



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN *BACKEND* APLIKASI
MALUKU EXPLORE DENGAN *NODE.JS*:
STUDI KASUS WISATA MALUKU**

TUGAS AKHIR

MUHAMAD RIYANDI

0110220214

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI
TERPADU NURUL FIKRI
AGUSTUS 2024**



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN *BACKEND* APLIKASI
MALUKU EXPLORE DENGAN *NODE.JS*:
STUDI KASUS WISATA MALUKU**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjanah komputer

STT - NF

MUHAMAD RIYANDI

0110220214

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI
TERPADU NURUL FIKRI
AGUSTUS 2024**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Muhamad Riyandi

NIM : 0110220214

STT - NF Depok, 18 April 2024

Tanda Tangan



Muhamad Riyandi

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Muhamad Riyandi

NIM : 0110220214

Program Studi : Teknik Informatika

Judul Skripsi : Rancang Bangun *Backend* Aplikasi Maluku Explore dengan
Node.JS: Studi Kasus Wisata Maluku

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

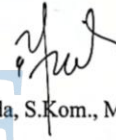
DEWAN PENGUJI

Pembimbing



(Nasrul, S.Pd.I., S.Kom., M.Kom.)

Penguji



(Efrizal Zaida, S.Kom., M.M., M.Kom.)

STT - NF

Ditetapkan di : Depok

Tanggal : 23 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah Subhanahu Wa Ta'ala.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi, S.T., M.M., M.T., selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tiffany Nabarian, S.Kom., M.TI., selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak Zaki Imaduddin, S.T., M.Kom., selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama perkuliahan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Nasrul, S.Pd.I., S.Kom., M.Kom., selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
8. Para teman-teman penulis yang sudah berjuang bersama dan membantu dalam menyelesaikan laporan bersama-sama.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila

terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 18 April 2024



Muhamad Riyandi



STT - NF

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Muhamad Riyandi

NIM : 0110220214

Program Studi : Teknik Informatika

Jenis karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty - Free Right*) atas karya ilmiah saya yang berjudul :

Rancang Bangun *Backend* Aplikasi Maluku Explore Dengan *Node.JS*: Studi Kasus Wisata Maluku

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

STT - NF

Dibuat di : Depok
Pada tanggal : 26 Juni 2024

Yang Menyatakan



(Muhamad Riyandi)

ABSTRAK

Nama : Muhamad Riyandi
NIM : 0110220214
Program Studi : Teknik Informatika
Judul : Rancang Bangun *Backend* Aplikasi Maluku Explore Dengan Menggunakan *Node.JS*: Studi Kasus Wisata Maluku

Maluku, sebuah provinsi kepulauan di Indonesia, memiliki keindahan alam dan kekayaan budaya yang unik, namun kurangnya informasi online yang terintegrasi menghambat promosi pariwisata di wilayah ini. Tugas akhir ini berfokus pada pengembangan *backend* aplikasi Maluku Explore menggunakan *Node.JS*, bertujuan untuk menyediakan sistem informasi wisata yang terstruktur dan mudah diakses. Metode *Extreme Programming* diterapkan dalam pengembangan, sementara evaluasi dilakukan menggunakan *Black Box Testing*. Aplikasi ini dibatasi pada pengembangan *backend* dan hanya mencakup platform *website* tanpa fitur *payment gateway*. Hasil pengujian menunjukkan bahwa aplikasi *backend* berjalan sesuai harapan, menyajikan informasi wisata dengan efisien dan andal. Dengan ini, aplikasi Maluku Explore diharapkan dapat meningkatkan aksesibilitas dan pengalaman wisatawan, serta mempromosikan destinasi wisata Maluku secara lebih luas.

STT - NF

Kata kunci : *backend*, *black box testing*, *Extreme Programming*, Maluku, *Node.Js*.

ABSTRACT

Name : Muhamad Riyandi
NIM : 0110220214
Study Program : Informatics Engineering
Title : Backend Development of Maluku Explore Application Using Node.JS: A Case Study on Maluku Tourism.

Maluku, an archipelagic province in Indonesia, is known for its natural beauty and rich cultural heritage, yet the lack of integrated online information hinders tourism promotion in the region. This final project focuses on developing the backend of the Maluku Explore application using Node.JS, aiming to provide a structured and accessible tourism information system. The development followed the Extreme Programming methodology, with evaluation conducted through Black Box Testing. The scope of the project is limited to backend development and web-based platform without including payment gateway features. The testing results indicate that the backend application operates as expected, efficiently and reliably presenting tourism information. The Maluku Explore application is anticipated to enhance tourist accessibility and experience, promoting Maluku's tourist destinations more effectively.

Key words : backend, black box testing, Extreme Programming, Maluku, Node.JS.

STT - NF

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK	viii
<i>ABSTRACT</i>	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	3
BAB II KAJIAN LITERATUR.....	5
2.1 Rancang Bangun.....	5
2.2 Aplikasi Berbasis <i>Website</i>	5
2.3 Wisata Maluku	5
2.4 Pengembangan <i>Backend</i>	6
2.4.1 <i>Javascript</i>	6
2.4.2 <i>Express.JS</i>	7

2.4.3	<i>Node.JS</i>	7
2.4.4	<i>Application Programming Interface</i>	8
2.4.5	<i>Representational State Transfer</i>	8
2.4.6	<i>Sequelize</i>	9
2.4.7	<i>MarianDB</i>	9
2.4.8	Postman.....	9
2.4.9	XAMPP	10
2.5	<i>Model View Controller (MVC)</i>	10
2.6	<i>Unified Modeling Language</i>	11
2.6.1	<i>Use Case Diagram</i>	11
2.6.2	<i>Class Diagram</i>	12
2.6.3	<i>Activity Diagram</i>	12
2.7	Pengujian Perangkat Lunak.....	13
2.7.1	<i>Black Box Testing</i>	13
2.8	Metodologi Pengembangan Perangkat Lunak.....	14
2.9	Penelitian Terkait	15
2.10	Posisi Penelitian.....	18
BAB III METODOLOGI PENELITIAN		22
3.1	Tahapan Penelitian	22
3.1.1	Studi Penelitian	23
3.1.2	Analisis Perancangan	23
3.1.3	Implementasi	23
3.1.4	Pengujian.....	23
3.1.5	Kesimpulan	24
3.2	Rancangan Penelitian	24

