

Penerapan *REST API* dalam Pengembangan Aplikasi Pemesanan Rental Mobil berbasis *Web* dan *Mobile* (Studi Kasus: CV. Dwi Cipta Rent Car)

I Gede Merta Ariantara¹, Issa Arwani², Widhy Hayuhardhika Nugraha Putra³

Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹merta_ariantara@student.ub.ac.id, ²issa.arwani@ub.ac.id, ³widhy@ub.ac.id

Abstrak

Usaha penyewaan kendaraan semakin banyak bermunculan sehingga membuat persaingan semakin ketat, pemilik usaha dituntut berinovasi, misal dengan menggunakan teknologi dalam menyelesaikan kegiatan operasionalnya. CV. Dwi Cipta Rent Car merupakan usaha yang bergerak di bidang jasa penyewaan kendaraan, khususnya mobil. Pada CV. Dwi Cipta Rent Car *customers* yang ingin melakukan penyewaan terkadang harus datang terlebih dahulu ke tempat penyewaan untuk melakukan *booking*, lalu datang kembali untuk mengambil mobil yang dipinjam sesuai waktu yang ditentukan saat melakukan *booking*, dan juga terkadang saat melakukan *booking* kendaraan yang ingin dipesan *customer* sudah habis. Selain itu transaksi *customer* masih dilakukan secara manual dengan mencatat data pemesanan *customer* yang menyewa kendaraan pada buku sewa. Berdasarkan masalah yang ada, maka pada penelitian ini dibuatkanlah sebuah aplikasi mengenai pemesanan rental mobil. Aplikasi ini dikembangkan memanfaatkan teknologi API (*Application Programming Interface*) yaitu *REST API*. Pengembangan aplikasi dikembangkan dengan metode pengembangan perangkat lunak *waterfall*, pada *platform web* dan juga *mobile*. Setelah itu dilakukan pengujian dengan *validation testing* dan *compatibility testing*. Hasil *validation testing* menggunakan metode *black-box testing* mendapatkan hasil valid untuk semua kebutuhan fungsional yang diujikan. Sedangkan untuk *compatibility testing* mendapatkan hasil bahwa sistem dapat dijalankan diberbagai browser yang diujikan untuk *web* dan berbagai API level yang berbeda untuk aplikasi *mobile*.

Kata kunci: penyewaan kendaraan, *REST API*, *web*, *mobile*,

Abstract

Vehicle rental businesses have sprung up and making the competition fiercer, business owners are required to innovate, for example by using technology in completing their operational activities. CV. Dwi Cipta Rent Car is a business that operates in the field of vehicle rental services, specifically cars. On the CV. Dwi Cipta Rent Car customer who want to make a rental sometimes must first come to the rental shop to make an order, then come back to pick up the booked car according to the specified time when making the order, and sometimes the vehicle that the customer wants to order is already up. In addition, customer transactions are still done manually by recording customer ordering vehicle data in the rental journal. Based on the existing problems, in this study an application was made regarding rental car reservation. This application was developed utilizing API (Application Programming Interface) technology, namely REST API. The application developed by the waterfall software development method, on the web and mobile platforms. The testing is done with validation testing and compatibility testing. Validation testing results using the black-box testing method got valid results for all the functional needs that have been tested. While for the compatibility testing, the result is the system can run on various browsers for the web, and various different API levels for mobile application.

Keywords: *vehicle rental, REST API, web, mobile,*

1. PENDAHULUAN

Pariwisata merupakan salah satu sektor andalan yang memiliki potensi besar dalam

menyumbang pendapatan daerah di beberapa wilayah yang ada di Indonesia, salah satunya yang terkenal adalah Pulau Bali. Pariwisata di Bali terus berkembang dalam meningkatkan perekonomian nasional (Wirasutama,

Suthanaya, dan Wedagama, 2015). Seiring perkembangan waktu, keindahan alam Pulau Bali memunculkan banyak destinasi wisata di berbagai daerah di Bali. Dengan perkembangan tersebut tentunya membuka peluang usaha bagi banyak orang yang ada di Bali khususnya Denpasar yang menjadi pusat kota, salah satunya yaitu usaha yang bergerak di bidang jasa penyewaan kendaraan. Banyaknya tempat wisata dan jarak antar lokasi yang tidak begitu berdekatan membuat tak sedikit dari wisatawan yang datang ke Bali memilih untuk menyewa kendaraan untuk berkeliling tempat-tempat wisata yang ada.

