

Pengembangan Aplikasi Web Reservasi Paket Wisata menggunakan MERN Stack (Studi Kasus: Zona Tamasya Tour Organizer)

Rezki Triandy¹, Nurudin Santoso²

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹rezkitriandy.contact@gmail.com, ²nurudin.santoso@ub.ac.id

Abstrak

Zona Tamasya sebagai salah satu agen travel, menyediakan beragam layanan wisata. Layanan yang paling populer dan diminati adalah paket wisata *open trip*. Pelanggan yang ingin mengikuti *open trip* tersebut dapat memesan atau mereservasi tempat terlebih dahulu dengan menghubungi pihak Zona Tamasya (*admin*) melalui aplikasi Whatsapp. Saat terjadi peningkatan jumlah pelanggan terutama saat musim liburan, *admin* Zona Tamasya cukup kewalahan untuk merespons setiap pesan yang masuk dari pelanggan, baik dari pelanggan yang hanya menanyakan informasi sampai melakukan konfirmasi pembayaran. Kelemahan ini membuat Zona Tamasya berencana untuk meningkatkan kualitas layanannya dengan memanfaatkan *website* sebagai media informasi maupun transaksi (reservasi). Pada penelitian ini dikembangkan aplikasi berbasis web yang dapat digunakan untuk melakukan reservasi paket wisata. Aplikasi ini dikembangkan menggunakan teknologi MERN stack (MongoDB, Express, React, dan Node.js) serta mengintegrasikan sistem pembayaran *online*. Pengembangan dilakukan mengikuti prinsip rekayasa perangkat lunak secara *linear*. Hasil penelitian adalah aplikasi berbasis web dengan fitur utama reservasi paket wisata dan pembayaran *online* yang telah diuji fungsionalitas nya.

Kata kunci: pengembangan aplikasi web, reservasi paket wisata, *full JavaScript development*, *single-page application*, MERN stack, basis data NoSQL, *payment gateway*

Abstract

Zona Tamasya as one of the travel agency provides a variety of tour services. The most popular and best seller service is the open trip tour package. Customers who want to take an open trip can book or reserve a place in advance by contacting Zona Tamasya (admin) via the Whatsapp application. When there is an increase in the number of customers, especially during the holiday season, the admin of Zona Tamasya is quite overwhelmed to respond to any messages that come from customers, both from customers who only ask for information to confirm payment. This weakness makes Zona Tamasya plan to improve the quality of its services by utilizing the website as a medium of information and transactions (reservations). In this research, a web-based application was developed that can be used to make travel package reservations. This application was developed using MERN stack technology (MongoDB, Express, React, and Node.js) and integrates online payment systems. Development is carried out following the principles of software engineering in a linear fashion. The results of the study are web-based applications with the main features of package tours and online payment that have been tested for their functionality.

Keywords: *web application development, tour package reservation, full JavaScript development, single-page application, MERN stack, NoSQL database, payment gateway*

1. PENDAHULUAN

Perkembangan teknologi yang begitu pesat, salah satunya ditandai dengan penggunaan internet yang terus meningkat. Akses internet yang semakin mudah memunculkan suatu bentuk jenis aplikasi baru yaitu aplikasi berbasis

web/aplikasi web (*web based application/webapp*). Aplikasi web pada dasarnya adalah bentuk pengembangan lanjut dari *web page* yang sebelumnya hanya menampilkan informasi statis (Löfberg & Molin, 2005). Adanya aplikasi web memungkinkan pengguna untuk mengakses aplikasi kapan pun dan dimana pun (Andhale, et al., 2017).

Saat ini, aplikasi web banyak dimanfaatkan dalam dunia bisnis digital yang dikenal dengan era *e-commerce*. Dengan memanfaatkan aplikasi web, pelaku bisnis menjadi lebih mudah dalam memperoleh informasi maupun mendukung aktivitas bisnisnya seperti pemasaran, penjualan, dan pelayanan pelanggan (Irmawati, 2011). Manfaat ini juga mendorong pihak Zona Tamasya untuk melakukan digitalisasi terhadap bisnisnya. Zona Tamasya merupakan *tour organizer/agen travel* yang bergerak di bidang jasa perjalanan wisata. Layanan yang ditawarkan salah satunya adalah paket wisata. Selama ini, untuk melakukan promosi maupun penjualan, pihak Zona Tamasya memanfaatkan aplikasi media sosial. Meskipun demikian, proses bisnis pada dasarnya masih dilakukan secara konvensional.