3.2.1	Jenis Penelitian.....	24
3.2.2	Metode Analisis	25
3.2.3	Metode Pengumpulan Data.....	25
3.2.4	Metode Implementasi dan Evaluasi	26
3.3	Metode Pengujian.....	26
3.4	Lingkungan Pengembangan	31
3.4.1	Lokasi Penelitian.....	32
3.4.2	Alat dan Bahan.....	32
BAB IV IMPLEMENTASI DAN EVALUASI.....		33
4.1	Analisis dan Perancangan Sistem.....	33
4.1.1	<i>User Stories</i>	33
4.1.2	Identifikasi Aktor	34
4.1.3	<i>Product Backlog</i>	35
4.1.4	<i>Use Case Diagram</i>	36
4.1.5	<i>Class Diagram</i>	38
4.1.6	<i>Activity Diagram</i>	38
4.2	Implementasi Sistem Maluku Explore	48
4.2.1	Antarmuka Kelola Data Pengguna.....	48
4.2.2	Antarmuka Kelola Data Blog.....	53
4.2.3	Antarmuka Kelola Data <i>Tour</i>	57
4.2.4	Antarmuka Kelola Data Order	62
4.3	Pengujian Sistem	66
4.3.1	Black Box Testing.....	66
BAB V KESIMPULAN DAN SARAN.....		72
5.1	Kesimpulan.....	72

5.2	Saran.....	73
DAFTAR PUSTAKA		75



STT - NF

DAFTAR GAMBAR

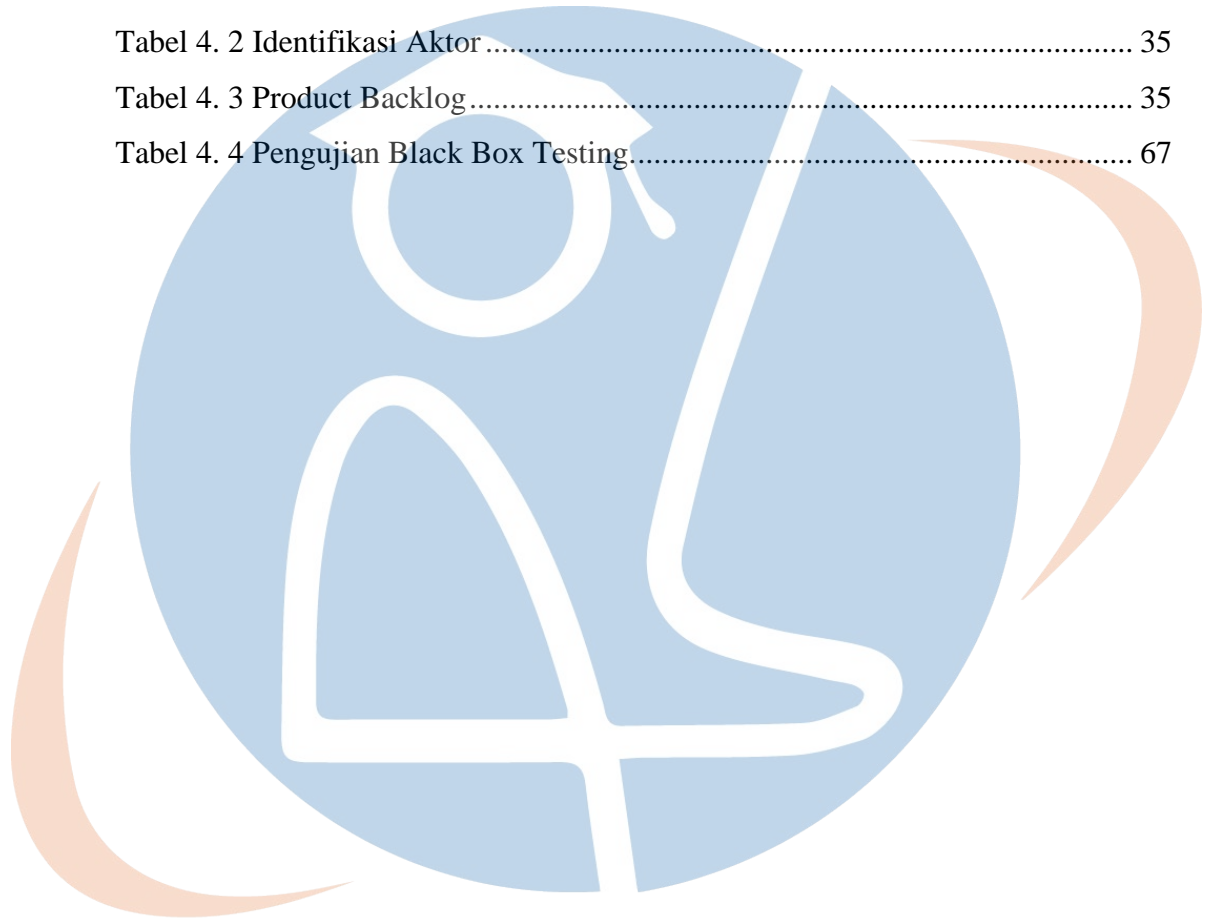
Gambar 2. 1 Tahapan Alur Metode Extreme Programming (XP) [28].....	15
Gambar 3. 1 Tahapan Penelitian	22
Gambar 4. 1 Use Case Diagram.....	37
Gambar 4. 2 Class Diagram	38
Gambar 4. 3 Activity Diagram Mengelola Data Staff	39
Gambar 4. 4 Activity Diagram Mengelola Data Pengunjung	41
Gambar 4. 5 Activity Diagram Mengelola Data Blog	42
Gambar 4. 6 Activity Diagram Mengelola Data Tour	44
Gambar 4. 7 Activity Diagram Mengelola Data Order.....	45
Gambar 4. 8 Activity Diagram Pengguna	47
Gambar 4. 9 Tampilan Input Data User.	48
Gambar 4. 10 Tampilan Melihat Data User Berdasarkan Id.....	49
Gambar 4. 11 Tampilan Melihat Seluruh Data User.....	50
Gambar 4. 12 Tampilan Log-in User	51
Gambar 4. 13 Tampilan Log-out User	51
Gambar 4. 14 Tampilan Mengubah Data User.	52
Gambar 4. 15 Tampilan Menghapus Data User.....	52
Gambar 4. 16 Tampilan Input Data Blog.....	53
Gambar 4. 17 Tampilan Melihat Data Blog Berdasarkan Id.	54
Gambar 4. 18 Tampilan Menampilkan Seluruh Data Blog.....	55
Gambar 4. 19 Tampilan Mengubah Data Blog	56
Gambar 4. 20 Tampilan Menghapus Data Blog.....	57
Gambar 4. 21 Tampilan Input Data Tour.....	58
Gambar 4. 22 Tampilan Menampilkan Data Tour Berdasarkan Id.....	59
Gambar 4. 23 Tampilan Menampilkan Seluruh Data Tour.....	60
Gambar 4. 24 Tampilan Mengubah Data Tour.	61
Gambar 4. 25 Tampilan Menghapus Data Tour.....	61
Gambar 4. 26 Tampilan input Data Order.	62
Gambar 4. 27 Tampilan Menampilkan Data Orde Berdasarkan id.....	63

Gambar 4. 28 Tampilan Menampilkan Seluruh Data Order..... 64
Gambar 4. 29 Tampilan Mengubah Data Order..... 65
Gambar 4. 30 Tampilan Menghapus Data Order..... 66



DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	16
Tabel 2. 2 Posisi Penelitian	18
Tabel 3. 1 Black Box Testing.....	27
Tabel 4. 1 User Stories	33
Tabel 4. 2 Identifikasi Aktor	35
Tabel 4. 3 Product Backlog.....	35
Tabel 4. 4 Pengujian Black Box Testing.....	67



STT - NF

BAB I

PENDAHULUAN

1.1 Latar belakang

Maluku merupakan wilayah kepulauan yang terdiri dari pulau besar dan pulau kecil yang tersebar dari wilayah timur mendekati Provinsi Papua barat dan di sebelah selatan yang mendekati Provinsi Nusa Tenggara Timur. Maluku, merupakan salah satu dari kepulauan di Indonesia yang memiliki keindahan alam serta keanekaragaman budaya yang berada di wilayah tersebut. Maluku, memiliki destinasi wisata yang sangat indah dan menawan, dari keindahan pantai yang tersembunyi dengan pasir putih yang bersih, hingga gunung yang menjulang tinggi.