Usaha penyewaan kendaraan ini semakin banyak bermunculan sehingga membuat persaingan semakin ketat, pemilik usaha dituntut berinovasi, misal dengan menggunakan teknologi dalam menyelesaikan kegiatan operasionalnya. CV. Dwi Cipta *Rent Car* yang beralamat di Jalan Danau Tamblingan no. 68 Denpasar merupakan usaha yang bergerak dalam bidang jasa penyewaan kendaraan, khususnya mobil yang ada di Denpasar. Dimana pada CV. Dwi Cipta *Rent Car customer* yang ingin melakukan penyewaan mobil, terkadang harus datang terlebih dahulu ke tempat penyewaan untuk melakukan *booking*, lalu datang kembali untuk mengambil mobil yang dipinjam sesuai dengan waktu yang ditentukan saat melakukan *booking*, hal tersebut mengakibatkan banyak waktu yang terbuang, dan juga terkadang kendaraan yang ingin dipesan *customer* sudah habis. Selain itu juga terdapat permasalahan lain yang ada di CV. Dwi Cipta *Rent Car* dimana transaksi *customer* masih dilakukan secara manual dengan mencatat data pemesanan *customer* yang menyewa kendaraan pada buku sewa, hal tersebut yang sering mengakibatkan terjadinya kehilangan maupun kesulitan mencari data transaksi yang ada. Dan juga masalah tersebut juga mengakibatkan terhambatnya penyusunan dalam laporan keuangannya.

Maka berdasarkan permasalahan yang ada pada CV. Dwi Cipta *Rent Car*, pada penelitian ini dibuatkanlah sebuah aplikasi mengenai pemesanan rental mobil. Aplikasi yang dikembangkan terdiri dari aplikasi *mobile* yang digunakan oleh *customers* untuk melakukan penyewaan mobil, dan aplikasi *web* untuk admin. Aplikasi ini dikembangkan dengan menggunakan teknologi API (*Application Programming Interface*) yaitu *REST API*. API adalah *software interface* yang terdiri dari kumpulan instruksi yang disimpan kedalam

bentuk *library* dan menjelaskan bagaimana supaya suatu *software* bisa berinteraksi dengan *software* yang lain (Arianto, Munir, Khotimah, 2016). Tujuan digunakannya *REST API* disini untuk mengintegrasikan antara aplikasi *mobile* dan aplikasi *web* yang akan dibuat. Selain itu juga, penggunaan *REST API* disini juga digunakan untuk memudahkan *developer* jika nanti kedepannya akan dilakukan pengembangan lebih lanjut terkait aplikasi yang dikembangkan saat ini, karena *REST API* sendiri menyediakan sebuah fungsi yang terpisah sehingga *developer* tidak perlu lagi membuat fitur yang sama secara berulang.

2. LANDASAN PUSTKA

2.1. Penyewaan

Menurut Kamus Besar Bahasa Indonesia (KBBI), sewa merupakan pemakaian sesuatu dengan melakukan pembayaran uang sewa, uang yang dibayar dikarenakan memakai atau melakukan meminjamkan sesuatu. Menurut Nauvzal & Julian, (2019) Penyewaan adalah yaitu kegiatan kontrak dimana satu pihak yang memberi sewa, memberikan hak menggunakan jasa atau barang tertentu dalam jangka waktu tertentu kepada pihak penyewa dengan ketentuan yang telah disepakati bersama.

2.2. Web Service

Menurut *World Wide Web Consortium*(W3C), *Web service* adalah suatu bentuk dari layanan interaksi mesin-ke-mesin melalui jaringan yang didesain dan dirancang untuk sistem perangkat lunak. *Web service* sendiri dibuat dalam format yang dapat dibaca oleh mesin untuk *interfacenya*.

