

# PENERAPAN TEKNOLOGI RESTFUL WEB SERVICE APLIKASI MOBILE JASA MAKE UP

## IMPLEMENTATION OF RESTFUL WEB SERVICE TECHNOLOGY MAKE UP SERVICE MOBILE APPLICATION

\*<sup>1</sup>Vivien Kay Gow, <sup>2</sup>Sandy Kosasi, <sup>3</sup>Tony Wijaya, <sup>4</sup>David, <sup>5</sup>Robertus Laipaka

<sup>1</sup>Mahasiswa STMIK Pontianak, <sup>2,3,4,5</sup>Dosen STMIK Pontianak

Jalan Merdeka No. 372 Pontianak, 0561-735555

e-mail: \*<sup>1</sup>vien.go999@gmail.com, sandykosasi@stmikpontianak.ac.id,

mail.tonywijaya@stmikpontianak.ac.id, david@stmikpontianak.ac.id,

robertus.laipaka@stmikpontianak.ac.id

### Abstrak

*Proses pemesanan jasa make up di Pontianak masih menggunakan aplikasi sosial media. Kendala yang sering dihadapi adalah ketika pelanggan ingin memesan, maka pelanggan masih harus menanyakan jenis dan harganya. Pelanggan juga harus menyesuaikan jadwal agar tidak terjadi pembentrokan jadwal dan masih harus menunggu konfirmasi. Pemilik jasa make up juga kesusahan untuk melihat laporan pemesanan dalam kurun waktu tertentu, sehingga penulis berinisiatif untuk membuat aplikasi mobile untuk pemesanan jasa make up. Aplikasi mobile ini dibuat dengan menggunakan bahasa pemrograman XML dan java dengan tools android studio versi 4.1.3. Bentuk penelitian adalah survey, metode penelitian menggunakan action research, metode pengembangan sistem menggunakan XP dan pemodelan sistem menggunakan UML. Aplikasi mobile yang dibuat untuk admin dan pelanggan saling terintegrasi dengan restful web service. Hasil akhir dari penelitian ini adalah aplikasi mobile pemesanan jasa make up. Aplikasi mobile ini dapat digunakan untuk menambahkan jenis jasa make up, juga dapat untuk melihat, menghapus, dan mengubah. Pelanggan juga dapat menggunakan aplikasi untuk melakukan pemesanan, pemesanan yang dilakukan akan masuk ke keranjang belanja dan pelanggan dapat melakukan check out dan mengupload bukti pembayaran berupa gambar. Admin dapat melihat pesanan serta dapat melihat history laporan pesanan. Pengujian sistem menggunakan metode black box testing dengan teknik graph-based. Metode pengujian black box testing dijadikan pilihan dikarekan tahap pengujian pada metode extreme programming adalah menguji fungsi aplikasi agar sesuai dengan kebutuhan pengguna dalam hal ini adalah admin dan pelanggannya.*

**Kata kunci:** *Android studio, restful, XP, UML.*

### Abstract

*The process for ordering a Make Up Artist in Pontianak is still using a social media. The problem that is often faced by customer is when the customer wants to order, they still have to ask the type and price. Customers also have to adjust the schedule to avoid the schedule conflicts and still have to wait for the futher confirmation. Admin also have the difficulty to seeing the order reports within a certain time, so by the problems author took the initiative to build a mobile application for ordering MUA. The application is made by using the XML and Java programming languages with android studio version 4.1.3. The research form is survey, the research method is using action research, system development method is using XP, and system modelling tools is using UML. The mobile application that made for admin and user are being integreted using restful web service. The final result is a mobile aplication for ordering MUA that can be used to add the types of makeup, also to view, delete, and change it. The customer can also use the aplication to order it, the order will add to the shopping cart and can be check put also can upload the payment image. Admin can see the orders list and can view the orders report history. System testing is using black box testing method with graph-based techniques that was chosen because the testing phase in XP method is to test the functions to suit the needs of user.*

**Keywords :** *Android studio, restful, XP, UML*

## 1. PENDAHULUAN

Perkembangan teknologi informasi khususnya pada perangkat *mobile* telah banyak dipergunakan oleh masyarakat untuk mengakses informasi dengan mudah dan cepat. Pemilihan menggunakan *mobile* daripada *website* juga karena sekarang layanan yang tersedia dalam *mobile* telah memungkinkan komunikasi, kolaborasi dan koordinasi melalui perangkat *mobile*. Aplikasi *mobile* juga memungkinkan pengguna untuk menggunakan teknologi informasi tanpa terikat ke satu lokasi, menyediakan pengguna dengan fleksibilitas, aksesibilitas dan kemudahan penggunaan telah menjadikan perangkat *mobile* sebagai pilihan dalam mengakses informasi [1].

Saat ini penggunaan perangkat *mobile* tidak hanya digunakan untuk memenuhi kebutuhan sehari-hari namun juga digunakan dalam bidang bisnis salah satunya adalah bisnis jasa *make up* seperti pada *make up artist* (MUA). MUA di Pontianak masih menggunakan beberapa aplikasi media sosial untuk bertransaksi dan melakukan promosi [2]. Promosi dengan media sosial masih kurang efektif karena informasi jasa yang ditampilkan masih kurang lengkap. Masyarakat juga masih mengalami kendala lain dalam mengatur jadwal pertemuan di mana konsumen harus mengirim pesan ke MUA untuk memberitahu jadwal yang diinginkan dan menunggu konfirmasi jadwal dari MUA, MUA kesulitan untuk mengetahui pelanggan mana yang sering memesan dan sudah berapa kali memesan jasa di tempatnya serta jenis jasa *make up* mana yang sering dipesan, kendala bagi konsumen ketika telah memesan salah satu jenis paket *make up* dan ingin melakukan perubahan pemesanan konsumen masih harus mengirim pesan kembali ke MUA dan menunggu balasan atau konfirmasi dari MUA nya. Permasalahan lain bagi MUA adalah MUA harus menanyakan kembali nama, alamat lengkap, dan pesanan yang dipilih pemesan. Kendala lain yang dihadapi adalah laporan mengenai transaksi tidak terekap secara lengkap sehingga MUA kesulitan untuk mengetahui berapa banyak pesanan dan penghasilan yang didapat dalam kurun waktu tertentu, serta pembayaran masih dilakukan ketika konsumen dan MUA bertemu atau konsumen mengirim bukti transfer ke media sosial MUA dan konsumen masih harus menunggu konfirmasi penerimaan bukti pembayaran dari MUA. Sehingga dengan adanya kendala yang telah dipaparkan memicu penulis untuk menghasilkan aplikasi *mobile* pemesanan jasa *make up*.