Destinasi wisata di Maluku menawarkan potensi wisata yang luar biasa dengan keindahan alam dan kekayaan budaya yang unik. Namun, keterbatasan akses informasi secara *online* dan kurangnya pengembangan sistem informasi wisata yang terintegrasi telah menjadi hambatan dalam mempromosikan dan meningkatkan daya tarik pariwisata di wilayah ini. Oleh karena itu, pengembangan sebuah aplikasi berbasis *web* yang dapat menyediakan informasi yang lengkap dan terstruktur tentang objek wisata di Maluku menjadi sebuah kebutuhan yang mendesak.

Dengan demikian, pengembangan *backend* aplikasi Maluku Explore dengan menggunakan *Node.JS* sebagai platform utama menjadi langkah yang tepat dalam mengatasi tantangan tersebut. Melalui penyediaan sistem informasi wisata yang terintegrasi dan dapat mudah diakses, diharapkan aplikasi ini dapat memenuhi kebutuhan pengguna yang ingin menjelajahi keindahan serta kekayaan budaya yang berada di wilayah Maluku. Dengan memanfaatkannya teknologi dalam menyediakan informasi wisata yang akurat serta terpercaya, aplikasi ini akan memberikan kontribusi positif dalam mempromosikan wisata di wilayah Maluku serta meningkatkan pengalaman wisatawan dalam merencanakan serta melakukan perjalanan mereka. Oleh karena itu, pengembangan *backend* aplikasi Maluku Explore menjadi langkah yang strategis dalam mendukung pertumbuhan sektor pariwisata dan memperluas jangkauan destinasi wisata di wilayah Maluku.

1.2 Rumusan Masalah

Dengan merujuk pada informasi latar belakang yang telah disajikan, perumusan masalah yang tepat dapat dirinci sebagai berikut:

1. Bagaimana rancang bangun *backend* aplikasi Maluku Explore dengan *Node.JS* untuk menyajikan informasi wisata di Maluku sesuai dengan kebutuhan pengguna?
2. Apakah aplikasi Maluku Explore yang dirancang dan dibangun dapat mencapai kemudahan dalam eksplorasi wisata di wilayah Maluku?

1.3 Tujuan Penelitian

Dalam penelitian ini, beberapa target yang ingin dicapai dapat dilaksanakan sebagai berikut :

1. Merancang dan membangun *backend* aplikasi Maluku Explore dengan *Node.JS* dalam menyajikan informasi wisata Maluku sesuai dengan kebutuhan pengguna. Hal ini termasuk mengimplementasikan fitur-fitur yang relevan, seperti pengelolaan data destinasi wisata, integrasi *API*, dan penyediaan informasi yang akurat dan terstruktur.
2. Merancang dan membangun aplikasi Maluku Explore dengan mempertimbangkan kesesuaian fitur dan kebutuhan wisatawan untuk mempermudah *explorasi* wisata Maluku.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Meningkatkan aksesibilitas informasi wisata Maluku dengan menyediakan antarmuka yang dapat diakses secara *online*.
2. Meningkatkan pengalaman pengguna dalam mencari informasi wisata Maluku.
3. Dapat mendukung pertumbuhan sektor pariwisata di wilayah Maluku.

1.5 Batasan Masalah

Penelitian ini menekankan kepada pengembangan *backend* menggunakan *Node.JS* untuk *website* Maluku Explore. Berikut merupakan batasan dari lingkup penelitian :

1. Fokus pada pengembangan dan evaluasi *backend* yang menyajikan fitur seperti *login*, *logout*, pencarian objek wisata, dan pengelolaan konten wisata. Pengelolaan konten akan dilakukan oleh admin dan staf, mencakup penambahan, pengubahan, dan penghapusan data objek wisata.
2. Aplikasi dikembangkan menggunakan *Express.js* versi 4.18.2 sebagai kerangka kerja, *Node.js* versi 20.12.2 sebagai *runtime server*, dan *MariaDB* versi 10.4.27 sebagai *basis data*.
3. Pengembangan aplikasi tidak meliputi fitur *payment gateway*.
4. Informasi yang disajikan dalam aplikasi hanya mencakup objek wisata yang berada di wilayah Maluku.

1.6 Sistematika Penulisan

Sistematika penulisan mempermudah pembaca dalam memahami isi dari penelitian yang telah dikembangkan. Sistematika yang diterapkan dalam penelitian ini yaitu;

BAB I PENDAHULUAN

Pada bab ini memberikan penjelasan tentang latar belakang penelitian, rumusan masalah, tujuan dan manfaat penelitian yang diharapkan dapat menjawab dari rumusan masalah, batasan masalah penelitian, dan pembahasan sistematika penulisan.

BAB II KAJIAN LITELATUR

Pada bab ini menjabarkan teori-teori pendukung dalam fitur pengembangan, metode pengembangan, tinjauan litelatur, serta posisi penelitian ini dalam kerangka konsep yang lebih luas.

BAB III METODE PENELITIAN

Bab ini akan menjabarkan mengenai rancangan penelitian yang digunakan, alat, dan teknik yang diterapkan, serta langkah-langkah dalam merancang solusi pengembangan *backend* aplikasi.

BAB IV IMPLEMENTASI DAN EVALUASI

Pada bab ini, akan difokuskan pada pelaksanaan proses pengembangan *backend* aplikasi yang telah diimplementasikan, serta melakukan evaluasi kepada hasil yang berhasil diperoleh dalam proses pengembangan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi rangkuman temuan-temuan dari penelitian, menjelaskan kesimpulan yang telah diperoleh dalam penelitian, dan memberikan saran yang relevan untuk penelitian yang mungkin akan dilakukan di masa yang akan datang.