2.3. API (*Application Programming Interface*)

API merupakan sekumpulan instruksi program yang digunakan untuk membangun sebuah aplikasi perangkat lunak, yaitu dengan menyediakan fungsi dan perintah dengan bahasa yang mudah dipahami programmer. *API* sendiri memungkinkan pembuat aplikasi untuk mengakses fungsi atau data dari sebuah sistem operasi, aplikasi atau layanan lainnya (Huda, 2020).

2.4. REST API

Menurut Feridi (2019) *REST API* (*Representational State Transfer*) *API*

merupakan standar dari arsitektur untuk pengembangan sistem berbasis web sebagai protocol untuk komunikasi data. Arsitektur *REST* merupakan hubungan antara client dan server, yang dimana sumber daya/data disediakan oleh REST server dan sumber daya/data tersebut untuk selanjutnya diakses dan ditampilkan oleh REST client. Setiap sumber daya/data diidentifikasi oleh URLs (*Universal Resource Identifiers*). Setiap sumber daya/data tersebut kemudian ditampilkan dalam bentuk teks, XML atau JSON. Umumnya menggunakan XML dan JSON. *REST* menjelaskan bagaimana aplikasi berkomunikasi melalui Hypertext Transfer Protocol (HTTP).

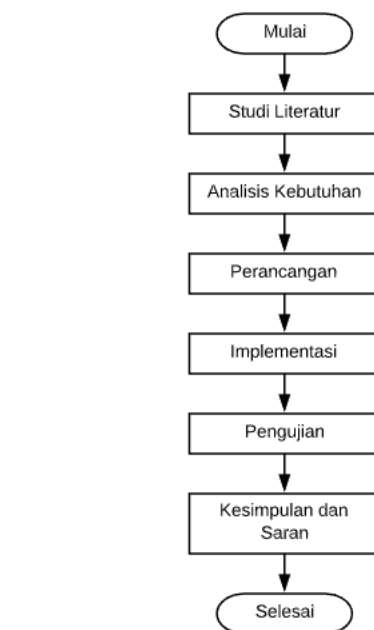
2.5. Web

Web merupakan aplikasi dengan informasi yang berbentuk multimedia seperti gambar, teks, suara, animasi dan lain-lain yang disimpan pada sebuah server Web Internet dan hiperteks sebagai bentuk penyajiannya. Informasi yang terdapat dalam web menggunakan format (HTML) Hypertext Markup Language berbentuk teks. Informasi bentuk sperti grafis digunakan format (PNG, GIF, JPG), untuk format suara (WAV, AU), dan untuk objek multimedia lain bisa menggunakan format seperti(Quicktime Movie, 3D World, Shockwave, MIDI) (Simarmata, 2010).

2.6. REST API

Aplikasi mobile yang sering disingkat dengan istilah Mobile Apps ialah aplikasi dari sebuah perangkat lunak yang dapat berjalan diperangkat mobile dalam pengoperasiannya seperti (Tablet, Smartphone, iPod, dll), dan memiliki sistem operasi yang mendukung perangkat lunak secara *standalone*.

3. METODOLOGI



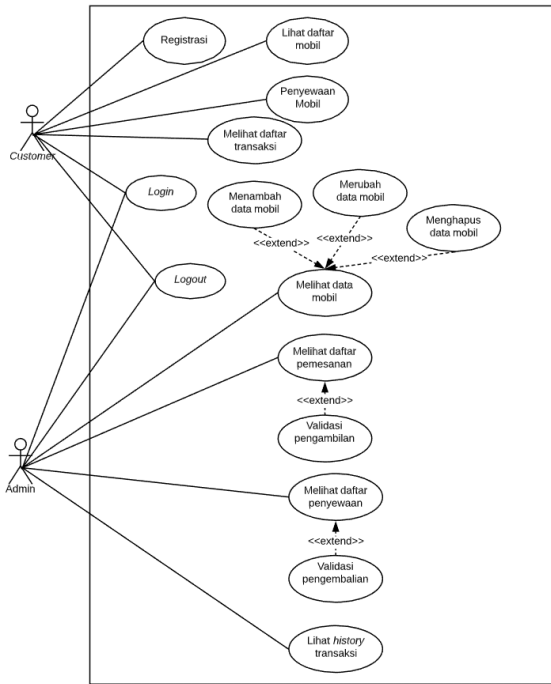
Gambar 1. Diagram alir metodologi penelitian

