



ARTIKEL ILMIAH

## Aplikasi Business to Business Marketplace Jasa Pariwisata Menggunakan Kotlin

I Wayan Dharmana<sup>1</sup>, Ricky Aurelius Nurtanto Diaz<sup>2</sup>, Ni Luh Ratniasih<sup>3</sup>

<sup>1,3</sup>) Sistem Informasi, STMIK STIKOM Bali, <sup>2</sup>) Sistem Komputer, STMIK STIKOM Bali

Jl. Raya Puputan No. 86 Renon Denpasar, (0361) 244445

Email : [dharmana@stikom-bali.ac.id](mailto:dharmana@stikom-bali.ac.id), [ricky@stikom-bali.ac.id](mailto:ricky@stikom-bali.ac.id)



**Citation:** Nama Penulis, "Aplikasi Business to Business Marketplace Jasa Pariwisata Menggunakan Kotlin" in *Jurnal MEDIA APLIKOM*, Vol.11 No. 1, Purwokerto: STIKOM Yos Sudarso Publisher. 2019, pp. 49-73.

**Editor:** Diwahana Mutiara Candrasari

**Received:** Bulan Mei, 2019

**Accepted:** Bulan Mei , 2019

**Published:** 01 Juni, 2019

**Funding:** Mandiri

**Copyright:** ©2019

I Wayan Dharmana, Ricky Aurelius Nurtanto Diaz, Ni Luh Ratniasih



### Abstract

*Tourism industry has the business process of tourism service transaction between merchant and agent. That business process between merchant and agent is known as business-to-business (B2B). Merchant has role as inventory providers and agent as sellers. Based on the information, the transaction between the merchant and agent is still done conventionally. The merchant has to find the potential agent that ready to sell their products, negotiate special rate and the term of payment. Long negotiation process makes the marketing coverage become less. On the agent side, the sales potential isn't maximal because the limitation of product that they can sell. Based on the explanation above, it is necessary to build an app as solution with concept of business to business tourism service marketplace. Android mobile application was developed to fulfill the merchant and agent necessary in mobility. Application was developed with features such as inventory management, stock management, transaction, booking management, and fund management (topup, withdraw and payment history). The transaction done by the agent will be paid by using active balance that can be added by bank transfer to virtual account. Kotlin was used by the writer to develop the application and black box testing method for testing*

**Keywords:** android, business-to-business, marketplace, kotlin, tourism



### Abstrak

Industri pariwisata memiliki proses bisnis jual beli jasa pariwisata yang melibatkan usaha jasa pariwisata (*merchant*) serta agen (*agent*). Proses bisnis antara dua pelaku industri ini dikenal dengan istilah *business-to-business* (B2B). Usaha jasa pariwisata berperan sebagai penyedia jasa dan agen sebagai penjual jasa. Berdasarkan informasi yang ada, transaksi antara usaha jasa pariwisata dengan agen masih dilakukan secara konvensional. Usaha Jasa Pariwisata harus mencari agen potensial yang siap menjual produk mereka serta melakukan negosiasi harga khusus dan termin pembayaran. Proses negosiasi yang lama membuat jangkauan pemasaran menjadi kurang. Dari sisi agen, potensi penjualan kurang maksimal karena keterbatasan produk yang dapat dijual. Berdasarkan penjelasan tersebut, maka dibutuhkan solusi aplikasi dengan konsep *business-to-business marketplace* jasa pariwisata. Aplikasi yang dikembangkan berbasis Android agar dapat memenuhi kebutuhan mobilitas usaha jasa pariwisata dan agen. Aplikasi ini dikembangkan dengan fitur manajemen jasa, manajemen stok, transaksi reservasi, manajemen reservasi, serta pengelolaan dana (*topup*, *withdraw* dan riwayat pembayaran). Transaksi pembelian jasa pariwisata oleh agen dibayar menggunakan saldo aktif yang dapat ditambah dengan transfer dana melalui transfer bank ke *virtual account*. Penulis melakukan pengembangan aplikasi menggunakan bahasa pemrograman Kotlin dan metode *black box testing* untuk pengujian.

**Kata kunci:** android, *business-to-business*, *marketplace*, kotlin, pariwisata

---

### Pendahuluan

Industri pariwisata tidak hanya memiliki proses jual beli antara usaha jasa pariwisata dan wisatawan atau konsumen akhir. Proses jual beli juga dilakukan antara usaha jasa pariwisata dengan agen. Usaha jasa pariwisata memiliki peran menyediakan jasa maupun daya tarik wisata yang memiliki nilai jual dan dapat dipasarkan kepada wisatawan. Sedangkan agen memiliki peran untuk membantu usaha jasa pariwisata dalam memasarkan produk jasa yang dikelola. Dalam proses bisnis tersebut, terdapat kesepakatan mengenai harga antara kedua pihak ini. Dengan harga khusus ini, agen memiliki kesempatan untuk meningkatkan margin keuntungannya dalam setiap penjualan.



Dalam praktik bisnis di lapangan, usaha jasa pariwisata harus mencari agen-agen yang ingin diajak bekerja sama dalam hal pemasaran jasa yang dimiliki. Kedua pihak ini harus saling berdiskusi untuk mendapatkan kata sepakat mengenai harga, mekanisme pemesanan hingga termin pembayaran. Tentu kegiatan ini memerlukan waktu yang lama. Dengan waktu yang lama, tentu mempengaruhi jangkauan pemasaran yang dapat dilakukan. Hal yang sama juga dialami oleh agen penjual. Agen mengalami keterbatasan inventori jasa atau produk yang dapat dijual karena harus lebih sering melakukan negosiasi harga dengan pemilik usaha jasa pariwisata. Tentu ini membuat peluang penjualan agen menjadi kurang maksimal karena inventori yang dimiliki juga terbatas. Dari sisi mekanisme pemesanan, agen membuat voucher dan mengkonfirmasi secara manual kepada usaha jasa pariwisata. Proses ini rawan terjadi kesalahan teknis seperti *overbooking* dan pemesanan yang tidak terkonfirmasi. Tentu hal ini tidak hanya merugikan agen dan usaha jasa pariwisata. Wisatawan sebagai pengguna jasa tersebut merasa dirugikan karena pelayanan yang tidak maksimal.