Aplikasi *mobile* jasa *make up* ini dirancang agar memberi kemudahan bagi konsumen, karena konsumen dapat melihat jenis *make up* yang disediakan sekaligus dapat langsung memesan dengan cara memilih jenis jasa *make up* yang diinginkan dari toko jasa *make up* yang tersedia di aplikasi dan konsumen dapat langsung memasukkan tanggal waktu serta alamatnya sehingga pemesanan akan lebih sistematis dan teratur, setelah itu konsumen dapat langsung *check out* pesannya dan mengupload gambar bukti pembayaran yang telah dilakukan.

Aplikasi *mobile* pemesanan jasa *make up* ini dibuat dengan menggunakan IDE (*Integrated Development Environment*) android studio 4.1.3 dan menggunakan database MariaDB yang merupakan database eksternal dan bukan database bawaan dari android seperti SQLite sehingga memerlukan *web service* sebagai penghubung antara sistemnya [3]. *Web Service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu website untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan oleh suatu sistem yang menyediakan *web service* [4]. *Jenis web service yang digunakan adalah Restful*. *Restful* mewakili arsitektur *client-server* di mana *client* mengirimkan permintaan, sementara *server* memprosesnya dan mengembalikan respons [5]. *Restful web service* memiliki arsitektur yang mencakup setiap permintaan dari *client* ke *server* harus berisi semua informasi yang diperlukan untuk memahami permintaan tersebut, *restful web service* memiliki antarmuka yang seragam dan batasan ini biasanya diartikan bahwa satu-satunya operasi yang diizinkan adalah operasi HTTP seperti *get*, *post*, *put*, dan *delete*, arsitektur berbasis REST dibangun dari potongan informasi yang diidentifikasi secara unik oleh URI (*Uniform Resource Identifier*) [6]. Gagasan utama dari REST adalah konsep *resource* sebagai komponen dari aplikasi yang perlu digunakan atau dialamatkan. REST membangun integrasi dengan cara lebih ringan dan sederhana, serta berfokus pada sumber daya [7]. Penggunaan *restful* atau API (*Application Programming Interface*) di aplikasi *mobile* memerlukan converter Gson untuk mengkonversikan file JSON (*JavaScript Object Notation*) karena android studio tidak bisa membaca file langsung dari database, harus dikonversikan terlebih dahulu dengan format JSON [8]. Salah satu *response* yang dapat dan sering digunakan dalam sebuah rest adalah JSON, karena

JSON merupakan format *response* yang berukuran kecil yang mudah dibaca dan ditulis oleh manusia, dan juga mudah ditulis dan diurai oleh mesin [9]. Aplikasi *client* android yang dibangun terintegrasi oleh API yang menggunakan metode REST. Data database yang direques oleh *client* nantinya akan digabungkan dan dikirim dalam bentuk JSON. Kemudian data tersebut akan dikirim ke server API. Server API akan melakukan proses permintaan ke database yang hasil dari proses permintaan akan dikembalikan lagi ke pengguna dalam bentuk JSON dan diuraikan oleh aplikasi sehingga data hasil proses bisa digunakan dan ditampilkan [10].

Penelitian terdahulu yang relevan adalah aplikasi makeup 4u. Penelitian ini dibangun berdasarkan survey yang sudah dilakukan terhadap beberapa pemilik jasa make up dan pelanggannya. Fitur-fitur dalam aplikasi mobile ini adalah penjadwalan, filter pencarian, notifikasi. Aplikasi ini dibuat menggunakan framework phonegap dan database yang digunakan adalah MySQL. Sebelum masuk ke aplikasi, pengguna diharuskan melakukan registrasi dan akun akan diverifikasi melalui email yang pengguna masukan.

Penelitian terdahulu ini digunakan sebagai acuan untuk penelitian ini di mana pada penelitian terdahulu menggunakan framework *phonegap* sedangkan pada penelitian ini tidak menggunakan framework karena aplikasi *mobile* langsung dibuat menggunakan android studio yang di mana layout untuk *interface* aplikasi langsung dibuat pada xml dan java activity yang terdapat pada android studio. Penelitian terdahulu juga menggunakan database mysql sedangkan pada penelitian ini menggunakan database MariaDB yang telah disediakan pada web server yang digunakan yaitu 000webhost dan menggunakan ekstensi mysql untuk menghubungkan antara database dengan API yang dibuat. Penelitian ini juga menerapkan fitur-fitur yang terdapat pada penelitian terdahulu seperti fitur registrasi akun dan log in akun, dan fitur penjadwalan agar tidak terjadi pembentokan setiap pelanggan memesan jasa make up yang diinginkan, karna ketika tanggal dan waktu yang dipilih sudah pernah dipilih oleh pelanggan sebelumnya maka akan muncul pemberitahuan untuk memilih ulang tanggal dan waktunya. Penulis juga melakukan pembaharuan seperti penambahan fitur-fitur agar lebih mempermudah penggunaan untuk admin pemilik jasa make up dan pelanggannya, seperti fitur untuk melihat, menambah, mengubah, dan menghapus jasa make up, fitur pembukuan atau *history* pesanan sehingga admin dapat melihat laporan pesanan yang telah terjadi, dan fitur untuk *check-out* pesanan yang dapat dilakukan pelanggan ketika sudah merasa yakin dengan pesanan yang dipilihnya serta fitur untuk mengupload bukti pembayaran yang telah dilakukan pelanggan.