Gambar 1 merupakan metodologi dalam pengerjaan penelitian ini. Jenis penelitian ini merupakan penelitian implementatif dimana pada pengembangan sistemnya menggunakan model pengembangan *waterfall*, yang dimulai dari tahap studi literatur untuk menemukan berbagai sumber dan referensi yang dapat digunakan untuk mendukung penelitian ini. Lalu ada analisis kebutuhan untuk memperoleh kebutuhan baik fungsional maupun non-fungsional. Berikutnya dilakukan perancangan kedalam bentuk diagram UML dari kebutuhan yang didapat. Lalu selanjutnya melakukan implementasi kedalam kode program dari perancangan yang dibuat. Setelah itu dilakukan pengujian dari hasil implementasi. Dan tahap terakhir dilakukan penarikan kesimpulan terhadap tahap-tahap yang telah dilakukan sebelumnya serta saran yang dapat digunakan untuk penelitian selanjutnya.

4. ANALISIS KEBUTUHAN

Sistem pemesanan rental mobil ini adalah adalah sebuah sistem berbasis *web* dan *android* dengan menggunakan *rest api*, yang dibuat untuk CV. Dwi Cipta Rent Car guna memberikan kemudahan dalam melakukan proses transaksi baik dari pihak *customer* yang ingin melakukan peminjaman, maupun dari pihak CV. Dwi Cipta Rent Car yang ingin mengolah data transaksi dan juga data mobil yang disewakan. Analisis kebutuhan dari sistem

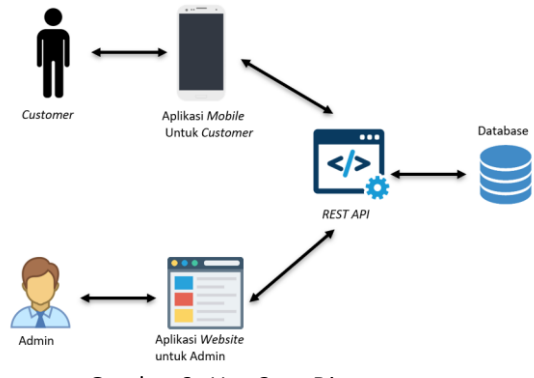
pemesanan rental mobil didapat melalui proses wawancara dengan pemilik dari CV. Dwi Cipta Rent Car dan didapatkan 2 aktor yaitu admin dan customer, serta 14 kebutuhan fungsional yang dimodelkan kedalam use case diagram pada Gambar 2.



