi yang akan dibuat.



**Gambar 11** *Deployment Diagram*

Gambar 11 merupakan *deployment diagram* dari sistem yang akan dibuat, terlihat bahwa ada beberapa perangkat yang digunakan saat *deployment*, yaitu sebuah komputer yang bertugas sebagai *server* yang di dalamnya terdapat *database* yang digunakan untuk menyimpan data-data *admin* serta *android device* untuk *client* yang bertindak sebagai pengguna aplikasi. *Client* akan terhubung dengan *web server* ketika melakukan akses data atau *input* data seperti melakukan registrasi, dan melakukan *reservasi* tiket.

#### – Evaluasi *prototyping*

Setelah aplikasi selesai dibuat, maka pada tahap ini dilakukan *testing* aplikasi terhadap *user*. *Testing* aplikasi ini bertujuan untuk mempermudah melakukan evaluasi terhadap kekurangan yang ada di dalam aplikasi sesuai dengan *user requirement* pada tahap awal. Ini merupakan tahap paling akhir yang ada di dalam *prototyping*, apakah tujuan umum dari pembuatan *software* ini telah tercapai. Apabila belum tercapai maka tahap selanjutnya yang harus dilakukan adalah kembali ke tahap paling awal, begitu seterusnya sampai tujuan umum dapat tercapai.

## 4. Hasil Dan Pembahasan

Hasil dan pembahasan meliputi pembuatan aplikasi *mobile* untuk reservasi tiket bus. Hasil yang dibahas adalah penerapan *web service* dengan *client* berupa aplikasi *web* dan aplikasi Android *mobile* yang dapat digunakan sesuai dengan analisis kebutuhan yang telah dilakukan. Terdapat dua aplikasi untuk aplikasi *reservasi* tiket bus ini, yaitu aplikasi *mobile* dan aplikasi *web*. Kedua aplikasi dihubungkan oleh *libraryksoap2-android*. Kode Program 1 adalah perintah untuk koneksi menggunakan *ksoap2-android*.

### Kode Program 1 Perintah Untuk Target Koneksi

```

1. private static final String NAMESPACE =
2. "http://service.me.com/";
3. private static String URL =
4. "http://10.0.2.2:8080/ServiceTiketing/ServiceTicketing?WSDL";
5. private static final String METHOD_NAME =
6. "cekLoginAnggotaMobile";
7. private static final String SOAP_ACTION =
8. "http://service.me.com/cekLoginAnggotaMobile";
  
```

Kode Program 1 merupakan perintah untuk menentukan target koneksi dari aplikasi *mobile* ke aplikasi *web*. Perintah pemanggilan koneksi dapat dilihat pada Kode Program 2.

**Kode Program 2** Perintah Untuk Pemanggilan Koneksi

```

1. SoapSerializationEnvelope envelope = new
2. SoapSerializationEnvelope (SoapEnvelope.V1);
3. envelope.setOutputSoapObject (request);
4. HttpTransportSE androidHttpTransport = new
5. HttpTransportSE (URL);
6. androidHttpTransport.call (SOAP_ACTION, envelope);
7. SoapPrimitive resultsRequestSOAP = (SoapPrimitive)
8. envelope.getResponse ();
9. String cek = resultsRequestSOAP.toString ();
    
```

Tampilan awal dari aplikasi *client* adalah *form login*, yang mengharuskan *user* memasukkan *username* dan *password* terlebih dahulu untuk dapat menggunakan aplikasi. Namun jika *user* belum mempunyai *username* dan *password* atau belum terdaftar, bisa melakukan registrasi seperti yang tampak pada Gambar 12.



Gambar 12 Form Registrasi User



Gambar 13 Halaman Login User



Gambar 14 Halaman Home

Gambar 12 merupakan *form* registrasi untuk *user* yang ingin mendaftar menjadi *member*. Informasi yang di-*input*-kan kemudian akan tersimpan ke dalam *database*. Halaman *login user* ditunjukkan seperti yang tampak pada Gambar 13.