Berdasarkan penjelasan sebelumnya, dibutuhkan solusi yang dapat memenuhi kebutuhan kedua pihak. Bagi usaha jasa pariwisata, mereka membutuhkan solusi untuk memperluas pemasaran melalui agen-agen yang berpotensi menjual jasa yang mereka miliki. Tentu tanpa biaya banyak untuk melakukan ekspansi ke daerah lain. Sedangkan agen membutuhkan solusi agar mereka lebih mudah menemukan jasa yang dapat dijual dengan harga khusus dan penyedia yang terpercaya. Dari kedua pihak tentu menginginkan adanya mekanisme pemesanan yang mudah, pasti dan efisien dalam proses bisnis mereka. Saat ini, industri pariwisata sudah menggunakan teknologi untuk berkomunikasi berupa telepon, SMS, *e-mail* maupun aplikasi *instant messaging*. Teknologi ini hanya sebatas membantu kedua pihak untuk saling berkomunikasi baik lewat media suara maupun teks. Hal tersebut belum mampu mengatasi permasalahan secara maksimal untuk hal pemasaran serta mekanisme pemesanan. Berdasarkan masalah tersebut, dibutuhkan inovasi teknologi yang dapat membantu transaksi bisnis kedua pihak ini. Salah satu



inovasi yang dapat dikembangkan adalah layanan *e-commerce* yang menghubungkan usaha jasa pariwisata dengan agen.

Perdagangan elektronik atau yang disebut juga *e-commerce*, adalah penggunaan jaringan komunikasi dan komputer untuk melaksanakan proses bisnis. Pandangan populer dari *e-commerce* adalah penggunaan internet dan komputer dengan *browser* Web untuk membeli dan menjual produk [1]. Dalam *e-commerce* terdapat klasifikasi *business-to-business* yang merupakan konsep interaksi perdagangan dengan menggunakan jaringan internet antar pelaku bisnis [2]. Konsep yang melibatkan pihak yang menjadi pembeli dan penjual dikenal juga dengan istilah *Online Marketplace*. Sebuah pasar online yang mempertemukan penjual dan pembeli dengan tidak harus bertatap muka atau bertemu langsung. Dalam hal ini, usaha jasa pariwisata bertindak sebagai penjual. Sedangkan agen bertindak sebagai pembeli dari jasa yang ditawarkan oleh usaha jasa pariwisata dengan harga khusus agen. *E-commerce* diharapkan tidak hanya menjadi media pemasaran saja. Dengan adanya layanan *e-commerce* diharapkan dapat membantu dalam hal sistem penerbitan voucher yang selama ini manual. Dengan menggunakan konsep ini diharapkan proses bisnis dapat berjalan secara otomatis dan meningkatkan efisiensi di antara dua entitas bisnis.

### Metode Penelitian

#### 1. Pengumpulan Data

Dalam penelitian ini, terdapat beberapa langkah pengumpulan data yang dilakukan. Adapun pengumpulan data tersebut terdiri atas:

##### a. Studi Literatur

Studi literatur merupakan metode pengumpulan data dan informasi dengan cara menggali pengetahuan atau ilmu dari beberapa sumber yang terkait dengan pembuatan aplikasi android melalui buku-buku, jurnal dan karya tulis serta mempelajari SOP yang terdapat pada industri pariwisata.

##### b. Observasi



Observasi merupakan salah satu metode pengumpulan data dan informasi yang dilakukan dengan melakukan pengamatan langsung terhadap objek yang akan diteliti atau direkayasa untuk memperoleh gambaran mengenai sistem yang berjalan atau prosedur yang ada pada obyek perekayasaan tersebut yaitu dalam hal ini dengan mengunjungi perusahaan yang menjadi calon pengguna aplikasi serta mempelajari proses bisnisnya. Hasil observasi ini nantinya akan dijadikan acuan dalam pembuatan aplikasi android *business-to-business marketplace* jasa pariwisata. Hal ini dilakukan agar program aplikasi yang akan dibangun nantinya dapat digunakan secara maksimal dan sesuai dengan kebutuhan pengguna.

c. Kuesioner

Kuesioner adalah teknik pengumpulan data melalui formulir-formulir yang berisi pertanyaan-pertanyaan yang diajukan secara tertulis pada seseorang atau sekumpulan orang untuk mendapatkan jawaban atau tanggapan dan informasi yang diperlukan oleh peneliti. Kuesioner digunakan oleh penulis untuk memperoleh penilaian dari pengguna mengenai kemudahan penggunaan aplikasi.