## 2. METODE PENELITIAN

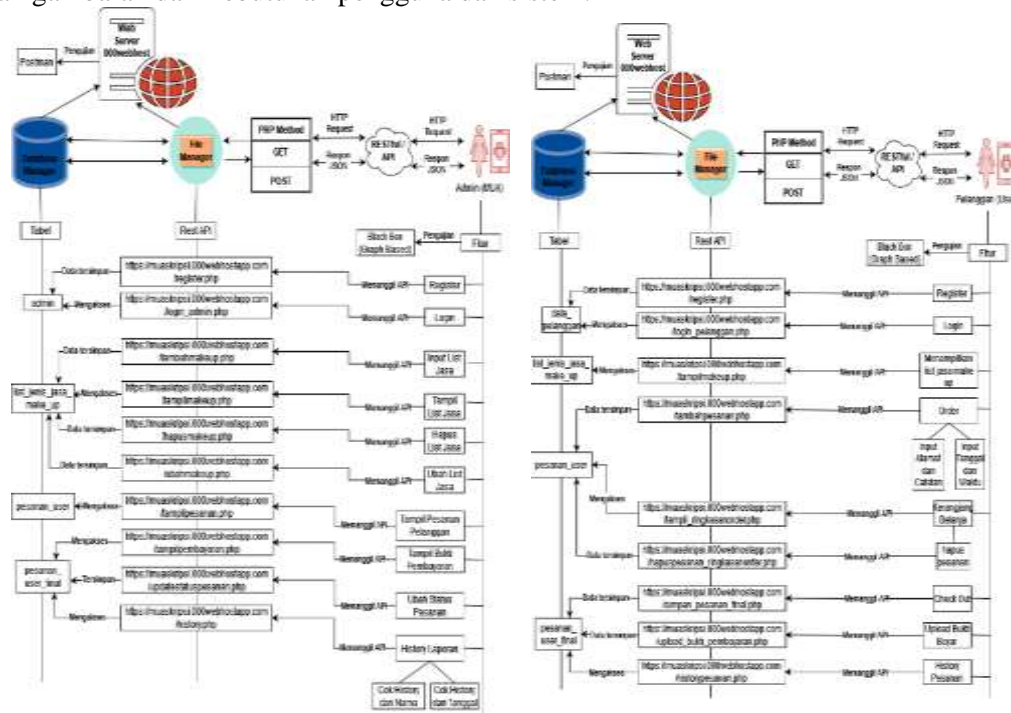
Penelitian ini menggunakan bentuk pemodelan survei dengan metode penelitian *action research* dan metode pengembangan sistem *extreme programming*. Bentuk penelitian survei menyediakan pertanyaan untuk penelitian tentang laporan keyakinan dan perilaku diri. Beberapa pertanyaan dapat disertakan pada suatu survei, yaitu berkenaan dengan perilaku, sikap, pendapat, keyakinan, karakteristik, ekspektasi, pengklasifikasian, dan pengetahuan [11]. Metode penelitian *action research* memiliki lima tahap, yaitu *diagnosing*, *action planning*, *action taking*, *evaluating*, dan *learning* [12]. *Action research* dapat mengatasi masalah saat bersamaan dan merupakan metode yang didasarkan pada tindakan masyarakat yang seringkali diselenggarakan pada suatu latar yang luas seperti di sekolah, rumah sakit, dan pabrik [13]. *Action research* dipilih karena kelebihan dan hubungan yang dimiliki antara metode *action research* dengan pembangunan aplikasi ini, yang di mana dengan *action research* penulis dapat berkolaborasi dengan admin dan user untuk mencapai kepuasan membangun aplikasi sesuai dengan kebutuhan admin dan user. Metode *extreme programming* (XP) terdiri dari *planning*, *design*, *coding* dan *testing*. Pemilihan metode XP dikarenakan dapat meningkatkan kepuasan kepada user, pembangunan sistem dibuat lebih cepat, dan menjalin komunikasi yang baik dengan *client* [14].

Pemodelan sistem menggunakan UML (*Unified Modeling Language*). Teknik pengumpulan data melalui teknik wawancara dan observasi kepada admin pemilik jasa make up atau MUA dan pelanggannya. Metode pengujian perangkat lunak yang digunakan adalah *black box testing* dengan teknik *graph based*. Pada *black box testing* dilakukan pengujian yang

didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh konsumen. *Black box testing* ini lebih menguji ke *interface* dari suatu aplikasi supaya mudah digunakan oleh *konsumen*. Pengujian ini tidak melihat dan menguji *source code* program [15]. Pengujian *black-box testing* akan diterapkan untuk menguji 2 (dua) bagian dari aplikasi yang akan dibangun pada penelitian ini yaitu aplikasi pelanggan dan aplikasi admin.

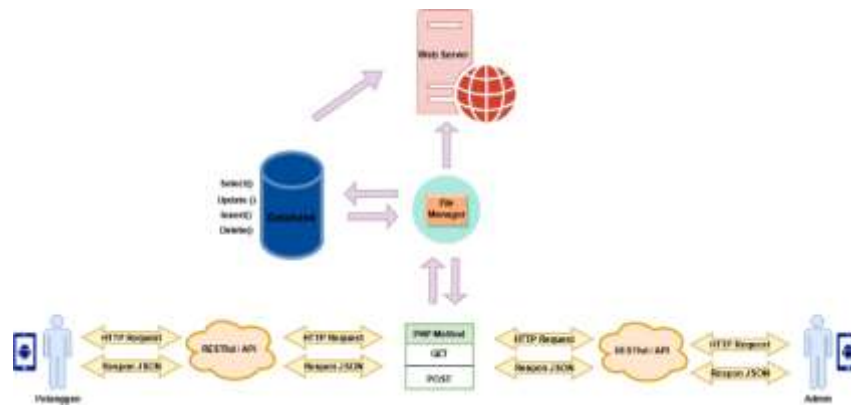
### 3. HASIL DAN PEMBAHASAN

Dalam merancang aplikasi ini digunakan metode pengembangan sistem *extreme programming*. Berdasarkan tahapan-tahapan yang ada pada metode XP, maka penulis membangun aplikasi seccara bertahap mengacu pada Langkah-langkah metode XP. Diawali dengan melakukan perencanaan atau *planning* berupa identifikasi kebutuhan pengguna dan analisis kebutuhan sistem. Tahap identifikasi kebutuhan pengguna dilakukan dengan cara observasi, wawancara seperti yang telah dijelaskan pada teknik pengumpulan data. Analisis kebutuhan sistem dilakukan dengan cara mempelajari masalah-masalah yang terjadi pada pemesanan jasa make up pada admin. Kemudian membangun sebuah sistem berbasis *android* untuk membantu untuk membantu menyelesaikan masalah-masalah yang ada. Tahap selanjutnya adalah *design*. Tahapan *design* dilakukan dengan menggambarkan arsitektur sistem dan dimodelkan dalam bentuk diagram UML. Rancangan sistem untuk aplikasi dijelaskan melalui *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity-relationship diagram*. Perancangan arsitektur sistem ini dimaksudkan untuk memberikan gambaran dari kebutuhan pengguna dan sistem.