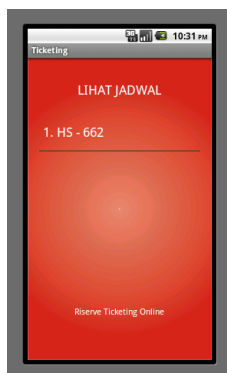
Berdasarkan Gambar 13 *user* diharuskan mengisi *username* dan *password*. *Username* dan *password* diisi berdasarkan registrasi *user* yang telah dilakukan sebelumnya. Jika data yang dimasukkan sesuai maka *user* bisa langsung masuk ke halaman utama. Halaman utama dari *user* ditunjukkan pada Gambar 14.

Gambar 14 merupakan halaman *home* dari aplikasi reservasi ini. *User* dapat keluar dari aplikasi dengan menekan tombol *Exit*. Jika tombol menu yang ada pada *device* android ditekan akan muncul lima menu yaitu lihat jadwal, pemesanan, *List* tiket, *account* dan *exit*. Tiga menu paling penting yang akan dibahas selanjutnya yaitu menu lihat jadwal, pemesanan, dan *list* tiket, karena menu ini berfungsi dalam pemesanan tiket.

Menu lihat jadwal jika dipilih maka tampilannya seperti Gambar 15.



Gambar 15 Menu Lihat Jadwal



Gambar 16 Halaman Data Jadwal

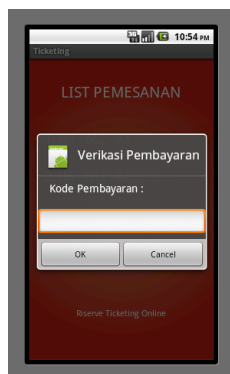


Gambar 17 Halaman Pemesanan Tiket

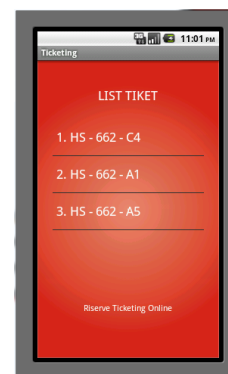
Pada Gambar 15 tersebut memperlihatkan halaman lihat jadwal bus pada aplikasi. *User* harus mengisi form pencarian jadwal keberangkatan bus sesuai dengan tanggal, asal, dan tujuan keberangkatan. Jika tombol cari telah diklik maka akan keluar data jadwal keberangkatan bus yang dipilih seperti pada Gambar 16.

Gambar 16 merupakan tampilan ketika *user* memencet tombol cari. Seperti contoh sebelumnya jika *user* memilih jadwal keberangkatan pada tanggal 30 Juli 2012 dari Semarang ke Solo, setelah tombol cari diklik maka keluar data jadwal keberangkatan dari pencarian tersebut. *User* dapat memesan tiket dengan mengklik pesan pada salah satu jadwal keberangkatan. Saat *user* klik salah satu data jadwal keberangkatan, *user* klik pesan maka *user* akan masuk ke menu pemesanan tiket seperti terlihat pada Gambar 17.

Gambar 17 merupakan halaman pemesanan tiket di mana *user* akan memilih agen keberangkatan dan tempat duduk yang diinginkan. *User* dapat memilih menu verifikasi untuk melakukan verifikasi pembayaran tiket seperti terlihat pada Gambar 18.



Gambar 18 Halaman Verifikasi Pembayaran



Gambar 19 Halaman List Tiket

Pada Gambar 18 merupakan tampilan halaman konfirmasi pembayaran. Pada halaman ini *user* harus mengisi form konfirmasi dengan mencantumkan kode transaksi pembayaran via bank. Setelah melakukan proses konfirmasi maka *user* akan dinyatakan *fix* memesan tiket. Daftar tiket yang sudah *fix* dapat dilihat pada menu *list* tiket seperti Gambar 19.

Gambar 19 merupakan tampilan dari halaman *list* tiket. Pada menu ini menampilkan daftar tiket yang sudah dipesan oleh *user* dan sudah melakukan verifikasi pembayaran.

### Antarmuka Aplikasi Web Admin

Tampilan awal dari aplikasi *web admin* adalah *form login*, yang mengharuskan *admin* memasukkan *username* dan *password* terlebih dahulu untuk dapat mengakses seluruh halaman *web*. Gambar 20 menunjukkan halaman *login* untuk *admin*.