STT - NF

BAB II

KAJIAN LITERATUR

2.1 Rancang Bangun

Rancang bangun dalam konteks pengembangan perangkat lunak adalah proses yang berfokus pada pembuatan dan penyempurnaan aplikasi, baik yang baru maupun yang sudah ada, melalui penerapan berbagai komponen yang ditentukan dari hasil analisis sistem. Tujuan utamanya adalah menghasilkan aplikasi berkualitas yang sesuai dengan kebutuhan spesifik. Proses ini melibatkan beberapa langkah penting seperti pemilihan teknologi yang tepat, pemodelan arsitektur yang jelas, pengujian yang ketat, dan optimasi berkelanjutan. Dengan pendekatan yang terstruktur, rancang bangun dapat memastikan aplikasi yang dikembangkan tidak hanya efektif dan efisien tetapi juga mampu memenuhi harapan pengguna secara optimal [1].

2.2 Aplikasi Berbasis Website

Website merupakan kumpulan halaman yang memuat berbagai informasi dan biasanya diakses melalui jaringan internet. *Website* memungkinkan siapa saja dari berbagai lokasi dan waktu untuk mengaksesnya, selama mereka terhubung ke jaringan internet. Secara teknis, *website* terdiri dari beberapa halaman yang saling terkait dan tergabung dalam satu *domain* atau *subdomain* tertentu. Halaman-halaman ini, sering disebut sebagai "*page*," dapat mencakup teks, gambar, video, dan berbagai jenis konten lainnya yang disusun secara sistematis untuk memberikan informasi atau layanan kepada pengguna. *Domain* atau *subdomain* berfungsi sebagai alamat unik yang mengidentifikasi *website* tersebut di internet, memungkinkan pengguna untuk menemukannya dengan mudah melalui browser mereka [2].

2.3 Wisata Maluku

Di era digital saat ini, kebutuhan akan akses yang mudah dan cepat ke berbagai layanan telah menjadi prioritas bagi banyak orang, terutama dalam

merencanakan perjalanan wisata. Fransiskus Xaverius Teguh (2019) dalam acara FGD bertema " Teknologi dan Kepariwisata ". Menjelaskan Teknologi dapat mempengaruhi dan membentuk cara seseorang dalam melakukan kegiatan wisata, mulai dari perencanaan perjalanan, saat dalam perjalanan, dengan saat kembali dari perjalanan [3].

2.4 Pengembangan *Backend*

Pengembangan *Backend* merupakan proses perancangan sistem yang berjalan dibelakang layar, baik pada situs *website* maupun aplikasi. Dalam pengembangan *backend* bahasa pemrograman umum yang digunakan seperti *Javascript*, *PHP*, *Ruby*, *Python*, dan lain sebagainya. Hal tersebut merupakan komponen yang dioperasikan dalam balik layar dan tidak terlihat secara langsung oleh pengguna. Namun, dalam sebuah perancangan situs *website* maupun aplikasi, peran dalam pengembangan *backend* memberikan peran pada proses yang terjadi terutama dalam sebuah pemrosesan serta manipulasi sebuah data yang akan disajikan [4]. *Backend* membantu dalam mengkomunikasikan informasi dari *database* kepada pengguna melalui antarmuka pengguna (*UI*) yang disediakan oleh bagian *Frontend*. Hal yang akan dikembangkan pada bagian *backend* antara lain metode *HTTP* (*post*, *get*, *put*, dan *delete*), serta *API*.

2.4.1 *Javascript*

Javascript merupakan bahasa tingkat tinggi berbentuk *script* yang berjalan pada dokumen *HTML*, bahasa pemrograman *javascript* memiliki peran dalam memperbaiki tampilan pada sistem aplikasi berbasis *web*. Beberapa ciri khas bahasa pemrograman *Javascript* bisa melakukan penekanan pada sisi klien, berorientasi objek, dan berkarakteristik *strictly typed*. Dalam mengimplementasikan bahasa pemrograman *Javascript* dapat menggunakan aplikasi atau alat standar seperti Nodepad++, Adobe Dreamweaver, dan NetBeans yang digunakan sebagai *teks editor*. Selain itu bahasa pemrograman *Javascript* bisa dijalankan pada *web browser*

seperti Mozilla, Firefox, Google Chrome, Internet Explorer, OperaMini, dan juga Safari yang telah mendukung penggunaan bahasa pemrograman *Javascript* [5].

2.4.2 *Express.JS*

Express.JS adalah sebuah *framework* bersifat *open source*, yang bertujuan untuk menyederhanakan serta mempercepat pengembangan aplikasi *web* berbasis *API* [6]. *Express.js* juga merupakan salah satu dari *framework* yang populer dan memiliki reputasi yang baik di kalangan pengembang, kemampuannya dalam mengintegrasikan fungsi-fungsi yang kuat dan fleksibel dalam pengembangan aplikasi *web*, serta dukungan yang luas dari komunitas pengembang menjadikannya pilihan yang menarik.

Express.JS merupakan sebuah *framework* yang sering digunakan sebagai alat bantu dalam mengembangkan aplikasi *web* di lingkungan *Node.JS*, hal ini dikarenakan *Express.JS* menawarkan pendekatan yang disederhanakan dan minimalis dalam mengembangkan aplikasi *web*, dengan struktur yang dapat disesuaikan dan modular, menjadikan *Express.JS* pilihan utama bagi pengembang *web* dalam membangun aplikasi yang efisien dan tangguh [7].

2.4.3 *Node.JS*

Node.JS merupakan lingkungan *runtime* yang memiliki sifat *open-source* dan juga merupakan platform yang digunakan dalam membangun aplikasi *web* di sisi *server*. Aplikasi yang dikembangkan menggunakan *Node.JS* dituliskan ke dalam bahasa pemrograman *Javascript* dan memiliki kemampuan dalam menjalankan berbagai *operation system*. *Node.JS* dapat dibangun dengan menggunakan arsitektur *event-driven* dengan menggunakan *non-blocking* dalam *API Input/Output* yang dapat menangani banyak permintaan secara bersamaan dengan efisien, memberikan *respons* dengan cepat terhadap permintaan yang dilakukan oleh pengguna, penggunaan sumber daya komputasi menjadi lebih efisien dapat meningkatkan *throughput* secara keseluruhan, dan aplikasi yang dirancang akan lebih fleksibel serta mudah di *maintenance* sehingga mempermudah

dalam menambahkan, melihat, mengubah, atau menghapus fitur di dalam aplikasi tanpa mengganggu fungsi lainnya [8].

2.4.4 *Application Programming Interface*

Application Programming Interface (API) menjadi alat utama dalam mengimplementasikan pertukaran data antara sistem yang berbeda. Secara spesifik *API* memfasilitasi komunikasi antar dua aplikasi perangkat lunak [9]. Komponen *API* mencakup berbagai element yang memungkinkan pengembangan membangun aplikasi menjadi lebih efisien. Penggunaan *API* bertujuan dalam mempercepat proses pengembangan dengan menyediakan fungsi yang sudah ada sehingga pengembang tidak perlu membuat fitur dari awal. Contoh dalam penerapan *API* adalah integrasi dengan *gateway* pembayaran, di mana penggunaan *API* mempercepat pengembangan fitur yang kompleks dan menghemat waktu [10].