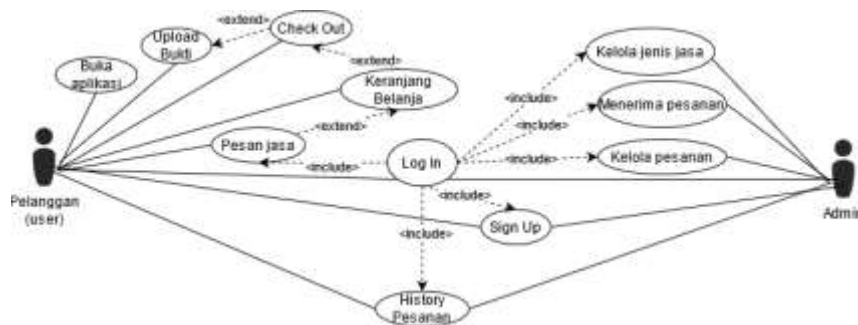
Gambar 1. Arsitektur Sistem pada Admin dan User

Gambar 1 merupakan arsitektur pengambilan data menggunakan *RESTful web service* oleh pelanggan dan admin. Bila pelanggan dan admin akan melakukan permintaan atau *request* ke server, maka data tersebut akan dikirim ke *web server* yang dibangun. *REST API* akan melakukan proses permintaan ke *database* yang kemudian hasil dari proses permintaan atau *request* akan dikembalikan lagi ke pengguna dalam bentuk *JSON* dan diuraikan oleh aplikasi sehingga data hasil proses bisa digunakan dan ditampilkan dan dapat dilihat dari semua fitur yang tersedia pada aplikasi admin dan pelanggan akan mengakses masing-masing API.



Gambar 2. Skema Sistem

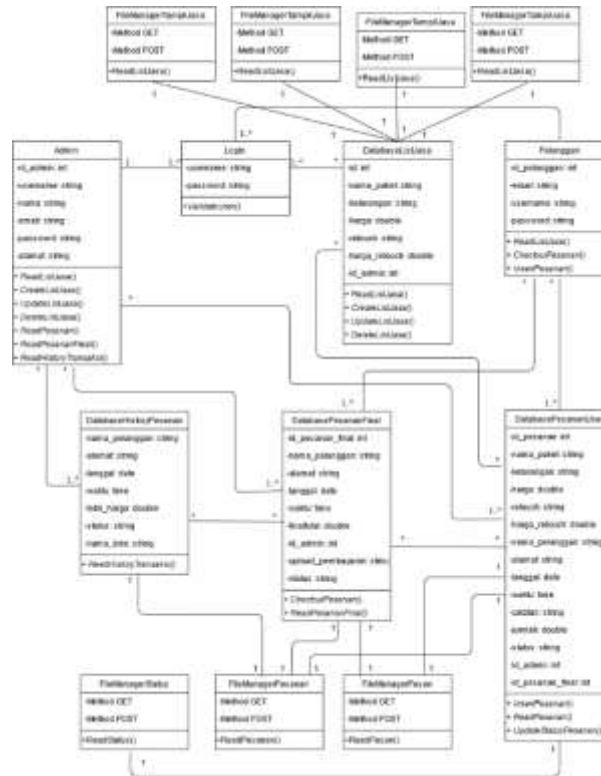
Gambar 2 merupakan gambaran dari sistem yang dibangun. Aplikasi *mobile* pelanggan dan admin akan menampung API. API dibuat dengan bahasa pemrograman PHP (*Hypertext Processor*) yang di-*hosting* ke dalam web server. Di dalam web server menampung file manager dan database yang digunakan. File manager berisi kumpulan file PHP yang akan digunakan sebagai API dan database berisi tabel-tabel untuk menampung data yang diperlukan. Data dalam database akan diakses menggunakan file-file PHP yang ada pada file manager sehingga data yang ada pada database dapat dikonversi ke dalam bentuk JSON agar dapat ditampilkan pada aplikasi *mobile* android. Penjelasan detail terdapat pada tabel 1 dan tabel 2.



Gambar 3. Use Case Aplikasi Mobile Pemesanan Jasa Make Up

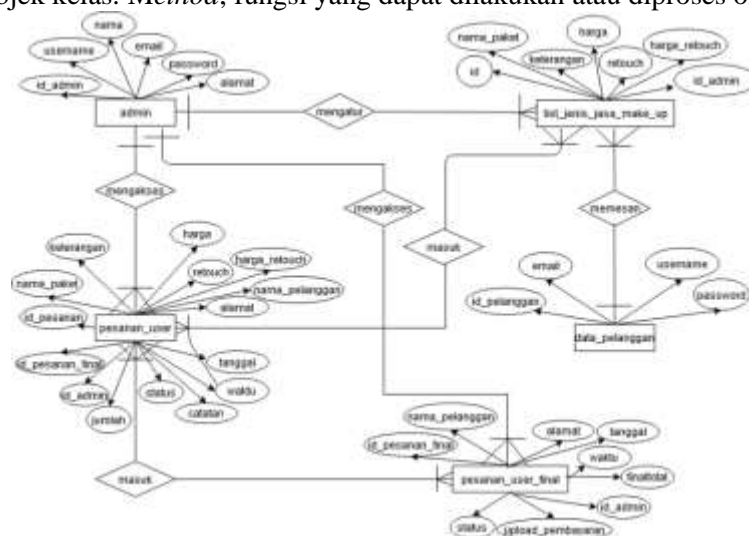
Gambar 3 menjelaskan *use case* sistem, di mana aktor dalam use case adalah admin dan pelanggan. Pelanggan dapat melakukan pemesanan setelah membuka dan men-*login* aplikasi dan setelah pelanggan melakukan pemesanan maka pesanan yang telah dipilih akan masuk kedalam keranjang belanja dan ketika pelanggan telah yakin dengan pesanan yang dipilih nya maka pelanggan dapat meng-*check out* pesanan tersebut dan mengupload bukti pembayaran berupa gambar. Admin juga perlu log in ke aplikasi untuk mengelola jenis jasa make up yang tersedia dan mengelola status pesanan pelanggan, status yang di maksud adalah status sudah atau belum nya pesanan jasa make up pelanggan terlaksanakan. Ketika admin telah mendatangi alamat yang telah diinputkan oleh pelanggan untuk melakukan jasa make up yang telah dipesan pelanggan, maka admin akan mengubah status pesanan. Pelanggan dapat melihat *history* pesanan di mana *history* pesanan yang dimaksud adalah semua pesanan yang telah berstatus selesai dan admin juga dapat melihat *history* pesanan, jadi ketika admin memilih tanggal tertentu maka semua laporan pesanan pada kurun tanggal yang telah ditentukan akan muncul sehingga admin dapat melihat berapa pemasukan dalam kurun tanggal tersebut serta admin dapat mencari nama pelanggan yang diinginkan maka semua laporan pesanan atas nama pelanggan tersebut akan ditampilkan sehingga admin dapat mengetahui berapa kali pelanggan tersebut memesan jasa nya dan jasa apa yang paling sering dipesan.