Gambar 20 Halaman Login Admin



Gambar 21 Halaman Add Armada



Gambar 20 merupakan tampilan halaman *login* untuk *admin*. Jika data yang dimasukkan sesuai maka *admin* bisa langsung masuk ke halaman utama. *Admin* yang telah berhasil *login* kemudian masuk ke halaman *home*. *Admin* bisa *manage* semua data tentang aplikasi *reservasi* tiket bus P.O Rosalia Indah dengan menu armada dan rute bus. Pada menu armada terdapat 2 kategori data yang bisa diolah, yaitu data *add* armada, dan kelola armada. Pada menu rute bus, *admin* dapat menambah rute bus dan kelola pemesanan tiket. Pada data pemesanan tiket *admin* dapat melihat daftar user yang memesan tiket dan daftar penumpang tiap armadanya. Jika *admin* memilih menu *add* armada maka tampilannya akan seperti Gambar 21.

Gambar 21 merupakan halaman tambah data armada yang tersedia. Data yang dibutuhkan untuk menambah data armada adalah nama no polisi, jenis armada, jumlah sit, dan keterangan. *Admin* juga dapat mengelola data rute tiap armada yang tersedia. Gambar 22 menunjukkan tampilan untuk tambah data rute.



Gambar 22 Tambah Data Rute



Gambar 23 Halaman Kelola Rute Bus

Gambar 22 merupakan halaman tambah data rute armada yang tersedia. Data yang dibutuhkan untuk menambah data rute armada adalah no polisi atau jenis armada, harga/sit, tanggal, jam, asal keberangkatan dan tujuan. Kemudian untuk mengelola data pemesanan dan daftar penumpang, *admin* dapat memilih menu kelola rute bus seperti Gambar 23.

Gambar 23 merupakan halaman kelola data rute bus. Data yang ditampilkan pada menu ini adalah id\_rute, no polisi, jumlah sit, sisa sit, harga/sit, kota berangkat, kota tujuan, tanggal, jam, daftar pemesanan, daftar penumpang. Gambar 24 menunjukkan halaman daftar pesan.



Gambar 24 Halaman Daftar Pesan



Gambar 25 Halaman Daftar Penumpang

Gambar 24 menunjukkan daftar pesan yang merupakan data dari calon penumpang, untuk calon penumpang yang belum melakukan konfirmasi pembayaran tiket maka nilai verifikasi pembayaran adalah 0. Setelah user melakukan verifikasi pembayaran, *admin* dapat memilih *cek out* bayar maka data akan langsung masuk ke daftar penumpang seperti Gambar 25.

Gambar 25 merupakan halaman daftar penumpang. Pada halaman ini adalah data dari penumpang yang sudah melakukan konfirmasi pembayaran dan dinyatakan *fix* memesan tiket.

### Pengujian Sistem

Tahap pengujian sistem ini merupakan tahapan saat seorang pengembang aplikasi melakukan pengujian terhadap sistem yang sudah dirancang. Tahap ini merupakan tahapan untuk menemukan beberapa kekurangan yang ada dalam sistem. Pengujian aplikasi ini seperti yang dijelaskan pada tahap sebelumnya, menggunakan dua teknik pengujian, yaitu:

#### Pengujian Alfa

Pengujian alfa merupakan pengujian program yang dilakukan oleh pembuat aplikasi ataupun orang-orang yang terlibat di dalamnya. Pengujian alfa hanya untuk sirkulasi internal dan masalah (*error*) atau ketidaklengkapan yang terdapat dalam aplikasi dapat diduga sebelumnya [12]. Pada tahap ini pengujian dilakukan dengan metode *blackbox*. Untuk rincian pengujian dapat dilihat pada Table 1 berikut :

- a. Pengujian *Login* yaitu untuk mengecek apakah fungsional dari *formlogin* telah berjalan dengan baik. Dari pengujian yang dilakukan didapatkan kesimpulan bahwa *form login* telah berfungsi dengan baik.