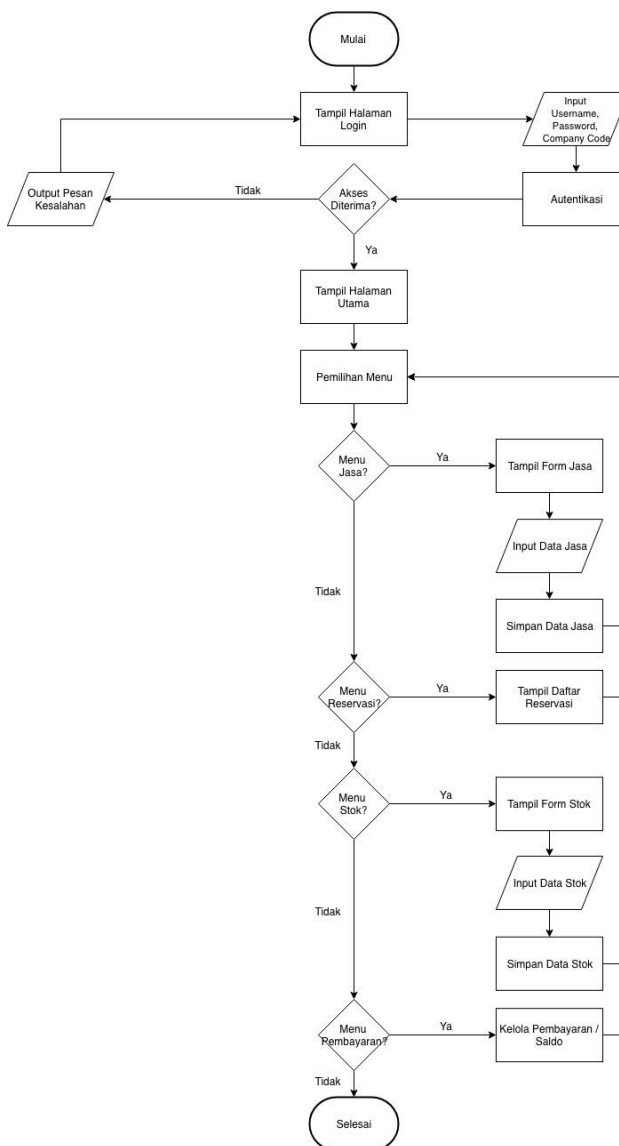
2. Desain Alur Sistem

Alur sistem dibagi menjadi dua sesuai dengan jenis pengguna, yaitu Usaha Jasa Pariwisata dan Agen. Berikut adalah gambaran dari *Flowchart* sistem :

a. Usaha Jasa Pariwisata

Pihak usaha jasa pariwisata secara umum memiliki alur proses bisnis yang berkaitan dengan manajemen jasa / *inventory*, manajemen reservasi, manajemen stok, serta manajemen pembayaran / saldo aktif yang dimiliki. *Flowchart* proses bisnis usaha jasa pariwisata dapat dilihat pada Gambar 1 :





Gambar 1 *Flowchart* Usaha Jasa Pariwisata

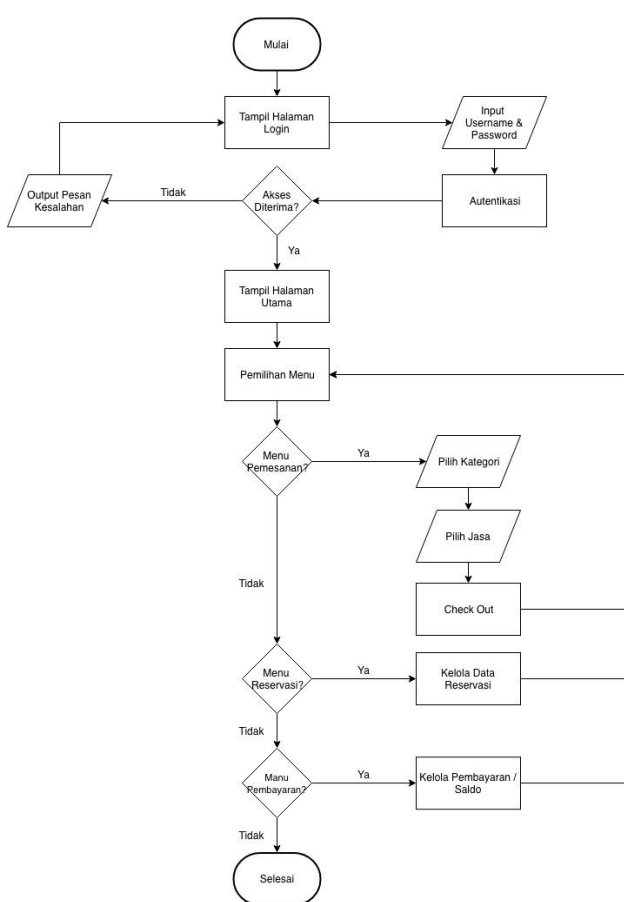
Proses bisnis usaha jasa pariwisata dimulai dengan proses login menggunakan *username*, *password* serta *company code*. *Company code* merupakan identitas khusus berupa kode singkat yang mewakili *username* dari bisnis. Selanjutnya, data yang dikirim harus melalui proses autentikasi. Pengguna akan memiliki akses ke halaman utama yang memiliki beberapa menu utama. Menu jasa / inventori untuk melihat daftar jasa, input baru, dan *update* data jasa. Menu reservasi memungkinkan pengguna untuk melihat daftar reservasi yang masuk. Setiap inventori memiliki stok yang dapat dikelola melalui menu Stok. Selanjutnya, menu pembayaran



memungkinkan pengguna untuk melihat dan menarik saldo aktif ke rekening.

b. Agen

Pihak agen secara umum memiliki proses bisnis yang berkaitan dengan transaksi pemesanan, manajemen reservasi, serta manajemen pembayaran atau saldo aktif. *Flowchart* proses bisnis agen dapat dilihat pada Gambar 2.



Gambar 2 *Flowchart* Agen

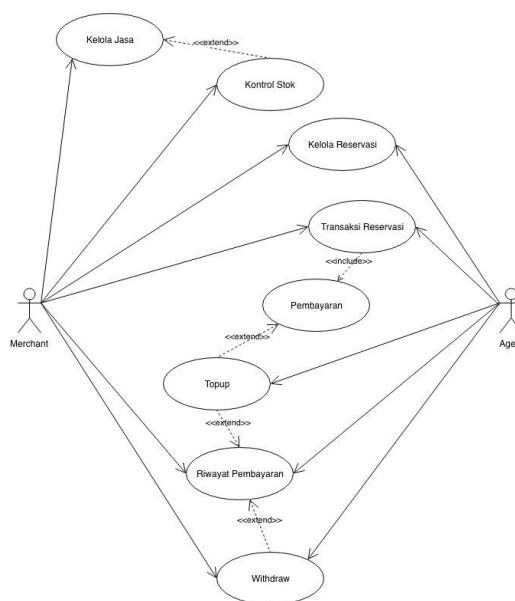
Proses bisnis agen dimulai dengan proses login menggunakan email dan password. Selanjutnya, data yang dikirim harus melalui proses autentikasi untuk memastikan agen memiliki hak akses. Pengguna akan memiliki akses ke halaman utama yang memiliki beberapa menu utama. Menu pemesanan memungkinkan agen untuk memilih kategori, memilih jasa dan *checkout* dengan saldo yang dimiliki. Menu reservasi digunakan



agen untuk mengelola reservasi yang telah dibuat melalui menu pemesanan. Selanjutnya, menu pembayaran memungkinkan agen untuk melihat, *topup* dan menarik saldo aktif ke rekening.