Penjualan jasa Zona Tamasya, yaitu paket wisata, berkaitan dengan proses pemesanan atau reservasi. Pelanggan yang ingin mengikuti perjalanan wisata (paket wisata) perlu melakukan reservasi terlebih dahulu melalui aplikasi Whatsapp. Pelanggan dapat menanyakan informasi terkait paket wisata yang dipesan kemudian dapat mengisi data diri pada formulir reservasi. Selanjutnya pelanggan dapat melakukan pembayaran melakukan konfirmasi pembayaran kepada pihak Zona Tamasya.

Mekanisme reservasi melalui aplikasi Whatsapp memiliki kelemahan yang terjadi saat jumlah pelanggan meningkat. Meningkatnya jumlah pelanggan ini biasanya terjadi saat musim liburan (*high season*). Pihak Zona Tamasya, dalam hal ini adalah *admin*, cukup kewalahan dan membutuhkan waktu untuk dapat membalas setiap pesan pelanggan baik untuk memproses konfirmasi pembayaran maupun sekadar memberikan informasi. Selain itu, *admin* juga tidak selalu dalam posisi *standby* untuk membalas. Hal ini tentunya dapat mengurangi kualitas pelayanan terhadap pelanggan. Mengingat tingkat kualitas pelayanan berpengaruh terhadap kepuasan pelanggan (Prasetyo, 2012) serta reservasi yang merupakan aktivitas bisnis utama, maka dari itu Zona Tamasya telah memiliki rencana untuk memiliki *website* yang dapat digunakan sebagai media promosi, informasi, dan reservasi paket wisata.

Berdasarkan permasalahan pihak Zona Tamasya tersebut, pada penelitian ini dilakukan pengembangan aplikasi web untuk reservasi paket wisata menggunakan teknologi MERN *stack*. MERN merupakan salah satu kombinasi

teknologi yang digunakan untuk mengembangkan aplikasi web terdiri dari MongoDB, Express, React, dan Node.js. Menurut Subramanian (2017), MERN cocok digunakan untuk mengembangkan aplikasi web yang memiliki interaktivitas yang tinggi. Kelebihan utama dari MERN adalah digunakannya satu bahasa, yaitu JavaScript baik pada sisi client maupun server bahkan untuk mengelola basis data (Subramanian, 2017). Selain itu, aplikasi web yang dikembangkan juga mengintegrasikan sistem *payment gateway* sebagai fasilitas pembayaran *online* sehingga transaksi pembayaran lebih aman dan terpercaya..

2. TINJAUAN PUSTAKA

2.1. Zona Tamasya Tour Organizer

Zona Tamasya *Tour Organizer* adalah salah satu agen *travel* dan wisata yang berlokasi di Kota Surabaya, Jawa Timur dengan kantor yang beralamatkan di Jl. Kaliwaron No.58, Mojo, Kec. Gubeng, Surabaya dan didirikan pada tanggal 5 Januari 2016. Setelah berjalan selama hampir empat tahun, saat ini Zona Tamasya menyediakan berbagai layanan di bidang *travel* khususnya paket perjalanan wisata ke berbagai destinasi favorit para wisatawan. Dengan paket perjalanan wisata tersebut, peserta tidak perlu repot untuk menyiapkan keperluan atau akomodasi karena semua telah disiapkan oleh pihak Zona Tamasya.