**Tabel 1** Pengujian *Black Box* untuk *form login*

Aktivitas dan Event	Input	Output	Status Pengujian
<i>Login user</i> dengan klik tombol login	<i>Username</i> dan <i>password</i>	a. Jika berhasil maka masuk halaman <i>home</i> b. Jika <i>password</i> salah maka muncul peringatan <i>password</i> salah c. Jika <i>username</i> salah muncul peringatan <i>user</i> belum tersedia	Valid
<i>Login admin</i> dengan klik tombol login	<i>Username</i> dan <i>password</i>	a. Jika berhasil maka masuk halaman <i>admin</i> b. Jika <i>password</i> salah maka muncul peringatan <i>password</i> salah c. Jika <i>username</i> salah muncul peringatan <i>user</i> belum tersedia	Valid

- b. Pengujian untuk menu-menu yang dapat diakses *user* pada aplikasi *mobile* meliputi menu lihat jadwal, pemesanan, list pemesanan, dan verifikasi pembayaran. Dari pengujian yang dilakukan didapatkan kesimpulan bahwa semua menu telah berfungsi dengan baik.

**Tabel 2** Pengujian *Black Box* untuk menu yang dapat diakses *user*

Aktivitas dan Event	Input	Output	Status Pengujian
Lihat jadwal keberangkatan	Memilih tanggal, asal, dan tujuan keberangkatan	a. Jika jadwal ditemukan maka akan muncul menu lihat data b. Jika jadwal tidak ditemukan muncul komentar "tidak ada jadwal keberangkatan"	valid
<i>User</i> memesan tiket	Memilih keberangkatan tempat duduk.	agen dan a. Jika pilih pesan maka masuk ke list pemesanan b. Jika batal maka kembali ke menu lihat data	valid
Melakukan verifikasi pembayaran	Memasukkan transfer pembayaran	kode a. Jika tidak memasukkan kode transfer maka akan muncul peringatan : *kode belum di isi b. Jika telah memasukkan kode transfer maka masuk ke menu list tiket	valid
Melihat <i>list</i> tiket		Menampilkan daftar tiket yang dipesan	valid

- c. Pengujian untuk menu-menu yang dapat diakses *admin* didapatkan kesimpulan bahwa telah berfungsi dengan baik

**Tabel 3** Pengujian *Black Box* untuk menu yang dapat diakses *admin*

Aktivitas			
dan	Input	Output	Status Pengujian
Event			
Admin tambah data armada	Masukkan no.polisi, jenis, jumlah sit dan keterangan	Jika data baru telah dimasukkan maka akan dilakukan proses tambah data armada	valid
Admin tamba data rute	Memasukkan no.polisi, harga/sit,tanggal keberangkatan, jam, asal keberangkatan dan tujuan	Jika data baru telah dimasukkan maka akan dilakukan proses tambah data rute	valid
Admin melihat daftar pemesanan		Jika tombol ditekan maka muncul tabel pemesanan	valid
Admin melihat data penumpang		Jika tombol ditekan maka muncul tabel daftar penumpang	valid
Logout dan Klik tombol <i>logout</i>		Jika <i>user</i> , dan <i>admin</i> menekan tombol <i>logout</i> maka mereka akan keluar dari sesi	valid

Berdasarkan pengujian yang dilakukan didapatkan hasil bahwa aplikasi ini sudah berjalan secara fungsional dan mengeluarkan informasi sesuai dengan yang diharapkan.

### Pengujian Beta

Kelompok pengujian beta tidak menyertakan orang-orang yang terlibat dalam pembuatan aplikasi. Pada pengujian ini dilakukan dengan cara memberikan tes kuisisioner yaitu menjawab kuisisioner sesudah menggunakan aplikasi. Pengujian ini digunakan untuk mengetahui sejauh mana aplikasi ini berfungsi dengan baik. Pengujian dilakukan dengan cara pembagian kuisisioner kepada *sample user*. *Sample user* berjumlah 30 responden yang berumur 17-30 tahun yang rata-rata dalam sebulan menggunakan jasa transportasi bus sebanyak 1-4 kali. Responden berasal dari berbagai kalangan seperti guru, mahasiswa, siswa, karyawan dan swasta. Dari data kuisisioner yang diisi para responden, diketahui jumlah jawaban untuk setiap nomor, yang dirangkum pada Tabel 4.

**Tabel 4** Tabel Jumlah Jawaban Kuisisioner

Jawaban Pertanyaan	A	B	C	D
1	6	15	7	2
2	11	12	6	1
3	4	17	6	3
4	19	11	-	-
5	24	6	-	-

Tahap selanjutnya adalah menguji tanggapan dari para responden yang sudah mengisi kuisisioner, dihitung dengan menggunakan skala Likert untuk menghitung skala pengukuran variabel, masing-masing kategori jawaban diberi skala skor 1-4.