### Hasil dan Pembahasan

Dalam pengembangan aplikasi ini, proses perancangan menggunakan *Unified Modelling Language* (UML) yang merupakan suatu *tools* untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berguna dalam pengembangan sistem selanjutnya [4]. Model pertama yang dirancang adalah *use case diagram* yang digunakan untuk merancang sistem berdasarkan kelakuan (*behavior*) sistem informasi yang akan dibuat [5]. Dalam *use case diagram* ini terdapat dua aktor yakni *merchant* dan *agent*. Rancangan *Use Case Diagram* dapat dilihat pada Gambar 3.



Gambar 3. Rancangan *Use Case Diagram*

Dalam *use case diagram* ini terdapat delapan *use case* yang digunakan oleh aktor *merchant* dan *agent*. *Use case* tersebut meliputi kelola jasa, kelola stok, kelola reservasi, transaksi reservasi, pembayaran, *topup*, riwayat pembayaran serta *withdraw*. Setiap *use case* dirancang sesuai fungsi masing-masing untuk menunjang kebutuhan aktor.





Setiap proses *use case diagram* akan dijabarkan dalam *high level use case* serta *expanded use case*. *High level use case* memiliki fungsi untuk menjelaskan fungsi *use case* secara umum dengan singkat. *Expanded use case* memberikan penjelasan lebih detail serta pengertian yang lebih mendalam mengenai *process* serta *requirement*. Setiap proses dalam *use case* akan dijabarkan sebagai berikut

1. Kelola Jasa

*Use case* kelola jasa melibatkan aktor *merchant* untuk mengelola data inventori yang dimiliki masing-masing *merchant*. *Use case* ini juga meliputi proses tambah, *update* serta hapus produk jasa pariwisata yang dimiliki. *Expanded use case* kelola jasa dapat dilihat pada Tabel 1.



<b>Use Case</b>	Kelola Jasa
<b>Goal in Context</b>	Memberikan akses kepada <i>merchant</i> untuk mengelola produk jasa pariwisata
<b>Primary Actor</b>	<i>Merchant</i>
<b>Secondary Actor</b>	-
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. <i>Use case</i> dimulai saat <i>merchant</i> bermaksud ingin mengelola jasa pariwisata yang dimiliki.	
2. <i>Merchant</i> memilih tab <i>Inventory</i> .	3. Sistem melakukan pengecekan terhadap data produk jasa pariwisata yang dimiliki <i>merchant</i> .
	4. Sistem menampilkan daftar produk jasa pariwisata.
5. <i>Merchant</i> memilih menu tambah.	6. Sistem menampilkan form tambah produk jasa pariwisata.
7. <i>Merchant</i> melakukan input data produk jasa pariwisata.	8. Sistem menyimpan data produk jasa pariwisata yang diinput oleh <i>merchant</i> .
	9. Apabila proses simpan berhasil, maka akan muncul pesan sukses dan sistem menampilkan daftar produk jasa pariwisata.
10. <i>Merchant</i> memilih salah satu produk jasa pariwisata dan memilih opsi <i>edit</i> .	11. Sistem mengecek data produk jasa pariwisata yang dipilih.
	12. Sistem menampilkan form edit untuk produk jasa pariwisata yang dipilih.
13. <i>Merchant</i> melakukan update	14. Sistem melakukan update data



data produk jasa pariwisata.	produk jasa pariwisata sesuai yang diinput oleh <i>merchant</i> .
	15. Apabila proses berhasil, maka muncul pesan berhasil dan menampilkan daftar produk jasa pariwisata.
16. <i>Merchant</i> memilih salah satu produk jasa pariwisata dan memilih opsi hapus.	17. Sistem melakukan hapus data sesuai produk jasa pariwisata yang dipilih.
	18. Sistem menampilkan daftar produk jasa pariwisata.
<b>Alternative Courses</b>	
Line 4: Apabila data tidak ditemukan, maka muncul pesan tidak ditemukan.	
Line 9: Apabila data diinput belum valid, sistem menampilkan pesan kesalahan.	
Line 15: Apabila data diinput belum valid, sistem menampilkan pesan kesalahan.	

## 2. Kontrol Stok

*Use case* kontrol stok melibatkan aktor *merchant* untuk melakukan pengolahan data stok dari masing-masing produk jasa pariwisata. *Expanded use case* kontrol stok dapat dilihat pada Tabel 2.

**Tabel 2 Expanded Use Case Kontrol Stok**

<b>Use Case</b>	Kontrol Stok
<b>Goal in Context</b>	Memberikan akses kepada <i>merchant</i> untuk mengontrol stok jasa pariwisata
<b>Primary Actor</b>	<i>Merchant</i>
<b>Secondary Actor</b>	-
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>



1. <i>Use case</i> dimulai saat <i>merchant</i> ingin mengelola jasa pariwisata yang dimiliki.	
2. <i>Merchant</i> memilih produk dan memilih opsi <i>manage inventory</i> .	3. Sistem mengecek data stok sesuai produk jasa pariwisata yang dipilih.
	4. Sistem menampilkan data stok sesuai produk jasa pariwisata yang dipilih.
5. <i>Merchant</i> memilih salah satu tanggal.	6. Sistem mengecek data stok sesuai tanggal yang dipilih.
	7. Sistem menampilkan data stok sesuai tanggal.
8. Apabila dibutuhkan, <i>merchant</i> memilih tombol <i>update</i> untuk melakukan <i>update</i> stok.	9. Sistem menampilkan <i>form update</i> stok.
10. <i>Merchant</i> input data stok / kapasitas baru.	11. Sistem menyimpan stok / kapasitas baru.
	12. Apabila berhasil, sistem menampilkan pesan berhasil dan muncul tampilan data stok yang telah diupdate.
<b>Alternative Courses</b>	
Line 12: apabila data stok / kapasitas yang dimasukkan tidak valid, muncul pesan kesalahan	

### 3. Kelola Reservasi

*Use case* kelola reservasi melibatkan aktor *merchant* dan *agent* untuk melakukan pengolahan data reservasi yang masuk. *Agent* membuat transaksi pembelian dan dapat mengakses riwayat transaksi melalui fitur ini. *Merchant* mengakses fitur ini untuk mengetahui reservasi yang masuk sesuai dengan

produk jasa pariwisata yang dijual. *Expanded use case* kelola reservasi dapat dilihat pada Tabel 3.