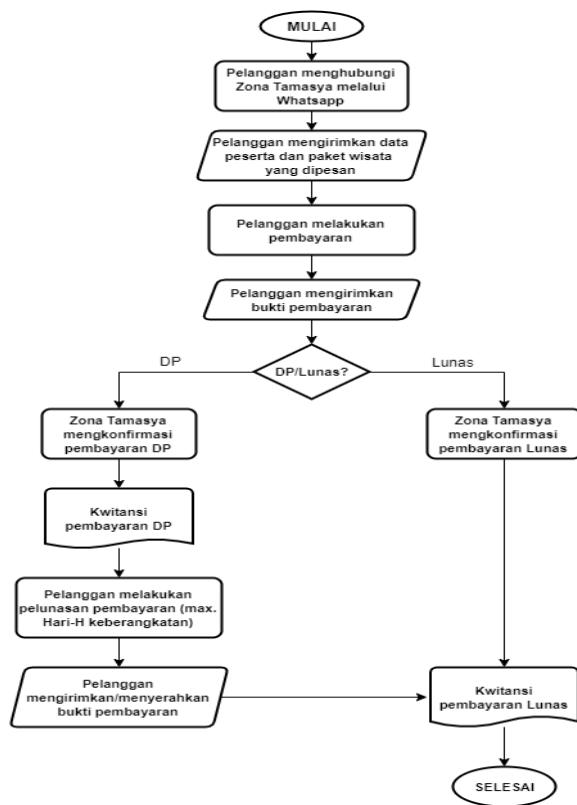
2.1.1. Open Trip

Open trip merupakan salah satu layanan paket wisata yang paling diminati oleh pelanggan. Secara sederhana *open trip* adalah berlibur bersama orang-orang yang sebelumnya tidak dikenal. Maka dari itu, *open trip* ditawarkan untuk umum dalam arti peserta *trip* tidak dibatasi untuk kelompok tertentu. Kelebihan paket wisata *open trip* ini salah satunya adalah biaya yang lebih murah dan mendapatkan pengalaman bertemu dengan teman-teman baru. Selain itu, tidak ada batasan minimum kuota peserta, berapa pun jumlah peserta nya, *trip* akan selalu diberangkatkan. Zona Tamasya sendiri memiliki jadwal rutin untuk *open trip*.

2.1.2. Reservasi Open Trip

Untuk melakukan pemesanan atau reservasi suatu *open trip*, pelanggan diharuskan untuk

menghubungi pihak Zona Tamasya. Prosedur reservasi dapat dilihat pada Gambar 1.



Gambar 1. Prosedur reservasi *open trip*

Setelah menghubungi pihak Zona Tamasya, pelanggan kemudian mengisi *form* reservasi dan memilih pembayaran yang tersedia, yaitu uang muka (DP) atau lunas. Pihak Zona Tamasya selanjutnya mencatat data reservasi tersebut.

2.2. Aplikasi Web

Salah satu dari pemanfaatan layanan web adalah aplikasi web (*web application*) atau disebut juga aplikasi berbasis web (*web-based application*). Dissanayake dan Dias (2017) berpendapat bahwa penggunaan kedua istilah tersebut berbeda, istilah *web-based application* memiliki cakupan ruang lingkup bahasan yang lebih luas dibanding istilah *web application*. Aplikasi web adalah program komputer yang menjalankan fungsinya pada *browser*. Maka dari itu, istilah web sebenarnya merujuk pada *platform* tempat aplikasi berjalan, sedangkan fokus utama adalah pada istilah “aplikasi” yang menjadi pembeda antara aplikasi web dengan *website* biasa (*web page*). Menurut Fowler dan Stanwick (2004) salah satu perbedaannya adalah dari aspek sifat interaksi. Pada *web page* sifat

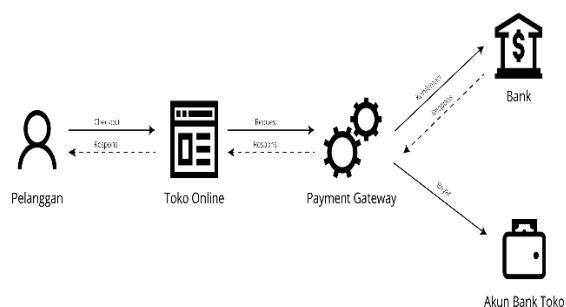
interaksi yang dimiliki adalah informasional, sedangkan aplikasi web memiliki sifat interaksi yang berorientasi transaksional.

2.3. MERN Stack

Kumpulan atau kombinasi teknologi yang digunakan dalam pengembangan aplikasi disebut dengan *stack*. Dalam lingkup pengembangan *website* atau aplikasi berbasis web, dikenal *stack* yang sangat populer yaitu LAMP (Linux, Apache, MySQL, dan PHP). Seiring dengan perkembangan dalam dunia pengembangan *website*, saat ini muncul konsep yang disebut dengan *Single Page Applications* (SPA). Munculnya konsep SPA menjadi pemicu munculnya teknologi-teknologi baru untuk mengembangkan *website* yang menerapkan SPA. Salah satu yang paling awal adalah MEAN *stack*. MEAN terdiri dari MongoDB, Express.js, AngularJS, dan Node.js. MERN pada dasarnya adalah penggantian “A” pada MEAN menjadi “R”, yaitu AngularJS menjadi ReactJS. ReactJS merupakan teknologi *front-end* yang dikembangkan oleh Facebook dan popularitasnya semakin meningkat (Subramanian, 2017). Baik MEAN atau MERN menggunakan satu bahasa yaitu JavaScript. Hal ini karena para *developer* menilai penggunaan bahasa yang berbeda dalam pengembangan web membuat pekerjaan semakin rumit (Adhikari, 2016).

2.4. Payment Gateway

Komponen utama dalam perdagangan atau bisnis adalah transaksi pembayaran. Oleh karena itu, perlu adanya teknologi yang menyediakan mekanisme pembayaran yang mudah, cepat, dan aman. Salah satu teknologi yang telah digunakan oleh *platform e-commerce* maupun pelaku bisnis lainnya adalah *payment gateway*. *Payment gateway* adalah jalur yang menghubungkan antara akun bank konsumen dengan platform pembayaran yang dituju (Parikh, 2019). Gambar 2 mengilustrasikan cara kerja dan peran *payment gateway* dalam suatu transaksi.