Gambar 2. Use Case Diagram

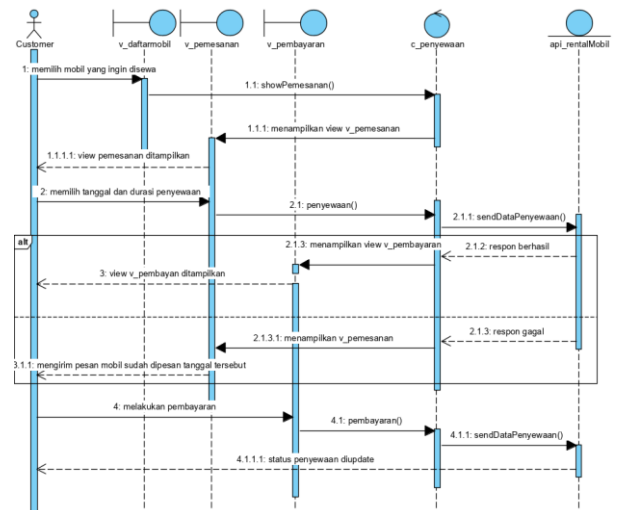
5. PERANCANGAN

Pada Gambar 3 merupakan perancangan arsitektur dari sistem yang akan dibuat, dimana terdiri dari dua aktor yaitu admin dan customer. Lalu pada aplikasi pemesanannya terdapat aplikasi mobile, web, rest api dan juga database. Dimana aplikasi mobile digunakan oleh customer untuk melakukan pemesanan mobil, aplikasi web digunakan oleh admin untuk mengelola data transaksi, rest api digunakan untuk berkomunikasi antara aplikasi mobile dan web serta digunakan untuk akses ke database dan database digunakan untuk menyimpan semua data penyewaan.

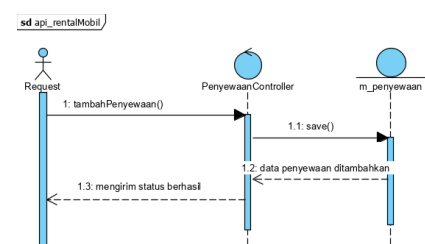


Gambar 3. Use Case Diagram

Perancangan selanjutnya terdapat sequence diagram, Sequence diagram menggambarkan pemodelan dari interaksi diantara aktor dengan objek pada suatu sistem, dan interaksi antara object itu sendiri. Seperti namanya sequence diagram menunjukkan urutan yang terjadi pada use case diagram. Terdapat 15 sequence diagram dari kebutuhan fungsional pada sistem pemesanan rental mobil dan 13 sequence diagram untuk REST API rental mobil. Gambar 4 merupakan sequence diagram dari penyewaan mobil dan Gambar 5 merupakan sequence diagram dari API penyewaan mobil.



Gambar 4. Sequence Diagram Penyewaan Mobil

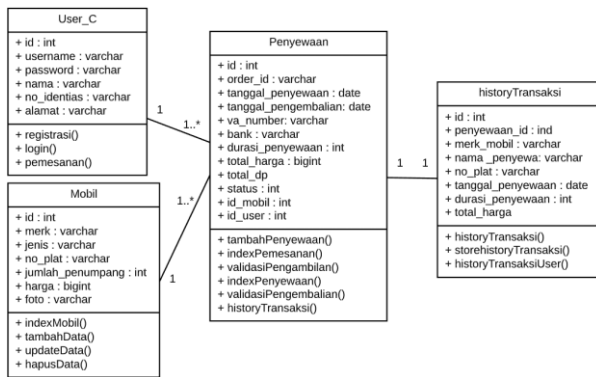


Gambar 5. Sequence Diagram API Penyewaan Mobil

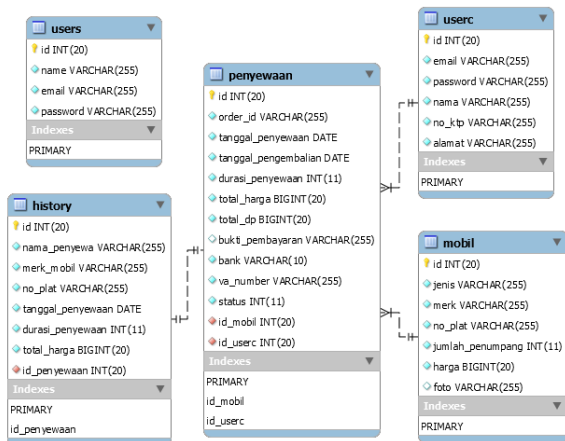
Selanjutnya terdapat class diagram yang dibuat berdasarkan *sequence* diagram dan terdiri dari 4 class yaitu, class penyewaan, userc, mobil, dan *history*. Class diagram sistem pemesanan rental mobil dapat dilihat pada Gambar 6.

Terdapat rancangan *physical data model* untuk dijadikan acuan dalam implementasi dari database yang akan dibuat. Rancangan *physical data model* dapat dilihat pada Gambar 7.