2.4.5 *Representational State Transfer*

Representational State Transfer (REST) merupakan paradigma arsitektur yang diterapkan dalam perancangan sistem terdistribusi, termasuk ke dalam pembangunan aplikasi *web*. *REST* memiliki konsep dengan menekankan pada representasi sederhana dari sumber daya sistem dan interaksi berbasis sederhana pada standar operasi *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE*. *REST* memungkinkan pengembangan antarmuka lebih mudah dipahami dan dapat diakses secara universal, dengan memisahkan sisi *client* dan *server* serta memungkinkan komunikasi yang terkait dengan status dapat disimpan. Dengan mengikuti prinsip *REST* dalam rancang bangun aplikasi, pengembang dapat membuat sistem yang *skalabel*, mudah dikelola, dan dipelihara. Oleh karena itu. *REST* sangat penting dalam merancang serta mengembangkan aplikasi yang memenuhi kebutuhan pengguna dengan efisien dan efektif [11].

2.4.6 *Sequelize*

Sequelize merupakan salah satu alat yang digunakan dalam lingkup pengembangan perangkat lunak berbasis *Node.JS*, dengan menggunakan data relasional yang berfungsi sebagai *Object Relational Mapping (ORM)*, dengan memanfaatkan *promises* dalam menangani operasi *asinkron* dengan efisien. *Sequelize* mendukung berbagai dialek basis data seperti *PostgresSql*, *MySQL*, *MariaDB*, *SQLite*, dan *MSSQL*, dengan menawarkan fitur lanjutan seperti hubungan antar table, replika baca, dan transaksi yang handal. Dengan demikian, *sequelize* dapat memberikan kemudahan dalam pengembangan untuk berinteraksi dengan basis data *relasional* menggunakan objek *Javascript*, dengan memungkinkan abstraksi yang tinggi dan pembelajaran yang mudah bagi pengembang yang sudah terbiasa dengan bahasa tersebut [12].

2.4.7 *MariaDB*

MariaDB adalah sebuah sistem manajemen basis data *relasional* yang merupakan pengembangan dari *MySQL*. Dikembangkan oleh komunitas pengembang yang sebelumnya berperan Dalam pengembangan *MySQL*, *MariaDB* menawarkan kompatibilitas dan peningkatan performa yang lebih baik. Selain itu, *MariaDB* dikenal karena fitur-fitur canggih seperti *engine* penyimpanan yang bervariasi, peningkatan kecepatan *query*, dan dukungan untuk bahasa pemrograman yang lebih luas. *MariaDB* juga berfokus pada keamanan, skalabilitas, dan stabilitas, menjadikannya pilihan populer bagi banyak organisasi yang memerlukan solusi database yang handal dan *open-source* [13].

2.4.8 *Postman*

Postman merupakan aplikasi *REST* klien yang bisa diunduh dan juga tersedia dalam bentuk aplikasi web, dari awal postman dirancang sebagai alat dalam membantu untuk melakukan sebuah pengujian terhadap *API*, akan tetapi saat ini postman menjadi sebuah platform kuat untuk pengembangan *API*, dengan bersajikan berbagai fitur yang dapat mendukung baik, dalam pembangunan maupun

penggunaan *API*. Postman populer di kalangan pengembang aplikasi karena kemudahan dalam penggunaannya serta fleksibilitas. Aplikasi postman dilengkapi dengan alat dokumentasi yang komprehensif yang memungkinkan kolaborasi dengan rekan tim atau komunitas yang lebih luas, serta mempermudah proses dalam iterasi proyek dan berbagai informasi [14].

2.4.9 XAMPP

XAMPP merupakan perangkat lunak *all-in-one* yang menyediakan pengembangan lengkap dalam membangun dan menguji aplikasi *web* dan juga bersifat *open source*. XAMPP juga dilengkapi dengan *server database MySQL* yang dapat digunakan oleh pengembang aplikasi *web* dalam mengembangkan dan menjalankan berbagai jenis aplikasi *web* secara lokal di komputer sebelum mempublikasinya secara *online* [15].

2.5 Model View Controller (MVC)

Model View Controller (MVC) merupakan pola desain pada perangkat lunak yang dibangun dengan memisahkan pengembangan aplikasi dengan keterhubungan tiga jenis komponen utama yang dibangun dalam aplikasi: *Model*, *View*, dan *Controller*, yang seringkali dengan berfokus pada paradigma *Object-Oriented Programming (OOP)* [16]. *MVC* adalah sebuah kerangka kerja dalam membangun aplikasi *web* menggunakan desain *MVC*. Hal tersebut merupakan arsitektur yang penting dalam mengembangkan perangkat lunak saat ini. Arsitektur tersebut dapat secara otomatis mengelola kode dan membantu programmer dalam mengembangkan aplikasi *web* dengan baik [17]. Berikut cara *Model-View-Controller* berperan dalam pengembangan perangkat lunak :

1. *Model* adalah objek yang mewakili data atau proses seperti basis data, dan proses dalam *machines*. Seperti, logika aplikasi adalah model.
2. *View* merupakan bentuk dari visualisasi status model atau komponen antarmuka pengguna (*UI*).
3. *Controller* merupakan fasilitas dalam mengubah sebuah status model.

2.6 *Unified Modeling Language*

Unified Modeling Language (UML) memegang peran penting dalam pengembangan sistem perangkat lunak berbasis objek yang sering digunakan dalam industri perangkat lunak untuk merancang atau memodelkan sistem pada perangkat lunak [18]. *UML* tidak hanya mengatur penulisan dalam *blueprint* sistem, akan tetapi juga melibatkan konsep proses bisnis, penulisan kelas dalam bahasa pemrograman yang spesifik, skema *database*, dan kebutuhan perangkat penting lainnya dalam pengembangan perangkat lunak [19]. *UML* juga dapat digunakan dalam mendefinisikan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek [20]. Oleh karena itu, penggunaan *UML* teramat sangat penting dalam memfasilitasi pengembangan sistem perangkat lunak berbasis *web* yang kompleks dan berorientasi objek. Adapun gambaran *UML* yang akan digunakan pada penelitian ini diantaranya yaitu:

2.6.1 *Use Case Diagram*

Use case diagram adalah jenis pemodelan yang digunakan untuk menggambarkan perilaku sistem yang sedang dirancang. Diagram ini mengilustrasikan interaksi antara satu atau lebih aktor dengan sistem, membantu dalam memahami fungsi-fungsi yang ada dalam sistem serta pihak-pihak yang berhak mengakses fungsi-fungsi tersebut. Dengan fokus pada perspektif pengguna, *use case diagram* menyajikan fungsi sistem secara sederhana dan jelas. Secara umum, *use case diagram* merupakan bagian dari *Unified Modeling Language (UML)* yang bertujuan untuk menyediakan representasi visual dari berbagai kemungkinan skenario yang dapat terjadi dalam interaksi antara aktor dan sistem. Aktor dalam *use case diagram* dapat berupa pengguna manusia atau sistem lain yang berinteraksi dengan sistem utama. Dengan memetakan skenario penggunaan ini, pengembang dapat lebih mudah memahami kebutuhan pengguna dan memastikan bahwa semua aspek penting dari sistem diperhitungkan dalam desain dan implementasinya [21].