Gambar 2. Cara kerja *payment gateway*

2.4.1. Midtrans

Midtrans adalah salah satu penyedia layanan *payment gateway* di Indonesia. Dikutip dari website resminya, Midtrans didirikan pada 2012 seiring dengan dimulainya era *e-commerce* yang memiliki banyak permasalahan salah satunya adalah pembayaran *online*. Oleh sebab itu, Midtrans hadir dengan menawarkan infrastruktur pembayaran *online* yang terintegrasi, aman, dan terpercaya. Saat ini Midtrans telah bekerja sama dengan berbagai bank dan perusahaan ternama di Indonesia.

3. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan yang mencakup tahapan non-teknis dan tahapan teknis terkait pengembangan sistem. Tahapan penelitian dapat dilihat pada Gambar 3 berikut.



Gambar 3. Tahapan penelitian

Setiap tahap penelitian dapat diuraikan sebagai berikut:

1. Studi Pustaka

Tahap studi pustaka ditujukan untuk mendapatkan informasi dasar mengenai konsep pengembangan aplikasi web melalui kajian penelitian serupa serta informasi mengenai teori dari objek penelitian yaitu MERN *stack*.

2. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi aktor dan mendapatkan daftar kebutuhan sistem baik fungsional maupun non-fungsional sesuai dengan kebutuhan pihak Zona Tamasya. Daftar kebutuhan didapatkan dengan melakukan wawancara secara informal dengan pemilik Zona Tamasya serta pengamatan terhadap dokumen rencana pengembangan web. Pada tahap ini juga dilakukan pemodelan kebutuhan.

3. Perancangan

Perancangan merupakan tahap untuk merepresentasikan kebutuhan dalam bahasa teknis melalui model yang tersedia. Level perancangan dimulai dari arsitektur sistem, komponen, data, dan antarmuka. Beberapa diagram yang digunakan antara lain *class diagram*, *sequence diagram*, dan PDM.

4. Implementasi

Implementasi adalah tahapan untuk merealisasikan sistem sesuai dengan kebutuhan dan rancangannya menggunakan MERN *stack*. Tahap implementasi terbagi dalam dua sisi, yaitu *front-end* dan *back-end*. Implementasi *front-end* menggunakan ReactJS sedangkan *back-end* menggunakan Express, Node.js, dan MongoDB untuk implementasi basis data.

5. Pengujian dan Evaluasi

Tahap akhir pengembangan sistem adalah melakukan pengujian terhadap kebutuhan yang telah diimplementasikan. Pengujian terdiri dari pengujian unit dengan metode *white-box*, pengujian validasi dengan metode *black-box*, serta pengujian *compatibility*. Selanjutnya dilakukan evaluasi sebagai penilaian terhadap hasil implementasi berdasarkan pengujian.

6. Kesimpulan dan Saran

Penelitian berakhir dengan mendefinisikan kesimpulan yang didapatkan serta saran terkait dengan perbaikan penelitian.

4. ANALISIS KEBUTUHAN

Berikut ini akan diuraikan hasil analisis kebutuhan yaitu identifikasi aktor dan kebutuhan sistem.