1. Sangat Baik (SB) = 4
2. Baik (B) = 3
3. Cukup (C) = 2
4. Tidak Baik = 1

Berdasarkan hasil analisa jawaban kuesioner yang telah dilakukan, pada pertanyaan pertama 20 % responden menjawab sangat mudah digunakan dan 50% menjawab mudah digunakan, sehingga untuk pertanyaan pertama dapat disimpulkan bahwa 70% responden menyatakan aplikasi ini mudah digunakan.

Berdasarkan hasil analisa jawaban kuesioner yang telah dilakukan pada pertanyaan kedua 36,67% responden menjawab sangat membantu dan 40% menjawab membantu, sehingga untuk pertanyaan kedua dapat disimpulkan bahwa 76,67% responden menyatakan aplikasi ini dapat membantu dan mempermudah *customer* dalam memesan tiket bus.

Berdasarkan hasil analisa jawaban kuesioner pertanyaan ketiga 13,33% responden menjawab sangat mudah dipahami dan 56,67% menjawab mudah dipahami, sehingga untuk pertanyaan ketiga dapat disimpulkan bahwa 70% responden menyatakan cara pemesanan tiket pada aplikasi ini mudah dipahami.

Berdasarkan hasil analisa jawaban kuesioner pertanyaan ke-empat 63,33% responden menjawab *fleksibel*, sehingga untuk pertanyaan keempat dapat disimpulkan bahwa 63,33% *customer* menyatakan aplikasi *reservasi* ini lebih *fleksibel* dari sistem sebelumnya.

Berdasarkan hasil analisa jawaban kuesioner yang telah dilakukan pada pertanyaan terakhir 80% responden menjawab setuju, sehingga untuk pertanyaan kelima dapat disimpulkan bahwa 80% *customer* menyatakan setuju untuk mengganti sistem *reservasi* yang lama dengan aplikasi *reservasi mobile* ini.

## 5. Kesimpulan

Dari hasil penelitian dan pengujian yang telah dilakukan, dapat diambil simpulan bahwa aplikasi *reservasi* tiket bus berbasis Android dapat diterapkan pada PO. Rosalia Indah untuk penjualan tiket seluruh trayek. Aplikasi *mobile* dengan tampilan yang sederhana dan *user input* yang tidak rumit dapat digunakan dalam melakukan *reservasi* tiket secara *online*. Aplikasi ini dapat membantu *user* menghemat waktu, biaya, dan tenaga dalam pembelian suatu produk, karena menggunakan *mobile device* yang bersifat *ubiquity* sehingga transaksi bisa dilakukan kapan saja dan dimana saja.

## 6. Daftar Pustaka/Pustaka Acuan

- [1] Rosalia Indah. 2012. *Company Profile*. Surakarta.
- [2] Microsoft Corp. (2000) *Application Service Provider: Evolution and Resources*, White Paper, USA.
- [3] Janwar. 2010. *Perancangan aplikasi Pemesanan Tiket dan Penjadwalan Keberangkatan Bus Antar Propinsi Berbasis web (Studi kasus: PO. Maju Lancar)*. Jakarta: Universitas Mercu Buana
- [4] Faizal. 2011. *Aplikasi Layanan Pesan Makanan Pada Restoran "XYZ" Dengan platform Android*. Bandung : Politeknik Telkom.
- [5] Nazruddin Safaat H. 2011. *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung : Informatika
- [6] Android Developer. 2011. What is Android ?. Diakses tanggal 25 November 2011.
- [7] Kreger, H., 2001, *Web-services Conceptual Architecture (WSCA 1.0)*, IBM Software Group, USA
- [8] Lily, dkk. 2006. *Toward Web Service*. Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2006).
- [9] Pressman, Roger, 2002, *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku Satu)*, Yogyakarta: Penerbit ANDI dan McGraw-Hill Book Co
- [10] Nugroho, Adi. 2005. *Rational Rose Untuk Pemodelan Berorientasi Objek*. Bandung: Informatika.
- [11] UML 1.3a IphaR S, Object Management Group Inc, March 1999.
- [12] Vaughan, Tay. 2004. *Multimedia: Making It Work*, Edisi 6. Yogyakarta: ANDI.