2.6.2 *Class Diagram*

Class diagram adalah cetak biru atau spesifikasi dalam pemrograman berorientasi objek (*object-oriented programming*) yang, ketika diinstansiasi, menghasilkan objek. *Class Diagram* merupakan elemen inti dalam desain dan pengembangan berbasis objek. *Class Diagram* mendefinisikan atribut seperti properti yang menggambarkan keadaan suatu sistem serta metode atau fungsi yang memungkinkan manipulasi terhadap keadaan tersebut. Dalam konsep pemrograman berorientasi objek (*OOP*), kelas bertindak sebagai kerangka kerja yang mendefinisikan struktur dan perilaku objek. Atribut dalam kelas mewakili data atau informasi yang disimpan dalam objek, sementara metode adalah fungsi yang dapat dijalankan untuk memodifikasi atau mengakses data tersebut. Dengan menggabungkan atribut dan metode dalam satu entitas, kelas memungkinkan pemrogram untuk membuat objek yang memiliki data dan perilaku tertentu, memfasilitasi enkapsulasi, abstraksi, dan pemrograman modular. Objek yang dibuat dari kelas dapat digunakan kembali dan diintegrasikan ke dalam sistem yang lebih besar, mendukung prinsip *OOP* seperti pewarisan, polimorfisme, dan enkapsulasi [22].

2.6.3 *Activity Diagram*

Activity diagram adalah jenis diagram *UML (Unified Modeling Language)* yang digunakan untuk memodelkan alur kerja (*workflow*) atau rangkaian kegiatan aktivitas dalam suatu sistem, perangkat lunak, atau proses bisnis. Diagram ini menggambarkan bagaimana kegiatan-kegiatan dalam sistem tersebut saling berhubungan dan mengalir dari satu langkah ke langkah berikutnya, baik secara berurutan maupun paralel [23]. Diagram aktivitas fokus pada aktivitas yang dilakukan oleh sistem, bukan oleh aktor eksternal, sehingga memberikan pandangan yang jelas mengenai bagaimana sistem beroperasi untuk mencapai tujuan tertentu. Dalam pengembangan perangkat lunak, *activity diagram* juga dapat digunakan untuk memetakan urutan tampilan atau interaksi dalam *user interface* serta menggambarkan rancangan menu yang muncul pada aplikasi [24].

Setiap aktivitas dalam *activity diagram* dianggap sebagai langkah atau state yang dapat memicu transisi ke langkah berikutnya. Dengan demikian, diagram ini sangat berguna dalam mendefinisikan logika bisnis yang kompleks, menggambarkan proses paralel, dan memastikan bahwa semua kemungkinan skenario alur kerja telah dipertimbangkan. Elemen-elemen utama dalam *activity diagram* meliputi aktivitas, *transisi*, *decision nodes*, *forks*, dan *joins*, yang secara kolektif membantu merancang sistem yang efisien dan responsif terhadap berbagai kondisi.

2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan proses menjalankan sebuah program aplikasi dengan bertujuan untuk menemukan masalah yang mungkin terjadi. Sebuah tes dianggap berhasil apabila mampu mengidentifikasi kesalahan dari yang tidak terdeteksi sebelumnya. Dalam kesalahan perangkat lunak dapat mengganggu proses implementasi perangkat lunak tersebut, memungkinkan keperluan dalam perbaikan yang signifikan jika terjadi begitu banyak masalah. Melalui pengujian perangkat lunak diharapkan bahwa, kesalahan dan kecacatan dalam sebuah aplikasi dapat diminimalkan, sehingga meningkatkan kualitas aplikasi tersebut [25].

2.7.1 Black Box Testing

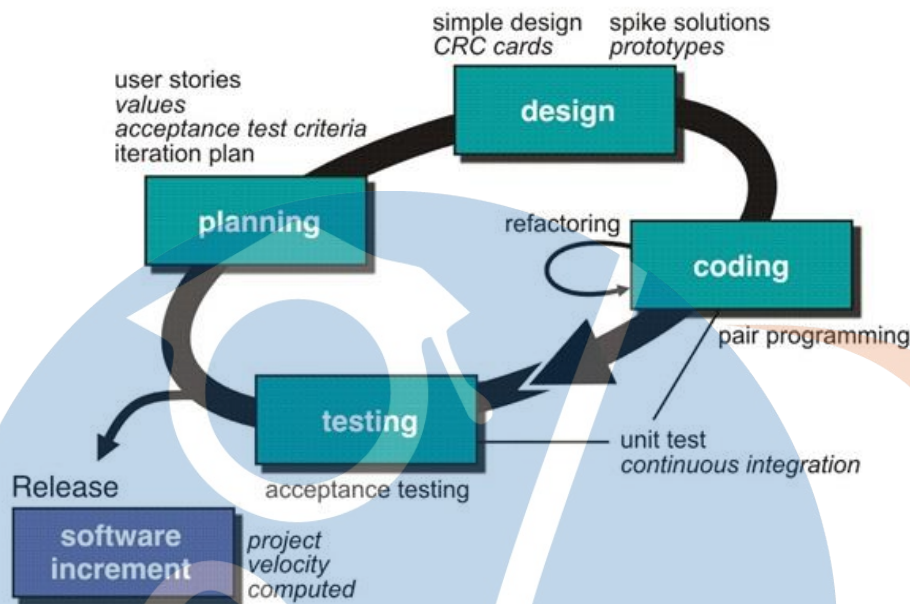
Black Box Testing adalah sebuah metode pengujian dengan melakukan pengamatan hasil dari eksekusi perangkat lunak dengan melalui data uji dan evaluasi fungsionalitas tanpa memperhatikan detail internal atau struktur kode. Fokus dalam metode pengujian ini adalah untuk memastikan bahwa aplikasi yang beroperasi sesuai dengan yang diharapkan dan dapat memenuhi fungsional yang telah ditentukan. Metode *Black Bock Testing* sangat penting digunakan dalam memastikan bahwa perangkat lunak dapat berfungsi secara benar dan konsisten dalam berbagai situasi pengguna [26]. Adapun manfaat *black box testing* diantaranya:

1. Independensi: Memisahkan pengujian dari pengembangan, membantu mengidentifikasi kesalahan yang mungkin terlewat oleh pengembang.
2. Sempel dan Efisien: Pengujian dapat dilakukan tanpa perlu memahami kode, memfasilitasi pengujian cepat oleh tim *QA (Quality Assurance)*.
3. *User-centric*: Menguji aplikasi dari sudut pandang pengguna, memastikan bahwa aplikasi berfungsi seperti yang diharapkan dalam skenario penggunaan nyata.
4. Cakupan yang Luas: Dapat digunakan untuk menguji aplikasi dari awal hingga akhir, mencakup semua aspek dari antarmuka pengguna hingga *response* sistem terhadap *input*.