Tabel 3 *Expanded Use Case* Kelola Reservasi

<b>Use Case</b>	Kelola Reservasi
<b>Goal in Context</b>	<i>Merchant</i> dan <i>Agent</i> dapat mengakses informasi reservasi yang dimiliki
<b>Primary Actor</b>	<i>Merchant</i>
<b>Secondary Actor</b>	-
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. <i>Use case</i> dimulai saat <i>merchant</i> dan <i>agent</i> mengecek daftar reservasi.	

2. <i>Merchant</i> dan <i>agent</i> memilih tab <i>Booking</i> pada navigasi.	3. Sistem mengecek data reservasi sesuai pemilik.
	4. Sistem menampilkan daftar reservasi.
5. <i>Merchant</i> dan <i>agent</i> memilih salah satu reservasi.	6. Sistem mengecek data reservasi sesuai yang dipilih oleh pengguna.
	7. Sistem menampilkan halaman detail reservasi.
<b>Alternative Courses</b>	
Line 4: Apabila data tidak ditemukan, sistem akan menampilkan pesan bahwa data tidak ditemukan.	
Line 7: Apabila data reservasi tidak ditemukan, sistem akan menampilkan pesan bahwa data reservasi tidak ditemukan.	

#### 4. Transaksi Reservasi

*Use case* kelola reservasi melibatkan aktor *agent* dan *merchant* untuk melakukan transaksi pembelian produk jasa pariwisata. *Agent* memiliki peran



untuk membeli produk jasa pariwisata melalui aplikasi. *Merchant* memiliki peran untuk menerima pesanan yang masuk melalui notifikasi. *Expanded use case* transaksi reservasi dapat dilihat pada Tabel 4.

Tabel 4 *Expanded Use Case* Transaksi Reservasi

<i>Use Case</i>	Transaksi Reservasi
<i>Goal in Context</i>	<i>Agent</i> melakukan pembelian produk jasa pariwisata dan <i>merchant</i> menerima notifikasi.
<i>Primary Actor</i>	<i>Agent</i>
<i>Secondary Actor</i>	<i>Merchant</i>
<i>Typical Course of Events</i>	
<i>Actor Action</i>	<i>Sytem Response</i>
1. <i>Use case</i> dimulai saat <i>agent</i> ingin melakukan pembelian produk jasa pariwisata.	
2. <i>Agent</i> memilih salah satu kategori atau lokasi di menu utama.	3. Sistem mengecek daftar produk jasa pariwisata sesuai kategori atau lokasi.
	4. Sistem menampilkan daftar produk jasa pariwisata sesuai kategori atau lokasi.

5. <i>Agent</i> memilih salah satu produk jasa pariwisata.	6. Sistem mengecek data produk jasa pariwisata sesuai yang dipilih <i>agent</i> .
	7. Sistem menampilkan halaman detail produk jasa pariwisata.
8. <i>Agent</i> memilih tombol <i>Select</i> .	9. Sistem menampilkan halaman form pemesanan.
10. <i>Agent</i> mengisi data pemesanan.	11. Sistem menyimpan data pemesanan.
	12. Apabila berhasil, sistem akan menampilkan halaman daftar tamu.
13. <i>Agent</i> memilih salah satu kolom daftar tamu yang ingin diisi.	14. Sistem mengecek data tamu sesuai yang dipilih.
	15. Sistem menampilkan form tamu.
16. <i>Agent</i> mengisi data tamu (nama tamu, panggilan, dan jenis tamu).	
17. <i>Agent</i> memilih tombol <i>Update</i> .	18. Sistem menyimpan data tamu.
	19. Sistem menampilkan kembali halaman daftar tamu.
20. <i>Agent</i> memilih tombol <i>Continue</i> .	21. Sistem menyimpan daftar tamu.
	22. Apabila berhasil, sistem akan menampilkan halaman <i>Checkout</i> .
23. <i>Agent</i> memilih tombol <i>Pay Now</i> .	24. Sistem melakukan proses verifikasi pembayaran.
	25. Sistem menyimpan data reservasi.
	26. Apabila berhasil, sistem akan memunculkan pesan berhasil dan mengalihkan ke halaman utama.
	27. Sistem akan mengirimkan notifikasi ke <i>merchant</i> melalui <i>real time notification</i> serta <i>email</i> .
28. <i>Merchant</i> menerima notifikasi	



reservasi.	
<b>Alternative Courses</b>	
Line 4: Apabila data tidak ditemukan, maka muncul pesan data tidak ditemukan.	
Line 7: Apabila data tidak ditemukan, maka muncul pesan data tidak ditemukan.	
Line 12a: Apabila data belum valid atau belum lengkap, maka muncul pesan peringatan.	
Line 12b: Apabila stok sesuai tanggal yang dipilih belum mencukupi, maka muncul pesan peringatan	
Line 19: Apabila data tidak valid atau belum lengkap, maka muncul pesan peringatan.	
Line 22: Apabila data tidak valid atau belum lengkap, maka muncul pesan peringatan.	
Line 26: Apabila terjadi masalah pembayaran, maka akan muncul pesan kesalahan.	

#### 5. Pembayaran

*Use case* pembayaran melibatkan aktor *agent* untuk melakukan pembayaran saat transaksi pembelian produk jasa pariwisata. *Expanded use case* pembayaran dapat dilihat pada Tabel 5.