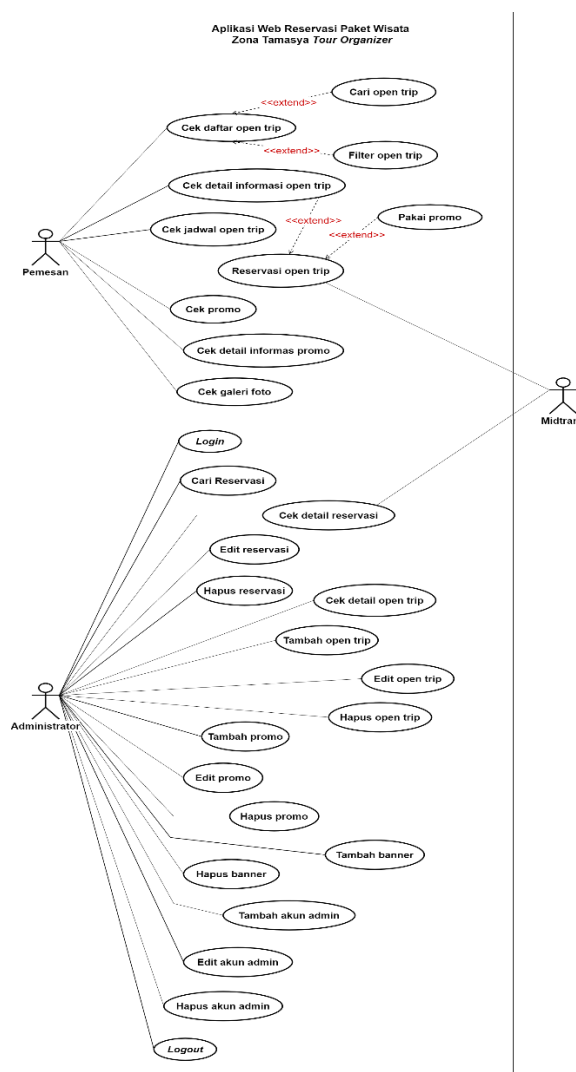
4.1. Identifikasi Aktor

Aktor atau pengguna adalah pihak yang dapat berinteraksi dengan sistem baik suatu individu, proses, atau sistem lainnya. Berdasarkan hasil analisis, didefinisikan sejumlah tiga aktor yang disebutkan pada Tabel 1 berikut.

Tabel 1. Daftar aktor dan karakteristik

Aktor	Karakteristik
Pemesan	Pemesan adalah individu yang dapat melakukan pemesanan paket wisata yang tersedia tanpa harus login ke dalam sistem.
Administrator	Administrator merupakan individu dari pihak Zona Tamasya yang memiliki akses <i>login</i> ke halaman admin sehingga dapat memantau dan mengelola informasi terkait.
Midtrans	Midtrans merupakan penyedia sistem <i>payment gateway</i> bertanggung jawab dalam memproses transaksi pembayaran reservasi.

4.1. Identifikasi Kebutuhan



Gambar 4. Use-case diagram aplikasi reservasi paket wisata

Kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Dari hasil analisis kebutuhan, aplikasi reservasi paket wisata ini memiliki sejumlah 28 kebutuhan fungsional dan 1 kebutuhan non-fungsional, yaitu *compatibility*. Gambar 4 adalah representasi kebutuhan fungsional dalam *use-case diagram*.

5. PERANCANGAN

Perancangan ditujukan untuk membuat model sebagai representasi sistem. Berikut ini adalah uraian hasil perancangan yang terdiri dari beberapa level.

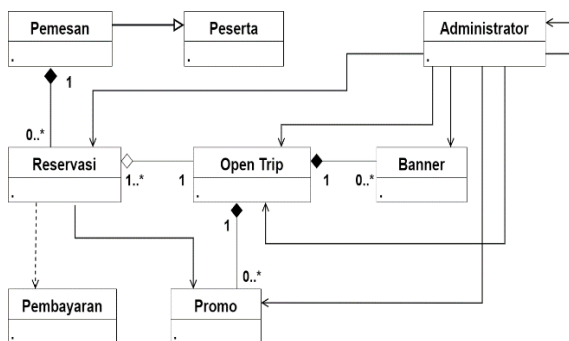
5.1. Perancangan Arsitektur

Pertama, perancangan terhadap arsitektur

sistem yang menggambarkan struktur sistem secara garis besar (abstrak). Struktur sistem disesuaikan dengan struktur sistem yang menggunakan MERN *stack*.

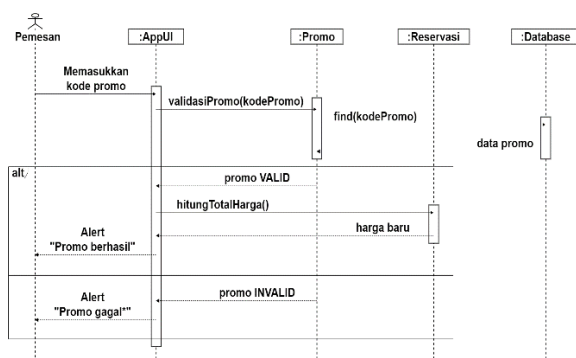
5.2. Perancangan Komponen

Perancangan kedua terkait dengan perancangan komponen sistem yang menggambarkan secara lebih detail struktur sistem terhadap komponen yang menyusunnya. Model yang digunakan adalah *class diagram* dan *sequence diagram*. Gambar 5 berikut ini adalah *class diagram* sebagai representasi objek yang terlibat dalam sistem.



Gambar 5. *Class diagram* aplikasi reservasi paket wisata

Selain itu, perancangan komponen juga dilakukan untuk menunjukkan aktivitas sistem secara dinamis atau interaksi antar objek saat suatu fungsi dijalankan menggunakan *sequence diagram*. Gambar 6 menunjukkan *sequence diagram* dari salah satu fungsi yaitu Pakai Promo.