2.8 Metodologi Pengembangan Perangkat Lunak

Extreme Programming (XP) adalah metode pengembangan aplikasi yang bertujuan dalam meningkatkan kualitas aplikasi serta responsif terhadap perubahan dan kebutuhan pelanggan. Pengembangan ini berfokus kepada peningkatan produktivitas dalam menghasilkan aplikasi yang berkualitas. *XP* memiliki beberapa tahapan penting, seperti perencanaan (*Planning*) yang bertujuan dalam mencangkup pemahaman pada kriteria pengguna dan perencanaan pengembangan, desain (*Designing*) meliputi perancangan *prototype* dan tampilan, pengkodean (*Coding*) mencangkup pengintergrasi, dan tahap terakhir pengujian (*Testing*). Dengan menerapkan metode *XP*, pengembang dapat bekerja secara efisien dan adaptif terhadap perubahan yang akan terjadi selama proses pengembangan aplikasi [27].

STT - NF



Gambar 2. 1 Tahapan Alur Metode Extreme Programming (XP) [28].

Pada Gambar 2.1 Tahapan Alur Metode *Extreme Programming* (XP) terlihat memiliki beberapa tahapan seperti *Planning* (perencanaan), *Design* (desain), *Coding* (pengkodean), dan *Testing* (pengujian) yang berjalan dalam perulangan yang singkat. Dengan tim yang akan berfokus kepada pengiriman perangkat lunak berkualitas tinggi, responsif dengan perubahan, dan penyesuaian kebutuhan pelanggan.

2.9 Penelitian Terkait

Pada penelitian ini, kajian literatur dilakukan dalam mengetahui penelitian terdahulu yang berkaitan dengan isu-isu yang menjadi fokus penelitian saat ini. Langkah tersebut bertujuan untuk menilai lebih jauh kebutuhan dalam pemecahan masalah yang sedang diteliti, dan memahami posisi dalam penelitian yang telah dilakukan sebelumnya. Informasi penelitian terkait dapat ditemukan dalam tabel yang disajikan, sebagai berikut :

Tabel 2. 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Lisa A., Muhammad N.D.S., & Debby A., 2020	Sistem Informasi Akademik dan Administrasi dengan Metode <i>Extreme Programming</i> pada Lembaga Kursus dan Pelatihan	Sistem Informasi Akademik dan Administrasi	Calon tenaga kerja Indonesia	Aplikasi kursus dan pelatihan
2	Rizqy E.P.R, Umar A.A, & Burhanuddin D., 2023	Impelementasi <i>REST API</i> pada Pengembangan Aplikasi <i>Backend</i> untuk <i>Platfrom</i> Kursus <i>Online</i> (Growup)	<i>Online Learning</i>	Masyaraka t usia produktif	<i>Platfrom</i> berbasis <i>website</i>
3	Hasanuddin, Hari A., & Budi H., 2022	Rancang Bangun <i>REST API</i> Aplikasi <i>WESHARE</i> Sebagai upaya Mempermudah Palayanan Donasi Kemanusiaan	Donasi kemanusiaan	Transaksi Donasi kemanusiaan	Sistem berbasis API
4	Nurhayati, Eli., & Aggusalim., 2023	Rancang Bangun <i>Back-end API</i> pada Aplikasi <i>Mobile</i> AyamHub Menggunakan	Aplikasi <i>mobile</i> AyamHub	Pengemba ngan <i>Backend</i> Mengguna	Sistem aplikasi berbasis mobile

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
		<i>Framework Node JS Express</i>		kan Node.js	
5	Mochamad Afif Solahudin	RANCANG BANGUN SISTEM INFORMASI <i>STAYCATION</i> BERBASIS WEB DENGAN IMPLEMENTASI TEKNOLOGI <i>MERN STACK</i>	<i>Booking hotel</i>	Rancang bangun aplikasi <i>booking</i> penginapan menggunakan teknologi <i>MERN</i>	<i>Website Staycation</i>
6	Muhamad Riyandi, 2024	Rancang Bangun <i>Backend</i> Aplikasi Dengan Menggunakan <i>Node.JS</i> : Studi kasus Maluku Explore	<i>Website wisata</i>	Wisata Maluku	Sistem aplikasi <i>backend</i>

Dari Tabel 2.1 Penelitian Terkait, dapat dilihat bahwa telah dilakukan beberapa penelitian mengenai pengembangan *backend*. Penelitian ini merujuk pada studi-studi sebelumnya yang relevan dengan topik yang sedang diteliti, membantu untuk memberikan dukungan serta kontekstualisasi terhadap penelitian yang sedang dilakukan.

2.10 Posisi Penelitian

Data penelitian ini mencakup perbedaan dalam teknologi yang digunakan dan subjek penelitian, dibandingkan dengan penelitian yang ada sebelumnya :

Tabel 2. 2 Posisi Penelitian

No	Nama dan Tahun	Penelitian	Rancang Bangun Backend	Metode XP	Node.JS	Aplikasi wisata
1	Lisa A., Muhammad N.D.S., & Debby A., 2020	Sistem Informasi Akademik dan Administrasi dengan Metode <i>Extreme Programing</i> pada Lembaga Kursus dan Pelatihan	✓	✓		
2	Rizqy E.P.R, Umar A.A, & Burhanuddin D., 2023	Impelementasi <i>REST API</i> pada Pengembangan Aplikasi Backend untuk <i>Platfrom</i> Kursus Online (Growup)	✓		✓	

No	Nama dan Tahun	Penelitian	Rancang Bangun Backend	Metode XP	Node.JS	Aplikasi wisata
3	Hasanuddin, Hari A., & Budi H., 2022	Rancang Bangun REST API Aplikasi WESHARE Sebagai upaya Mempermudah Palayanan Donasi Kemanusiaan	✓		✓	
4	Nurhayati, Eli., & Aggusalim., 2023	Rancang Bangun Back-end API pada Aplikasi Mobile AyamHub Menggunakan Framework Node JS Express	✓		✓	
5	Mochamad Afif Solahudin	RANCANG BANGUN SISTEM INFORMASI STAYCATION BERBASIS WEB DENGAN	✓	✓	✓	

No	Nama dan Tahun	Penelitian	Rancang Bangun Backend	Metode XP	Node.JS	Aplikasi wisata
		IMPLEMENTASI TEKNOLOGI MERN STACK				
6	Muhamad Riyandi, 2024	Rancang Bangun Aplikasi Dengan Menggunakan <i>Node.JS</i> : Studi kasus Maluku Explore	✓	✓	✓	✓

Dalam Table 2.2 Posisi Penelitian, dapat terlihat terdapat beberapa penelitian yang memiliki persamaan, seperti dalam penggunaan metode dan teknologi yang telah digunakan. Berikut merupakan penjelasan dari rincian kepada masing-masing penelitian :

1. Lisa Ariyanti, Muhamad Najib, dan Debby Allita, melakukan penelitian tentang sistem informasi akademik dan administrasi pada LKPDuta Bahasa Korea dengan metode pengembang sistem yang dipilih menggunakan *Extreme Programming* dan pengembangan sistem administrasi menggunakan *PHP*.
2. Rizqy, Umar, dan Buharnuddin, mereka melakukan penelitian tentang penerapan *REST-API* dalam pengembangan aplikasi berbasis *web* menggunakan *Node.JS*, akan tetapi metode yang diterapkan bukan metode *Extreme Programming*. Hal tersebut membuktikan bahwa penelitian yang mereka lakukan berfokus kepada penggunaan *REST-API* dalam pengembangan aplikasi berbasis *web*.