Gambar 6. Class Diagram

Gambar 6 merupakan *class diagram* untuk menunjukkan struktur fisik dan aplikasi yang dikembangkan oleh penulis. Terdapat 13 (tiga belas) *class* diagram yang digunakan oleh aplikasi ini yaitu *class Admin*, *class Login*, *class ListJasa*, *class Pelanggan*, *class Pesanan User Final*, dan *class History Pesanan*, *class File Manager Tampil Jasa*, *class File Manager Tambah Jasa*, *class File Manager Ubah Jasa*, *class File Manager Hapus Jasa*, *class File Manager Pesan*, *class File Manager Pesanan*, *class File Manager Status*. *Class* diagram diatas terdiri dari *class name*, merupakan sesuatu yang mewakili dari nama kelas. atribut, merupakan properti dari sebuah kelas, atribut melambangkan batas nilai kelas yang mungkin terdapat dalam objek kelas. *Method*, fungsi yang dapat dilakukan atau diproses oleh sebuah kelas.



Gambar 7. ERD Aplikasi Mobile Pemesanan Jasa Make Up

Gambar 7 merupakan *Entity relationship diagram* (ERD) menjelaskan tentang hubungan antar entitas yang saling berkaitan satu dengan yang lainnya. Setiap entitas memiliki beberapa atribut masing-masing. Ketika pelanggan (entitas *data\_pelanggan*) memesan produk (entitas

ist\_jenis\_jasa\_make\_up) akan masuk ke tabel pesanan (entitas pesanan\_user) baik itu data pelanggan maupun data produk yang dipesan oleh pelanggan. Kemudian admin (entitas admin) dapat mengakses pesanan pelanggan (entitas pesanan\_user) untuk melihat dan mengubah status pesanan yang telah selesai.

Tahap ketiga adalah *coding* yang merupakan tahap implementasi planning dan design yang telah dibuat yang di mana terdapat dua bagian yaitu bagian *front end* yang merupakan bagian interface antara user dan admin karena akan menampilkan fitur – fitur dalam aplikasi mobile dan *back end* yang berisi database yang akan menampung data - data yang akan diakses oleh aplikasi mobile ini. Pada bagian *front end* ada beberapa bagian yaitu *interface* untuk admin melakukan penambahan, pengubahan, penghapusan, dan menampilkan jenis jasa make up apa saja yang tersedia, interface untuk admin melihat ringkasan order pelanggan dan mengubah status pesanan pelanggan, interface untuk *user* (pelanggan) melihat dan memesan list jasa make up yang tersedia dan , dan interface untuk *user* melihat ringkasan pesanan yang telah dipesan.



Gambar 8. Layout Menu untuk User

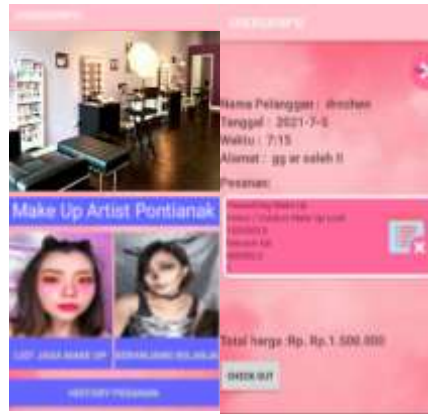
Gambar 8 merupakan layout dashboard untuk user yang di mana terdapat tiga fitur icon button yaitu icon button list jasa make up yang akan menampilkan jenis – jenis make up yang tersedia, icon button keranjang belanja untuk user melihat rincian pesanan yang telah dilakukan, dan icon history pesanan untuk melihat laporan pesanan yang telah terjadi.



Gambar 9. Layout List Jasa Make Up untuk User

Seperti pada gambar 9 ketika user mengklik icon list jasa make up maka akan berpindah halaman untuk user melakukan pemesanan dan juga yang menampilkan jenis-jenis make up apa saja yang tersedia dan di dalam halaman tersebut juga terdapat beberapa fitur seperti icon untuk memilih tanggal, memilih waktu, icon untuk memilih jumlah pesanan, dan icon untuk melakukan order. User dapat memilih langsung tanggal dan waktu nya yang di mana jika tanggal dan waktu yang dimasukan sama maka user tidak dapat melakukan order dan harus mengubah tanggal atau waktu nya dan jika tanggal dan waktu yang dipilih user masih belum pernah dipilih oleh user lain maka user akan berhasil melakukan orderan.





Gambar 10. Layout Keranjang Belanja pada User

Seperti pada gambar 10 ketika user mengklik icon keranjang belanja maka akan berpindah ke halaman yang mencampilkkan rincian orderan yang telah dilakukan oleh user dan ketika user sudah yakin dengan pesanan yang dipilih maka user dapat mengklik icon check out.