Tabel 5 *Expanded Use Case* Transaksi Pembayaran

<b>Use Case</b>	Pembayaran	
<b>Goal in Context</b>	<i>Agent</i> melakukan pembayaran saat proses <i>checkout</i> reservasi.	
<b>Primary Actor</b>	<i>Agent</i>	
<b>Secondary Actor</b>	-	
<b>Typical Course of Events</b>		
<b>Actor Action</b>	<b>System Response</b>	
1. <i>Use case</i> dimulai saat <i>agent</i> melakukan proses <i>checkout</i> transaksi pembelian produk jasa pariwisata.		
2. <i>Agent</i> memilih tombol <i>Pay Now</i> .	3. Sistem menampilkan halaman verifikasi PIN.	
4. <i>Agent</i> memasukkan PIN.	5. Sistem menampilkan kembali halaman <i>checkout</i> .	
	6. Sistem menyimpan data pembayaran.	
	7. Apabila berhasil, sistem akan memunculkan pesan berhasil dan mengalihkan ke halaman utama.	
<b>Alternative Courses</b>		
Line 7a: Apabila PIN salah, maka muncul pesan peringatan.		
Line 7b: Apabila saldo tidak mencukupi, maka muncul pesan peringatan.		

#### 6. *Topup*

*Use case topup* melibatkan aktor *agent* untuk melakukan penambahan saldo aktif yang dapat digunakan untuk media pembayaran reservasi. *Expanded use case topup* dapat dilihat pada Tabel 6.

Tabel 6 *Expanded Use Case Topup*



<i>Use Case</i>	<i>Topup</i>
<i>Goal in Context</i>	<i>Agent melakukan topup saldo.</i>
<i>Primary Actor</i>	<i>Agent</i>
<i>Secondary Actor</i>	-
<i>Typical Course of Events</i>	
<i>Actor Action</i>	<i>Sytem Response</i>
1. <i>Use case</i> dimulai saat <i>agent</i> ingin melakukan <i>topup</i> saldo.	
2. <i>Agent</i> memilih menu <i>Topup</i> .	3. Sistem mengecek daftar metode pembayaran yang tersedia.
	4. Sistem menampilkan daftar metode pembayaran yang tersedia.
5. <i>Agent</i> memilih salah satu metode pmebayaran.	6. Sistem mengecek ketersediaan <i>virtual account</i> sesuai metode pembayaran.
	7. Apabila ditemukan, sistem menampilkan halaman instruksi pembayaran serta nomor <i>virtual account</i> .
8. <i>Agent</i> melakukan transfer dana ke nomor <i>virtual account</i> menggunakan aplikasi perbankan.	9. Sistem mengecek dana yang ditransfer.
	10. Sistem menambahkan saldo <i>agent</i> .
	11. Sistem mengirimkan <i>real time notification</i> ke <i>agent</i> .
<i>Alternative Courses</i>	
Line 7: Apabila tidak ditemukan, sistem melakukan <i>request</i> ke API NICEPAY untuk mendaftarkan <i>virtual account</i> sesuai <i>agent</i> dan metode pembayaran dipilih.	



## 7. Riwayat Pembayaran

*Use case* riwayat pembayaran melibatkan aktor *merchant* dan *agent* untuk mendapatkan informasi mengenai arus kas dari saldo yang dimiliki. *Expanded use case* riwayat pembayaran dapat dilihat pada Tabel 7.

Tabel 7 *Expanded Use Case* Riwayat Pembayaran

<i>Use Case</i>	Riwayat Pembayaran
<i>Goal in Context</i>	<i>Merchant</i> dan <i>Agent</i> dapat mengakses informasi arus kas saldo.
<i>Primary Actor</i>	<i>Merchant</i> dan <i>Agent</i>
<i>Secondary Actor</i>	-
<i>Typical Course of Events</i>	
<i>Actor Action</i>	<i>Sytem Response</i>
1. <i>Use case</i> dimulai saat <i>merchant</i> dan <i>agent</i> ingin mengetahui informasi arus kas saldo.	
2. <i>Merchant</i> dan <i>Agent</i> memilih menu <i>Payment History</i> .	3. Sistem mengecek daftar riwayat pembayaran.
	4. Apabila tersedia, sistem akan menampilkan daftar riwayat pembayaran.
<i>Alternative Courses</i>	
Line 4: Apabila data tidak ditemukan, maka muncul pesan data tidak ditemukan.	

8. *Withdraw*

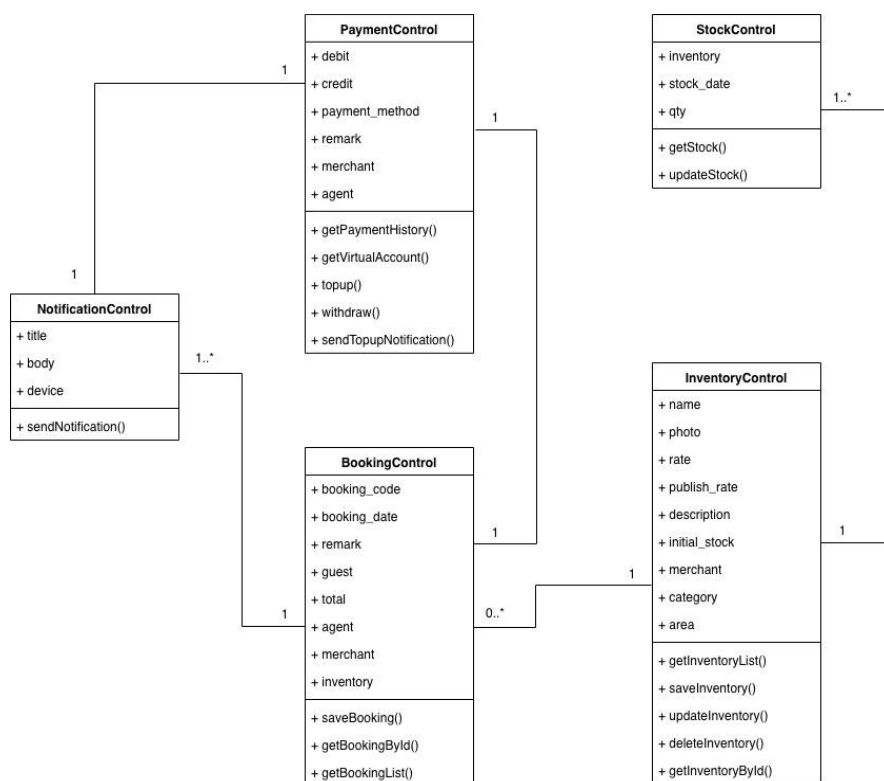
*Use case withdraw* melibatkan aktor *merchant* dan *agent* untuk membuat permintaan penarikan dana dari saldo aktif ke rekening pemilik. *Expanded use case withdraw* dapat dilihat pada Tabel 8.