Dilakukan juga perancangan dari REST API, dari segi url yang akan diakses oleh web client, method yang digunakan, request yang dibawa, dan respon yang diterima dalam bentuk status code dan juga konten dari respon itu sendiri. Dimana terdapat 13 rancangan dari REST API dan pada Tabel 1 merupakan rancangan REST API untuk penyewaan mobil.



Gambar 6. Class diagram



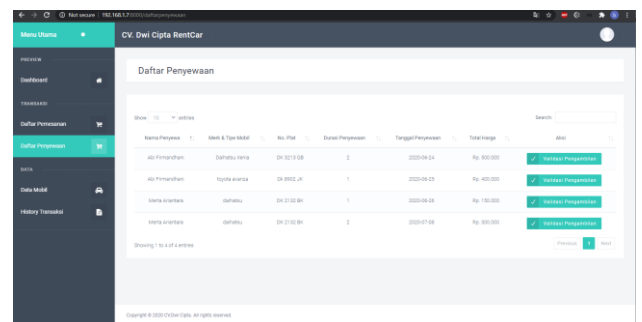
Gambar 7. Rancangan *Physical data model*

Tabel 1. Rancangan API Penyewaan Mobil

| Penyewaan Mobil | |
|---------------------|---|
| URL | /penyewaan |
| Method | POST |
| Request body | userC_id: int (20) mobil_id: int (20) order_id: int (20) tanggal_penyewaan: date tanggal_pengembalian: date waktu_pengambilan: time durasi_penyewaan: int (11) total_dp: bigint (20) total_harga: bigint (20) bank: varchar (10) va_number: varchar (203) status: int (11) |
| Response | |
| Status Code | 201 Created |
| Content | { "status": "berhasil" } |

6. IMPLEMENTASI

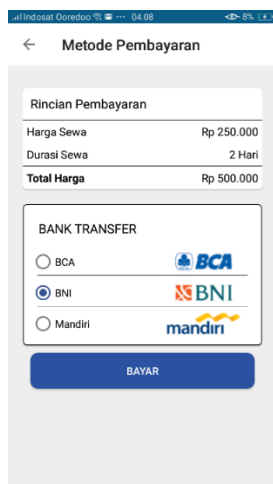
Dilakukan implementasi dari perancangan yang dibuat kedalam bentuk kode program. Sistem di implementasikan menggunakan *framework* lumen untuk REST API, laravel untuk *web*, dan pengembangan aplikasi *mobile* menggunakan android studio. Pada tahap ini diimplementasikan juga *user interface* dari aplikasi yang dibuat. Seperti pada Gambar 8 merupakan antarmuka dari daftar penyewaan yang ada pada aplikasi *web*. Sedangkan Gambar 9 merupakan implementasi dari antarmuka pada aplikasi *mobile* untuk melihat data mobil, dan Gambar 10 juga merupakan implementasi dari antarmuka pada aplikasi *mobile* untuk memilih metode pembayaran.



Gambar 8. Implementasi Antarmuka Daftar Penyewaan pada aplikasi *web*



Gambar 9. Implementasi antarmuka data mobil pada aplikasi *mobile*



Gambar 10. Implementasi antarmuka metode pembayaran pada aplikasi *mobile*

7. PENGUJIAN

7.1. Pengujian fungsional

Dari pengujian fungsional yang dilakukan didapatkan hasil bahwa semua fungsi yang ada pada aplikasi pemesanan rental mobil dapat berjalan sebagaimana mestinya sesuai dengan kebutuhan yang didefinisikan.

Hal tersebut didukung dari status pengujian validasi yang dilakukan dimana didapatkan status valid untuk semua kasus uji. Tabel 2 merupakan salah satu pengujian validasi pada fungsi penyewaan mobil.