Gambar 6. *Sequence diagram* aplikasi reservasi paket wisata

5.3. Perancangan Data

Perancangan terhadap data juga dilakukan untuk merepresentasikan struktur data dalam basis data NoSQL dengan menggunakan model

Physical Data Model (PDM) seperti yang ditunjukkan pada Gambar 7 di bawah ini.



Gambar 7. PDM basis data

5.4. Perancangan Antarmuka

Perancangan terakhir dilakukan terhadap antarmuka (*user interface*) sistem dengan menggunakan aplikasi pembuat *prototype* yaitu Marvel. Konsep rancangan antarmuka yang digunakan adalah *low-fidelity* yang menggambarkan secara umum struktur atau susunan antarmuka.

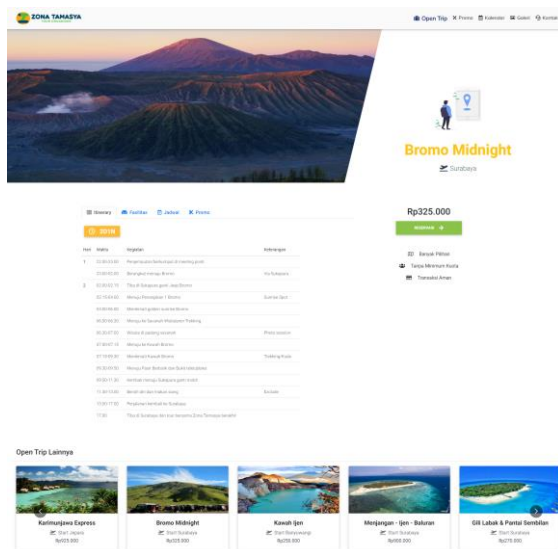
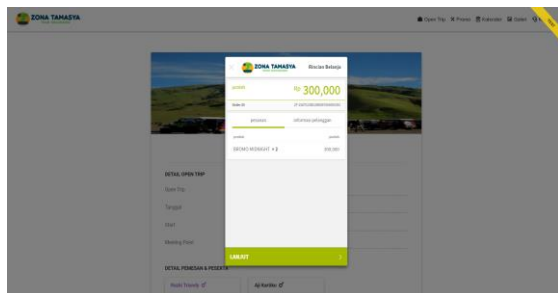
6. IMPLEMENTASI

Implementasi sistem dilakukan menggunakan teknologi yang telah ditentukan yaitu MERN *stack* dengan bantuan *library* lainnya. Hasil implementasi dapat diuraikan sebagai berikut.

6.1. Implementasi Fungsi

Implementasi fungsi mencakup implementasi untuk seluruh kebutuhan fungsional baik dari sisi *front-end* dan *back-end*. Sisi *front-end* diimplementasikan menggunakan ReactJS dengan bantuan *library/tools* lain seperti Bootstrap, Font Awesome, CSS Generator, dll; sedangkan sisi *back-end* diimplementasikan menggunakan Express dan Node.js yang juga dibantu dengan *library* lain.

Fungsi dalam implementasinya dapat ditinjau dari dua sisi, yaitu antarmuka dan kode programnya. Maka dari itu, implementasi fungsi mengacu pada rancangan antarmuka dan rancangan komponen. Contoh hasil implementasi fungsi dapat dilihat pada Gambar 8 yaitu halaman detail *open trip* sebagai implementasi fungsi Cek Detail Informasi *Open Trip* serta pada Gambar 9 yang menunjukkan tampilan jendela pembayaran *online* sebagai bagian implementasi fungsi Reservasi *Open Trip*.

Gambar 8. Implementasi halaman detail *open trip*Gambar 9. Implementasi pembayaran *online*

6.2. Implementasi Basis Data

Basis data diimplementasikan dengan menggunakan MongoDB sesuai dengan hasil perancangan data. Dalam implementasinya, juga digunakan *library* dari MongoDB yaitu Mongoose. Total terdapat lima koleksi (tabel) yang diimplementasikan, yaitu Reservasi, Open Trip, Promo, Banner, dan Admin.

```
{
  "_id": "ObjectId(\"5e79c19e38ebc8136c69a12f\")",
  "tripId": "5e79b68838ebc8136c69a12a",
  "code": "BROMO2020",
  "discount": 120000,
  "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod...",
  "expDate": "2020-05-02",
  "_v": 0
}
```

Gambar 10. Implementasi basis data koleksi Promo

7. PENGUJIAN DAN EVALUASI

Pengujian dilakukan terhadap seluruh kebutuhan yang telah diimplementasikan baik kebutuhan fungsional dan kebutuhan non-fungsional. Untuk kebutuhan fungsional, terdapat dua level pengujian yaitu pengujian unit dan pengujian validasi. Kebutuhan non-fungsional diuji menggunakan bantuan *tools*

tertentu.