Tabel 8 *Expanded Use Withdraw*

<i>Use Case</i>	<i>Withdraw</i>
<i>Goal in Context</i>	<i>Merchant dan Agent dapat membuat permintaan penarikan dana.</i>
<i>Primary Actor</i>	<i>Merchant dan Agent</i>
<i>Secondary Actor</i>	-
<i>Typical Course of Events</i>	
<i>Actor Action</i>	<i>Sytem Response</i>
1. <i>Use case</i> dimulai saat <i>merchant</i> dan <i>agent</i> ingin melakukan penarikan dana.	
2. <i>Merchant</i> dan <i>Agent</i> memilih menu <i>Withdraw</i> .	3. Sistem mengecek saldo aktif.
	4. Sistem menampilkan form <i>withdraw</i> .
5. <i>Merchant</i> dan <i>Agent</i> mengisi data <i>withdraw</i> (jumlah dana dan informasi rekening tujuan).	
6. <i>Merchant</i> dan <i>Agent</i> memilih tombol <i>withdraw</i> .	7. Sistem menyimpan data <i>withdraw</i> .
	8. Apabila berhasil, sistem menampilkan pesan sukses.
<i>Alternative Courses</i>	
Line 8a: Apabila data rekening belum valid, muncul pesan peringatan.	
Line 8b: Apabila jumlah dana melebihi saldo aktif, muncul pesan peringatan	

Selain *use case diagram*, penulis juga menggambarkan rancangan sistem menggunakan *class diagram*. *Class diagram* menggambarkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Rancangan *Class diagram* dapat dilihat pada Gambar 4.





Gambar 4. Rancangan Class Diagram

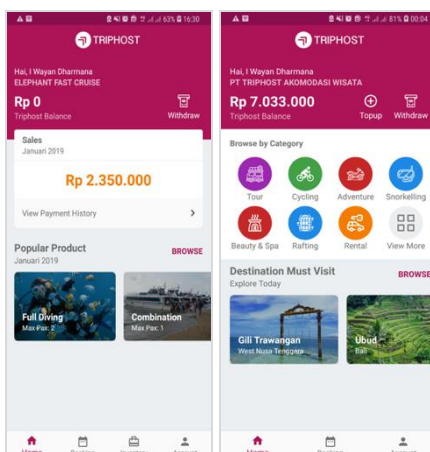
Pada gambar diatas terlihat bahwa aplikasi ini memiliki beberapa class yaitu class *BookingControl*, *InventoryControl*, *PaymentControl*, *StockControl*, serta *NotificationControl*. Class *BookingControl* berelasi dengan *InventoryControl*, *PaymentControl* dan *NotificationControl*. Class *InventoryControl* sendiri berelasi dengan class *StockControl*.

Selanjutnya, dari tahap perancangan, aplikasi dikembangkan dengan bahasa pemrograman Kotlin dan dihasilkan sebuah aplikasi B2B yang memiliki beberapa fitur sebagai berikut :

1. Halaman utama *merchant*

Fitur ini berfungsi untuk menampilkan informasi pengguna, jumlah penjualan, produk populer, serta saldo. Halaman utama *agent* meliputi kategori, informasi saldo, opsi *topup*, opsi *withdraw*, serta daftar destinasi populer. Hasil implementasi halaman utama dapat dilihat pada Gambar 5.

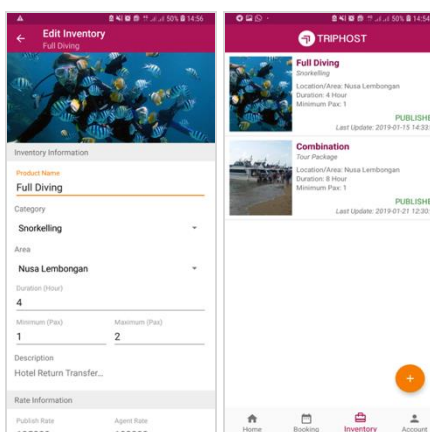




Gambar 5. Halaman Utama *Merchant* (kiri) dan *Agent* (kanan)

## 2. Kelola Jasa & Inventory

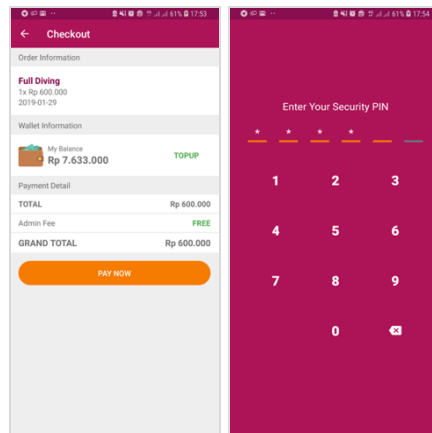
Fitur ini digunakan untuk melakukan penambahan atau update data inventori yang dimiliki *merchant*. Setiap inventori memiliki stok yang dapat diupdate sesuai dengan tanggal yang dipilih. Hasil implementasi kelola jasa / inventori dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Daftar Inventori dan Form Inventori

## 3. Transaksi Reservasi

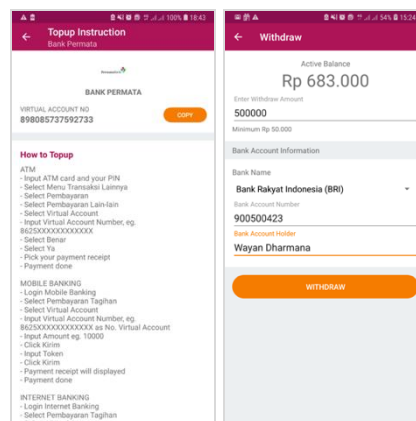
Transaksi reservasi dilakukan oleh *agent* dari pemilihan produk, pengisian form pemesanan, daftar tamu, proses *checkout* dan pembayaran dengan saldo. Hasil implementasi transaksi reservasi dapat dilihat pada Gambar 7.