Gambar 11. Layout Upload Bukti Pembayaran pada User

Seperti pada gambar 11 ketika user telah mengklik icon check out maka akan masuk ke halaman untuk mengupload bukti pembayaran, user dapat memilih foto bukti pembayaran dan mengupload nya dan user icon memilih icon upload maka akan muncul pemberitahuan berhasil diupload yang menandakan bukti pembayaran telah terupload dan masuk ke database pesanan final user.



Gambar 12. Layout History Pesanan pada Admin

Seperti pada gambar 12 ketika admin mengklik icon (tampil history pesanan) maka akan membuka browser yang terdapat di hp (admin) dan menampilkan form laporan penjualan. Admin harus memilih tanggal mulai dan akhir yang diinginkan setelah itu memilih button (cek transaksi) maka akan menampilkan penjualan yang telah terjadi selama kurun waktu yang telah dimasukkan

sebelumnya dan admin juga dapat mengetahui berapa banyak penghasilan yang telah di dapat, siapa pelanggan yang sering melakukan pesanan dan jenis jasa make up apa yang sering dipesan oleh pelanggan. Admin juga dapat memasukkan nama pelanggan yang ingin diketahui dan memilih button (cek transaksi) maka admin dapat mengetahui berapa kali pelanggan tersebut memakai jasa make up nya dan jenis jasa make up mana yang sering digunakan oleh pelanggan tersebut.

Berikut ini merupakan beberapa contoh *rest api / restful web service* yang digunakan dalam penelitian ini yang ditulis menggunakan bahasa pemrograman php (*Hypertext Preprocessor*). Terdapat 19 *restful web servie* yang digunakan untuk mengelola setiap fitur pada sistem yang dibangun, namun akan ditampilkan salah satu contoh saja.

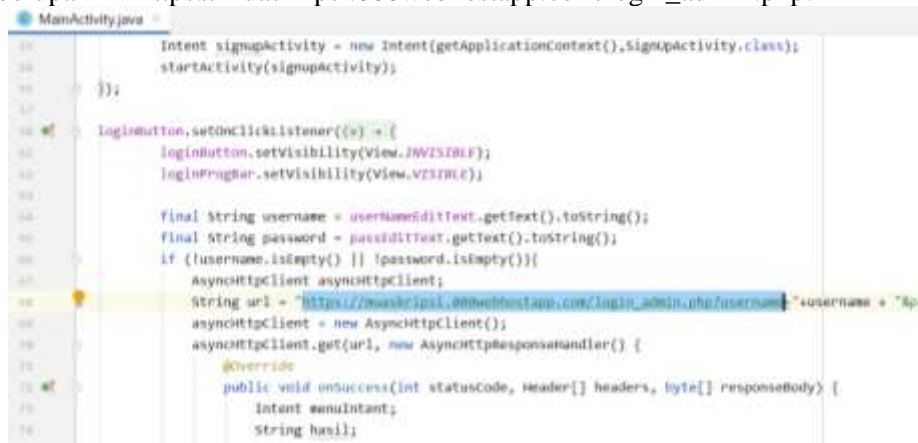
```

/public_html/login_admin.php
<?php

require 'conn.php';
$username = $_GET['username'];
$password = $_GET['password'];
$sql_get_user = "SELECT * FROM admin where username='$username' AND password='$password'";
$query= $conn->query($sql_get_user);
$result=mysqli_num_rows($query);
$idCount = null;
if($result==1){
    $row= $query-> fetch_array();
    $idCount[] = (array("idCount"=>"1", "id_admin"=>$row["id_admin"]));
} else{
    $idCount[] = (array("idCount"=>"0"));
}

header('Content-Type: application/json');
$response = $idCount;
echo json_encode($response);
?>
    
```

File yang memuat *script* ini berfungsi untuk mengecek kecocokan antara username dan password yang diinputkan admin dan data yang ada pada *database*. Kemudian menggunakan *php array* untuk hasil *result*, jika data sama maka *idCount*1 dan jika berbeda maka *idCount*0. Ditampilkan menggunakan *json\_encode*. Ketika file php di-view maka akan menampilkan link yang berupa API [https://muaskripsi.000webhostapp.com/login\\_admin.php](https://muaskripsi.000webhostapp.com/login_admin.php).



Gambar 13. API yang Digunakan untuk Fitur Login

Seperti pada gambar 13 API tersebut akan digunakan pada file java dalam android studio, jadi ketika admin melakukan *request* pada sistem (aplikasi mobile) maka sistem akan memanggil API yang telah dimasukkan tadi dan API akan mengakses data ke dalam *database* admin.

Berikut adalah detail dari tiap proses terkait skenario yang dilakukan, file java pada android yang terlibat, file php yang digunakan, url restful pada web server yang digunakan, serta tabel *database* yang digunakan.

Tabel 1. Proses Admin Menggunakan Sistem

No	Android		Web Server File Manager		Web Server Database Manager
	Layout	Java	PHP	Restful	

1.	Sign Up	SignUpActivity.java	registeradmin.php	https://muaskripsi.000webhost.com/registeradmin.php	admin
2.	Log In	MainActivity.java	login_admin.php	https://muaskripsi.000webhost.com/login_admin.php	admin
3.	Tambah, Tampil, Hapus Jasa Make Up	TampilHapus MakeUpArtist Activity.java, MUAdapter.java	tampilmakeup.php, hapusmakeup.php, tambahmakeup.php	https://muaskripsi.000webhost.com/tampilmakeup.php, https://muaskripsi.000webhost.com/hapusmakeup.php, https://muaskripsi.000webhost.com/tambahmakeup.php	list_jenis_jasa_make_up
4.	Tampil Pesanan Pelanggan	TampilPesanan Activity.java	tampilpesanan.php	https://muaskripsi.000webhost.com/tampilpesanan.php	pesanan_user, pesanan_user_final
5.	Ubah Status Pesanan User	PesananAdapter.java	updatestatus pesanan.php	https://muaskripsi.000webhost.com/updatestatus pesanna.php	pesanan_user_final
6.	Tampil Bukti Bayar	TampilPembayaran Activity.java	tampilpembayaran .php	https://muaskripsi.000webhost.com/tampilpembayaran .php	pesanan_user, pesanan_user_final