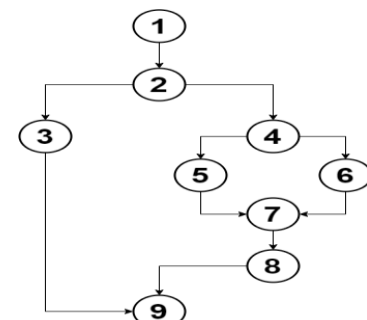
7.1. Pengujian Unit

Pengujian unit dilakukan secara *white-box* dengan metode *basis path testing*. Berikut ini adalah rincian pengujian terhadap salah satu fungsi (*method*), yaitu *hitungTotalHarga()*.

Tabel 2. Pseudocode fungsi *hitungTotalHarga()*

```
function hitungTotalHarga(jenis) {
  int totalHarga, jumlahPeserta, diskon;

  switch (jenis) {
    case "DP":
      totalHarga = jumlahPeserta * hargaDP;
      break;
    case "LUNAS":
      if (promo == null) {
        totalHarga = jumlahPeserta * hargaLUNAS;
      } else {
        totalHarga = jumlahPeserta * hargaLUNAS -
        diskon;
      }
      break;
  }
}
```

Gambar 10. Control flow graph fungsi *hitungTotalHarga()*

- Cyclomatic Complexity ($V(G)$)

$$V(G) = E - N + 2$$

$$= 10 - 9 + 2$$

$$= 3$$
- Jalur Independen
 Jalur 1 : 1-2-3-9
 Jalur 2 : 1-2-4-5-7-8-9
 Jalur 3 : 1-2-4-6-7-8-9

Tabel 2 mendefinisikan *pseudocode* dari alur logika fungsi. *Pseudocode* menjadi acuan untuk membuat *control flow graph* (CFG) seperti yang ditunjukkan pada Gambar 10. Selanjutnya berdasarkan CFG tersebut, dilakukan perhitungan *cyclomatic complexity* serta jalur independen. Hasil perhitungan digunakan untuk menentukan jumlah kasus uji. Dalam fungsi *hitungTotalHarga()* terdapat

sejumlah tiga kasus uji. Hasil pengujian dapat dilihat pada Tabel 3. Dari hasil pengujian tersebut, didapatkan hasil yang valid untuk seluruh kasus uji.

Tabel 3. Hasil pengujian fungsi *hitungTotalHarga()*

<i>Test Scenario</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Pass/Fail</i>
Hitung total harga untuk pembayaran DP (DP: 150.000; pax: 2)	Total harga terhitung sejumlah 300.000	<i>As expected</i>	<i>Pass</i>
Hitung total harga untuk pembayaran LUNAS tanpa promo (LUNAS: 325.000; pax: 2)	Total harga terhitung sejumlah 650.000	<i>As expected</i>	<i>Pass</i>
Hitung total harga untuk pembayaran LUNAS dengan promo (LUNAS: 325.000; pax: 2; diskon: 120.000)	Total harga terhitung sejumlah 530.000	<i>As expected</i>	<i>Pass</i>

7.2. Pengujian Validasi

Pengujian validasi dilakukan terhadap seluruh kebutuhan fungsional dengan menggunakan teknik *black-box*. Hasil pengujian yang diperoleh menunjukkan bahwa seluruh kebutuhan fungsional 100% valid untuk semua kasus uji.

7.3. Pengujian Compatibility

Pengujian *compatibility* dilakukan untuk menguji dua aspek, yaitu *cross-browser testing* dan *responsive web design*. *Cross-browser testing* dilakukan di situs pengujian online Sortsite dengan hasil pengujian yang menunjukkan bahwa aplikasi telah mendukung untuk dijalankan pada beragam *browser* terutama Chrome, Firefox, Edge, dan Opera. Hasil serupa juga didapatkan pada pengujian aspek *responsive design* pada situs Website Responsive Testing Tool yang menunjukkan bahwa desain antarmuka aplikasi telah mendukung tiga segmentasi *device* yaitu *mobile*, *tablet*, dan *desktop*.

7.4. Evaluasi

Berdasarkan seluruh hasil pengujian yang telah didapatkan, dapat dinyatakan bahwa seluruh kebutuhan sistem berhasil diimplementasikan sesuai dengan rancangannya

dan sistem siap untuk dioperasikan untuk keperluan terkait.