Tabel 2. Pengujian Fungsi Penyewaan Mobil

| Penyewaan Mobil | |
|-----------------|------------|
| Kode Kebutuhan | SPRM_F_005 |

| | |
|---------------------------|---|
| Tujuan Pengujian | Pengujian bertujuan memastikan pengguna dapat melakukan penyewaan mobil dan data penyewaan ditampilkan dalam aplikasi <i>web</i> |
| Kasus Pengujian | Pengujian dilakukan terlebih dahulu dengan pengguna harus <i>login</i> kedalam sistem lalu memilih mobil yang ingin dipesan setelah itu memilih tanggal penyewaan, dan melakukan pembayaran |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Terlebih dahulu penguji <i>login</i> kedalam aplikasi <i>mobile</i>. 2. Penguji memilih mobil yang disewakan. 3. Penguji memilih tanggal penyewaan, jam pengambilan, dan juga durasi penyewaan. 4. Penguji memilih metode pembayaran. Penguji melakukan pembayaran. |
| Hasil Pengujian | Pengguna dapat melakukan penyewaan mobil dan data penyewaan berhasil ditampilkan pada aplikasi <i>web</i> |
| Status | Valid |

7.1. Pengujian non-fungsional

Pada kebutuhan non-fungsional dilakukan pengujian *compatibility* untuk mengetahui apakah aplikasi yang dibuat bisa digunakan pada berbagai macam perangkat keras, sistem operasi, jaringan, aplikasi atau perangkat bergerak yang berbeda beda. Pada pengujian *compatibility* ini digunakan *tools SortSite* untuk menguji aplikasi *web* dan *firebase test lab* untuk menguji aplikasi *mobile*.

Untuk pengujian pada *SortSite* menunjukkan bahwa aplikasi web yang dibuat dapat berjalan pada berbagai macam browser yang diujikan. Hasil pengujiannya dapat dilihat pada Gambar 11.

Pengujian *compatibility* pada *firebase test lab* juga menunjukkan bahwa aplikasi *mobile* yang dibuat dapat berjalan pada semua API level yang diujikan. Hasil pengujiannya dapat dilihat pada Gambar 12.

| Browser | IE | Edge | Firefox | Safari | Opera | Chrome | iOS | | | Android | |
|-----------------|----|------|---------|--------|-------|--------|------|----|----|---------|----|
| Version | 11 | 81 | 76 | 13 | 68 | 81 | ≤ 11 | 12 | 13 | ≤ 3 | 4* |
| Critical Issues | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| Major Issues | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| Minor Issues | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |

Gambar 11. Pengujian *compatibility* aplikasi *web*

| Device | Locale | Orientation | Failed | Flaky | Passed | Skipped | Inconclusive |
|--|-------------------------|-------------|--------|-------|--------|---------|--------------|
| Galaxy A5 2017, API Level 24 | English (United States) | Portrait | 0 | 0 | 0 | 0 | 0 |
| Essential PH-1, API Level 25 | English (United States) | Portrait | 0 | 0 | 0 | 0 | 0 |
| Galaxy J2 Prime SM-G532M, API Level 23 | English (United States) | Portrait | 0 | 0 | 0 | 0 | 0 |
| Pixel 2, API Level 28 | English (United States) | Portrait | 0 | 0 | 0 | 0 | 0 |