Gambar 7. Tampilan *Checkout* dan Verifikasi PIN

#### 4. Manajemen Dana

Manajemen dana meliputi lingkup transaksi *topup* dana, *withdraw* dana serta riwayat pembayaran. Hasil implementasi manajemen dana dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Instruksi Pembayaran dan *Withdraw*

Dalam penelitian ini terdapat proses pengumpulan data menggunakan kuesioner. Kuesioner ini dilakukan dalam proses pengujian pengguna untuk mendapatkan data mengenai kemudahan penggunaan aplikasi android *business-to-business marketplace* jasa pariwisata. Proses pengumpulan data menggunakan kuesioner yang memiliki 25 pernyataan yang dibagi menjadi bagian kemudahan pengguna sisi agen dan *merchant*. Jumlah responden



berjumlah 12 orang dengan kategori penilaian meliputi: Sangat Setuju (SS), Setuju (S), Netral / Ragu-Ragu (N), Tidak Setuju (TS) serta Sangat Tidak Setuju (STS). Setelah kuesioner diberikan kepada responden, data kuesioner diolah untuk mendapatkan hasil penilaian kemudahan penggunaan aplikasi *business-to-business marketplace* jasa pariwisata. Adapun hasil pengujian pengguna berupa kuesioner dijabarkan sesuai pernyataan, kategori penilaian dan jawaban responden dapat dilihat pada Tabel 9 :

Tabel 9. Rekap Kuisisioner Pengguna

No	Pernyataan	STS	TS	N	S	SS
<b>A. Kemudahan Penggunaan Sisi Agen</b>						
1	Halaman login sebagai agen mudah digunakan	-	-	-	4	8
2	Halaman registrasi sebagai agen mudah digunakan	-	-	-	10	2
3	Tampilan Halaman utama agen mudah digunakan	-	-	-	10	2
4	Informasi yang disajikan dalam halaman utama agen mudah saya mengerti	-	-	1	7	4
5	Navigasi utama halaman utama agen mudah digunakan	-	-	-	3	9
6	Fitur <i>topup</i> saldo agen mudah digunakan	-	-	-	5	7
7	Halaman <i>withdraw</i> / penarikan dana mudah digunakan	-	-	1	8	3
8	Halaman daftar produk jasa pariwisata dalam proses transaksi memuat informasi yang mudah dipahami	-	-	1	3	8
9	Form pemesanan produk jasa pariwisata mudah digunakan	-	-	-	6	6
10	Form daftar tamu dalam proses transaksi mudah digunakan	-	-	-	7	5
11	Tampilan halaman <i>checkout</i> menyajikan informasi yang mudah dimengerti	-	-	-	1	11
12	Verifikasi PIN mudah digunakan	-	-	-	3	9
13	Halaman daftar reservasi sisi agen mudah digunakan	-	-	-	7	5
14	Halaman riwayat pembayaran mudah digunakan	-	-	-	3	9
<b>B. Kemudahan Penggunaan Sisi Merchant</b>						
1	Halaman login sebagai <i>merchant</i> mudah digunakan	-	-	-	5	7
2	Halaman registrasi sebagai <i>merchant</i> mudah digunakan	-	-	1	9	2
3	Tampilan halaman utama <i>merchant</i> mudah digunakan	-	-	-	7	5
4	Informasi yang disajikan dalam halaman utama <i>merchant</i> mudah saya mengerti	-	-	-	4	8
5	Navigasi utama halaman utama <i>merchant</i> mudah digunakan	-	-	-	6	6
6	Tampilan halaman <i>withdraw</i> / penarikan dana mudah digunakan	-	-	1	7	4
7	Halaman daftar produk jasa pariwisata mudah digunakan	-	-	-	6	6
8	Halaman form tambah / edit produk jasa pariwisata mudah digunakan	-	-	1	7	4
9	Halaman form kelola stok produk jasa pariwisata mudah digunakan	-	-	-	6	6
10	Halaman daftar reservasi sisi <i>merchant</i> mudah digunakan	-	-	-	5	7
11	Halaman riwayat pembayaran mudah digunakan	-	-	-	1	11

Dari hasil yang diberikan oleh responden terlihat bahwa sistem yang dibangun memiliki kemudahan bagi merchant maupun agent dalam menjalankan proses bisnisnya.

### Kesimpulan

Kesimpulan yang didapat dari penelitian aplikasi android *business-to-business marketplace* jasa pariwisata menggunakan Kotlin ini adalah sebagai berikut :

1. Aplikasi Android *Business-to-Business Marketplace* Jasa Pariwisata dapat membantu proses bisnis dalam penyediaan jasa pariwisata.





2. Hasil pengumpulan data menggunakan kuesioner menyimpulkan bahwa sebagian besar responden menyatakan aplikasi yang dibangun mudah digunakan.
3. Aplikasi *business-to-business* (B2B) *marketplace* dapat dikembangkan ke arah *business-to-customer* (B2C) untuk menyasar wisatawan langsung.
4. Aplikasi dapat dikembangkan dengan menyediakan opsi *refundable* dalam setiap transaksi.

#### DAFTAR PUSTAKA

- A. Hendini, "Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)", Jurnal Khatulistiwa Informatika, vol.4, no.2, December 2016.
- Albertus J. Strategi Business to Business (B2B) Dalam Manajemen Bisnis Perusahaan. Kajian Ilmiah Mahasiswa Manajemen. 2012; 1(1): 26-29.
- Jony W. Internet marketing for beginners. Jakarta: Elex Media Komputindo, 2013.
- N. Rohman and R. Toro, Kotlin Android Developer Expert. Bandung: PT. Presentologics, 2018
- S. Sutejo, "Pemodelan UML Sistem Informasi Geografis Pasar Tradisional Kota Pekanbaru", Jurnal Teknologi Informasi dan Komunikasi, vol. 7, no.2, pp. 89-99. November 2016.