Tabel 2. Proses Pelanggan Menggunakan Sistem

No	Android		Web Server File Manager		Web Server Database Manager
	Layout	Java	PHP	Restful	
1.	Sign Up	SignUpActivity.java	register.php	https://muaskripsi.000webhost.com/register.php	data_pelanggan
2.	Log In	MainActivity.java	login_pelanggan .php	https://muaskripsi.000webhost.com/login_pelanggan .php	data_pelanggan
3.	Tampil Jenis Jasa	TampilJenisMU Activity.java	tampilmakeup.php	https://muaskripsi.000webhost.com/tampilmakeup.php,	list_jenis_jasa_make_up
4.	Pesan Jasa	MUAdapter.java	tambahpesanan.php	https://muaskripsi.000webhost.com/tambahpesanan.php	pesanan_user
5.	Keranjang Belanja	Keranjang RingkasanOrder Activity.java	tampil_ringkasanorder.php	https://muaskripsi.000webhost.com/tampil_ringkasanorder.php	pesanan_user
6.	Hapus Pesanan di Keranjang Belanja	KeranjangRingkasan Adapter.java	hapuspesanan_ringkasanorder.php	https://muaskripsi.000webhost.com/hapuspesanan_ringkasanorder.php	pesanan_user
7.	Upload Bukti Bayar (gambar)	PesananUser Final.java	upload_bukti_pembayaran.php	https://muaskripsi.000webhost.com/upload_bukti_pembayaran.php	pesanan_user_final
8.	Tampil History Pesanan	HistoryPesanan Activity.java	historypesanan.php	https://muaskripsi.000webhost.com/historypesanan.php	pesanan_user_final

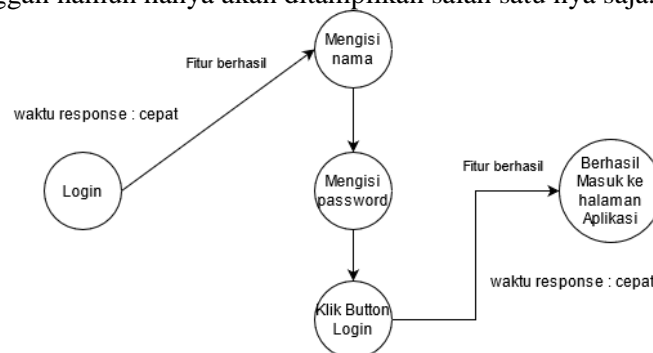
Tahap terakhir dari metode XP adalah melakukan *testing* dari sistem yang dibangun. Penelitian ini menggunakan metode pengujian *black box* dengan teknik *graph based* untuk menguji setiap fungsi dari sistem yang dibangun apakah sudah berjalan sebagaimana mestinya, namun hanya akan ditampilkan punya *user*.

Tabel 3. *Tessting Front End Interface User*

No	Pengujian	Pengujian yang dilakukan	Hasil yang	Hasil Pengujian	Kesimpulan
----	-----------	--------------------------	------------	-----------------	------------

	terhadap		diharapkan		
1	Layout register untuk user	Membuat akun dengan memasukkan username, email, dan password	Berhasil register dan dapat melakukan login	Berhasil membuat akun	Valid
2	Layout login untuk user	Memasukan username dan password yang telah buat dalam layout register	Berhasil log in dan masuk ke layout dashboard user	Berhasil login dan berpindah layout	Valid
3	Pada dashboard user	Mencoba fitur icon list jasa make up	Masuk ke layout untuk melihat jenis jasa make up apa saja yang tersedia dan untuk melakukan orderan	Berhasil berpindah layout dan dapat melihat serta melakukan pengorderan	Valid
4	Pada dashboard user	Mencoba fitur kembali dari layout list jasa make up	Dari layout untuk list jasa make up dapat langsung kembali ke halaman depan (dashboard user)	Tidak dapat langsung kembali ke halaman depan	Tidak Valid
5	Layout Tampil Keranjang Belanja	Mencoba menampilkan pesanan yang akan dipesan oleh pelanggan	Menampilkan semua pesanan yang dipilih dan akan dipesan oleh pelanggan	Berhasil menampilkan semua pesanan yang akan dipesan oleh pelanggan	Valid
6	Layout Check Out dari Keranjang Belanja	Mencoba untuk men-check out pesanan yang akan dipesan dari keranjang belanja	Dapat men-checkout pesanan dan berpindah layout untuk mengupload bukti pembayaran	Berhasil men-check out pesanan	Valid
7	Layout Check Out	Mencoba meng-upload bukti pembayaran	Dapat meng-upload bukti pembayaran	Berhasil meng-upload gambar	Valid
8	Layout History Pesanan	Mencoba menampilkan pesanan yang telah dilakukan oleh user	Dapat menampilkan pesanan	Berhasil menampilkan history pesanan	Valid

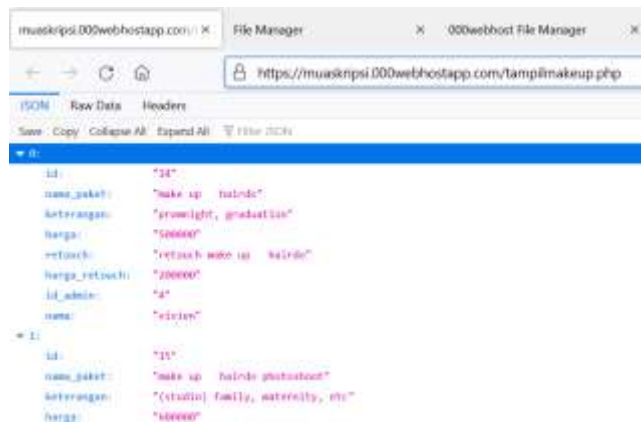
Berikut adalah teknik *graph test* dari metode *black box testing* pada aplikasi *mobile* jasa make up untuk pelanggan namun hanya akan ditampilkan salah satu nya saja.