Gambar 12. Pengujian *compatibility* aplikasi *mobile*

8. KESIMPULAN DAN SARAN

8.1. Kesimpulan

Berdasarkan dari serangkaian tahap penelitian yang dilakukan sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Pada proses analisis kebutuhan yang dilakukan, didapatkan hasil yang terdiri dari identifikasi aktor yang berjumlah 2 yaitu admin dan *customer*, kebutuhan fungsional dari sistem yang akan dibuat berjumlah 15, dan kebutuhan non-fungsional berjumlah 1. Dari analisis kebutuhan fungsional yang didapat kemudian dimodelkan kedalam *use case* diagram yang selanjutnya diperjelas lebih rinci pada *use case scenario*, dan juga dibuatkan *activity* diagram.
2. Pada tahap perancangan sistem, dihasilkan arsitektur dari sistem yang akan dibuat, serta pemodelan analisis kebutuhan yang didapat bentuk diagram UML yang terdiri dari 16 *sequence* diagram dari kebutuhan fungsional, 14 *sequence* diagram dari *REST API*, rancangan *class* diagram yang terdiri dari 4 *class*, rancangan database yang terdiri dari 5 tabel. Serta terdapat juga 12 rancangan *REST API* dan juga 17 rancangan antarmuka dari sistem.
3. Pada tahap implementasi terdapat hasil yaitu penentuan spesifikasi lingkungan implementasi dari pengembangan aplikasi yang terdiri dari spesifikasi *hardware* dan juga *software*. Dilanjutkan dengan implementasi database dengan menggunakan MySQL, lalu implementasi fungsi yang ada pada aplikasi *mobile* menggunakan bahasa pemrograman java, aplikasi *web* dan *REST API* menggunakan bahasa pemrograman PHP dengan menggunakan *framework* Laravel untuk

web dan Lumen untuk *REST API*. Setelah itu implementasi antarmuka dari aplikasi *web* menggunakan HTML, JavaScript, dan CSS, sedangkan untuk aplikasi *mobile* menggunakan XML.

4. Pada tahap pengujian dilakukan pengujian dari kebutuhan fungsional dan non-fungsional yang telah didefinisikan pada tahap analisis kebutuhan. Pada pengujian fungsional dilakukan dengan menggunakan metode black-box testing yaitu pengujian validasi. Hasil dari pengujian validasi yaitu bahwa semua kebutuhan fungsional dapat bekerja dengan sebagaimana mestinya atau bernilai valid. Dan untuk kebutuhan non-fungsional dilakukan pengujian *compatibility testing* dengan menggunakan bantuan *tools* sortSite dan *firebase test lab* yang menunjukkan hasil bahwa aplikasi *web* dapat berjalan pada semua *browser* yang diujikan dan untuk aplikasi *mobile* dapat dijalankan pada API level yang berbeda.

8.2. Saran

Dari penelitian mengenai aplikasi pemesanan rental mobil yang telah dilakukan, terdapat saran yang diberikan sebagai bahan untuk dijadikan pertimbangan dalam pengembangan lanjut pada aplikasi pemesanan rental mobil, yaitu dengan penambahan fitur untuk melakukan pelacakan lokasi kendaraan yang sedang disewa oleh *customer*, dan informasi mengenai lokasi kendaraan tersebut dapat ditampilkan kedalam aplikasi *web* yang digunakan oleh admin.

DAFTAR PUSTAKA

- Arianto, Mukhammad Agus; Munir, Sirojul; Khotimah, K. (2016). Analisis dan Perancangan Representational State Transfer (REST) Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii Framework. *Jurnal Teknologi Terpadu*, 2(2), 1–8.
- Feridi, 2019. Mengenal RESTful Web Service, [online] Tersedia di: <<https://www.codepolitan.com/mengenal-restful-web-services>> [Diakses 4 Maret 2020]
- Huda, Miftaul (2020) Operation System Interconnection [e-book] Jawab Barat: Bisakimia Tersedia di: Google Play

Books [Diakses 22 Februari 2020]

Kamus Besar Bahasa Indonesia. [Online]. Tersedia melalui <kbbi.kemdikbud.go.id/entri/sewa>. [Diakses 19 Februari 2021]

Nauvzal, E. D., & Julian, W. C. (2019). *Sistem Informasi Penyewaan Pada Barokah Rental Mobil Berbasis Web* (April).

Simamata, Janner (2010) *Rekayasa Web* [e-book] Yogyakarta: C.V Andi Offset Tersedia di: Google Play Books [Diakses 04 Maret 2020]

Wirasutama, C. P., Suthanaya, P. A., & Wedagama, D. M. P. (2015). Analisis Kelayakan Investasi Angkutan Pariwisata Di Propinsi Bali. *Jurnal Spektran*, 3(1), 10–19.

W3C, 2002. *Web Service Description Requirements*. [Online] Tersedia melalui: <<https://www.w3.org/TR/ws-desc-reqs/>> [Diakses 20 Februari 2020]