8. KESIMPULAN

Kesimpulan yang dapat ditarik pada penelitian ini diantaranya:

1. Hasil analisis kebutuhan mendapatkan sejumlah 3 aktor, 28 kebutuhan fungsional, dan 1 kebutuhan non-fungsional yang kemudian dimodelkan menggunakan *use-case diagram* dan *use-case scenario*.
2. Perancangan dilakukan dalam berbagai level; yaitu arsitektur, komponen, data, dan antarmuka. Perancangan arsitektur dibuat berdasarkan penggunaan MERN *stack*. Untuk perancangan data menggunakan *class diagram* yang mendefinisikan sejumlah delapan kelas serta *sequence diagram*. Selanjutnya perancangan data dibuat menggunakan PDM yang mendefinisikan sejumlah lima koleksi dan perancangan antarmuka dibuat secara *low-fidelity*.
3. Implementasi dilakukan sesuai dengan rancangan menggunakan MERN *stack*. Implementasi *front-end* menggunakan ReactJS, implementasi *back-end* menggunakan Express dan Node.js, serta implementasi basis data menggunakan MongoDB.
4. Hasil pengujian terhadap seluruh kebutuhan sistem baik pengujian unit, pengujian validasi, dan pengujian *compatibility* mendapatkan hasil yang valid.

9. SARAN

Adapun saran yang dapat dipertimbangkan untuk meningkatkan kualitas penelitian selanjutnya antara lain:

1. Penggunaan *state management* React seperti Context atau Redux untuk mengoptimalkan penyimpanan dan penggunaan *state*.
2. Penggunaan GraphQL sebagai protokol komunikasi API untuk meningkatkan efisiensi pemanggilan data.
3. Menambahkan beberapa fitur/konten pada aplikasi seperti artikel (blog), *rating*, testimoni, dan komentar untuk setiap paket wisata, serta fitur untuk mengunduh jadwal paket wisata dan *itinerary*.
4. Menyediakan fungsi untuk mengirimkan

detail informasi keberangkatan trip kepada pemesan meliputi tour guide yang bertugas serta armada kendaraan yang digunakan.

5. Menambahkan fitur bagi administrator untuk dapat menampilkan dan mengeksport laporan statistik reservasi paket wisata setiap bulan.

10. DAFTAR PUSTAKA

- Adhikari, A., 2016. *Full Stack Javascript: Web Application Development With MEAN*. Bachelor's degree. Helsinki Metropolia University of Applied Sciences.
- Andhale, P., Ghadigaonkar, S., Lahoti, K., Patil, K. and Jadhav, S., 2017. Webbased Application. *IJARCCCE*, [online] 6(5), p.685. Tersedia di: <<http://ijarcce.com/upload/2017/may-17/IJARCCCE%20131.pdf>> [Diakses 29 Juni 2020].
- Dissanayake, N. dan Dias, K., 2017. Web-based Applications: Extending the General Perspective of the Service of Web. Dalam: 10th International Research Conference of KDU (KDU-IRC 2017) on Changing Dynamics in the Global Environment: Challenges and Opportunities. [online] University of Colombo School of Computing. Tersedia di: <https://www.researchgate.net/publication/319058851_Web-based_Applications_Extending_the_General_Perspective_of_the_Service_of_Web> [Diakses 18 Oktober 2019].
- Fowler, S. dan Stanwick, V., 2004. *Web Application Design Handbook: Best Practices For Web-Based Software*. [ebook] San Francisco: Morgan Kaufmann. Tersedia di: <<https://epdf.pub/web-application-design-handbook-best-practices-for-web-based-software.html>> [Diakses 18 Oktober 2019].
- Irmawati, D., 2011. PEMANFAATAN E-COMMERCE DALAM DUNIA BISNIS. *Jurnal Ilmiah Orasi Bisnis*, [online] 4, p.95. Tersedia di: <https://www.academia.edu/4911623/PEMANFAATAN_ECOMMERCE_DALAM_DUNIA_BISNIS> [Diakses 29 Juni 2020]
- Löfberg, M. and Molin, P., 2005. *Web Vs. Standalone Application - A Maintenance Application for Business Intelligence*. Master. Blekinge Institute of Technology.
- Parikh, S., 2019. *What Is A Payment Gateway And How Does It Work*. [online] Razorpay. Tersedia di: <<https://razorpay.com/blog/payment-gateway-101/>> [Diakses 9 Januari 2020].
- Prasetyo, A., 2012. PENGARUH KUALITAS PELAYANAN DAN HARGA TERHADAP KEPUASAN PELANGGAN. *Management Analysis Journal*, [online] 1(1), p.2. Tersedia di: <<https://journal.unnes.ac.id/sju/index.php/maj/article/view/497>> [Diakses 29 Juni 2020].
- Subramanian, V., 2017. *Pro MERN Stack: Full Stack Web App Development With Mongo, Express, React, And Node*. [ebook] Bengaluru: Apress. Tersedia di: <<https://ru.ok2.org/book/5242227/3a8764>> [Diakses 9 September 2019].