Gambar 14. Graph Login pada User

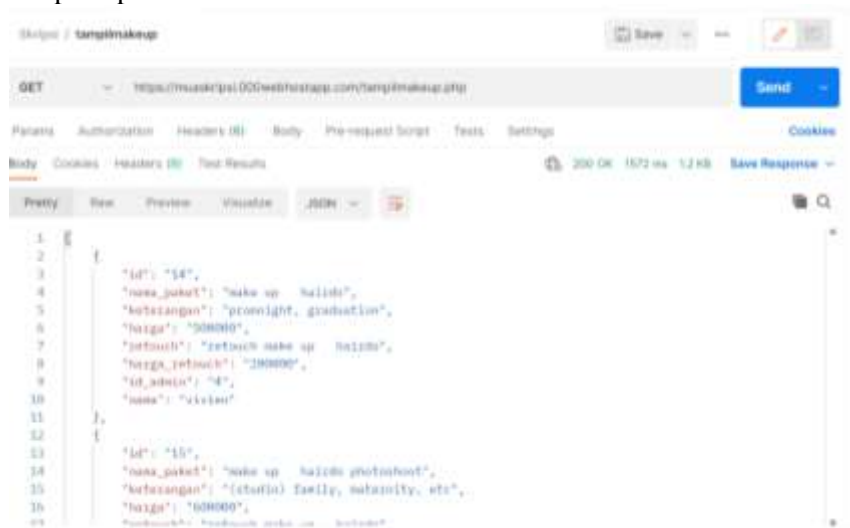
Seperti pada gambar 14 dimasukkan nama yaitu drche dan password yaitu drchen dan berhasil masuk ke halaman utama (dashboard *user*) pada aplikasi.

Adapun untuk pengujian *restful* pada penelitian ini menggunakan aplikasi postman. Pengujian pada postman dapat dilakukan dengan mendownload aplikasi postman kemudian membuat workspace seperti pada gambar 4.86 kemudian setelah itu dapat memilih *add request* kemudian pada bagian *enter request url* dapat dimasukkan link file php yang telah di view, salah satu contoh nya dapat dilihat pada gambar



Gambar 15. REST untuk Postman

Seperti pada gambar 4.87 link yang berupa rest/api dapat disalin dan dipaste pada kolom *enter request url* pada postman.



Gambar 16 Pengujian dengan Postman untuk Tampil List Jasa Make Up

Kemudian setelah link yang berupa rest tersebut dimasukkan pada kolom di postman maka seperti pada gambar 4.88 yang merupakan contoh dari pengujian restful pada aplikasi postman, yang dimana merupakan rest untuk menampilkan data list jasa make up yang tersedia dan data list jasa make up akan di tampilkan pada aplikasi untuk pelanggan maupun untuk admin.

#### 4. KESIMPULAN

Aplikasi mobile pemesanan jasa make up menggunakan restful web service telah berhasil dibangun berdasarkan kebutuhan admin dan user dan menghasilkan dua aplikasi yaitu aplikasi untuk admin dan untuk user yang saling terintegrasi, yang di mana fitur-fitur yang tersedia seperti admin dapat melakukan *crud*, admin dapat melihat rincian pesanan jasa make up yang telah di lakukan pelanggan, admin dapat melihat bukti pembayaran yang telah diupload oleh pelanggan, dan admin dapat mengubah status pesanan belum atau selesai serta admin dapat melihat pembukuan atau *history* pemesana. Pelanggan atau *user* dapat menggunakan aplikasi mobile ini untuk melihat dan melakukan pemesanan jasa make up yang tersedia dengan cara memilih jenis jasa make up yang diinginkan dan mengisi data diri seperti nama, alamat, dan catatan yang berupa pesan untuk admin, dan setelah user melakukan pemesanan maka pesanan user akan masuk ke keranjang belanja, ketika user telah yakin dengan pesanan yang dipilih nya maka *user* dapat men-

*check out* pesanya dan mengupload bukti pembayaran yang telah dilakukan *user*. *User* juga dapat melihat history pesanan yang telah dilakukannya.

#### 5. SARAN

Terdapat beberapa saran yang diharapkan sebagai perbaikan dan pengembangan lebih lanjut dari aplikasi *mobile* pemesanan jasa make up yaitu adanya fitur obrolan atau *chat* untuk admin dan pelanggan dan terdapat fitur GPS (*Global Positioning System*) untuk melihat titik lokasi *user* serta dapat fitur notifikasi sebagai pemberitahuan ketika ada pesanan yang masuk.

#### UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada pihak STMIK Pontianak yang telah memberikan kesempatan untuk melakukan penelitian ini.

#### DAFTAR PUSTAKA

- [1] Gat, 2016, Pengembangan Sistem Reservasi Hotel Berbasis Mobile dengan Framework JQuery Mobile, Program Studi Sistem Informatika, STMIK Pontianak.
- [2] Thomas K.W., 2010, Aplikasi Make Up 4U, Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra.
- [3] Swono N.K., Jayanti, 2018, Teori Basis Data, STIKOM Bali.
- [4] Ceria A.M., Sandy K., 2020, Rancangan dan Pengujian Perangkat Lunak Menggunakan Web Service pada Sistem Informatika Akademik, Jurusan Teknik Informatika, STMIK Pontianak.
- [5] Festim H., Web Service: A Comparison of SOAP and Rest Service, Fakultas Matematika Sains, Departemen Informatika.
- [6] Mark D.H., 2007, SOA Using Java Web Services, Pearson Education.
- [7] Guinard D., 2015, Restifying Real World System A Practical Case Study in RFID.
- [8] Roihan A., Aditya A.W., Wahyu P., 2019, Implementasi Metode Realtime Live Data dan Parsing JSON Berbasis Mobile dengan Menggunakan Android Studio dan PHP Native, Universitas Raharja.
- [9] R. Sianipar, 2015, Membangun Web PHP dan MySQL, Informatika, Bandung
- [10] Irfan K., Humaira, 2020, REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android, Jurusan Teknologi Informasi, Politeknik Negeri Padang.
- [11] Adiyanta F.C.S., 2019, Studi Penelitian Empiris : Penggunaan Metode Survei Sebagai Instrumen Penelitian, Fakultas Diponegoro.
- [12] Miroslaw S., 2020, Action Research in Software Engineering, Department of Computer Science and Engineering, University of Gothenburg, Sweden.
- [13] Zainal A. Hasibuan, 2007, Metodologi Penelitian pada Bidang Ilmu Komputer dan Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Indonesia.
- [14] Pressman Roger S., 2015, Software Engineering A Practitioner's Approach.
- [15] Mahdi, 2015, Comparison of Software Testing Review Black Box, Department of Computer, Science and Research Branch, Islamic Azad University.