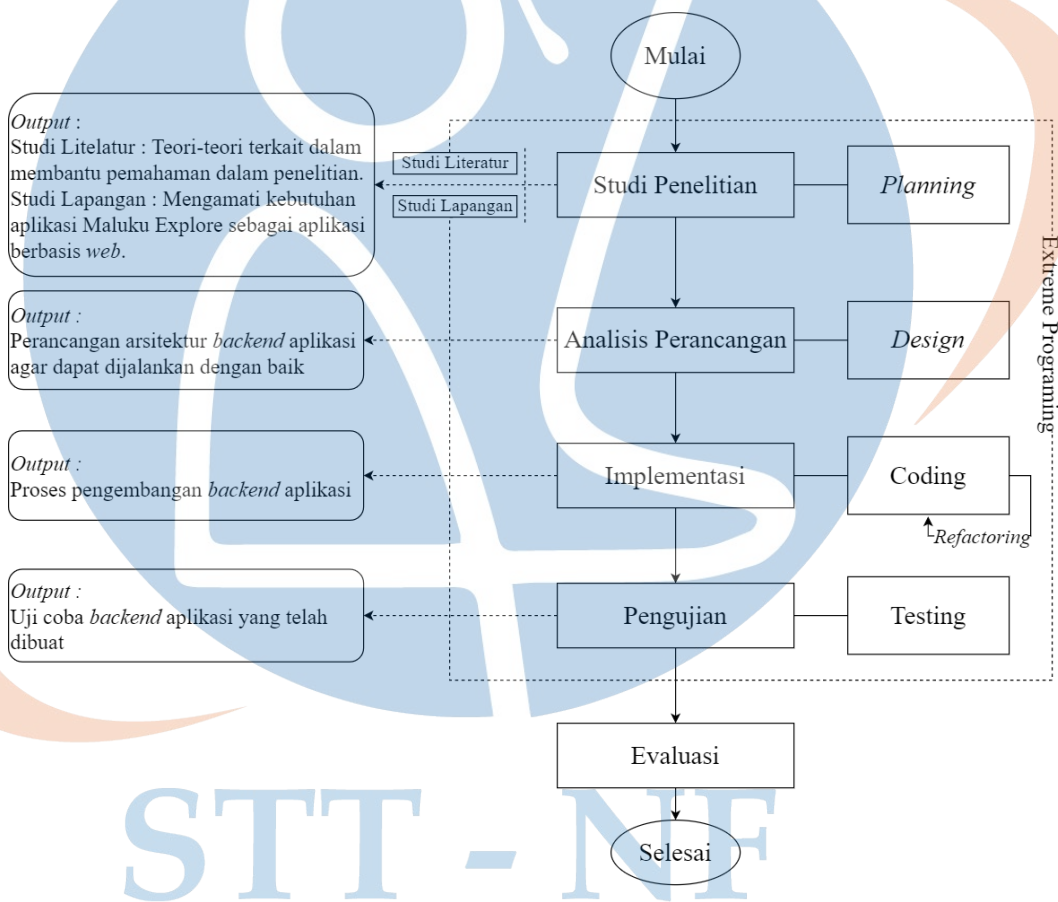
3. Hasanuddin, Hari Asgar, dan Budi Hartono, mereka melakukan penelitian tentang pengembangan aplikasi yang dapat mempermudah pelayanan donasi kemanusiaan, dengan menggunakan *Node.JS*, akan tetapi metode pengembangan yang dilakukan bukan menggunakan metode *Extreme Programming*, dan mereka hanya berfokus kepada pengembangan *REST-API*.
4. Eli Nurhayati, dan Agussalim, penelitian yang mereka lakukan adalah perancangan dan pembangunan layanan berbasis *web* menggunakan *Node.JS* untuk pengembangan dengan metode yang digunakan *Waterfall* bukan *Extreme Programming*, yang difokuskan pada pengembangan *API* pada layanan *web*.
5. Mochamad Afif Solahudin, melakukan penelitian tentang perancangan dan pembangunan layanan berbasis *website* untuk melakukan *booking* hotel dengan menggunakan *Node.JS* dan metode digunakan untuk pengembangan layanan tersebut menggunakan metode *Extreme Programing*, hal tersebut menunjukkan bahwa penelitian tersebut berfokus kepada pengembangan pelayanan aplikasi berbasis *web booking* hotel dengan pendekatan menggunakan *Extreme Programming* dan menggunakan *Node.JS* sebagai pengembangan arsitektur *backend*.

Dari uraian sebelumnya, dapat disimpulkan bahwa belum ada penelitian yang membahas pengembangan *backend* aplikasi untuk wisata Maluku dengan menggunakan *framework Node.JS* dan metode *Extreme Programming (XP)*. Oleh karena itu, penelitian ini akan mengisi kekosongan tersebut dengan berfokus kepada pengembangan *backend* aplikasi untuk *website* wisata Maluku dengan teknologi dan pendekatan tersebut.

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Berikut, penelitian menjalani beberapa tahapan yang sesuai dengan tujuan dari penelitian. Tahapan proses penelitian ini melibatkan alur yang diuraikan dengan rinci dalam setiap proses yang sudah disusun, sehingga penelitian tersebut dapat terlaksanakan dengan jelas dan terstruktur.



Gambar 3. 1 Tahapan Penelitian

Pada gambar 3.1 Tahapan Penelitian ini menerapkan metode *Extreme Programming (XP)* untuk pengembangan *backend* aplikasi wisata Maluku. *Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang pada umumnya akan melakukan pendekatan berorientasi objek. Metode ini cocok digunakan ketika pengembang dihadapkan dengan suatu kebutuhan yang tidak jelas atau perubahan yang cepat.

3.1.1 Studi Penelitian

Dalam studi penelitian, dilakukan studi literatur dalam memahami teknologi dan pendekatan terkini dalam pengembangan *backend* aplikasi. Fokusnya dengan mengidentifikasi solusi teknologi baru terkait pengembangan *backend*. Selain itu, dilakukannya studi lapangan dengan mengamati secara tidak langsung (*online*) dengan mengamati *website* yang memberikan informasi terkait wisata yang berada di Wilayah Maluku untuk mendapatkan wawasan yang praktis dalam konteks penelitian.

3.1.2 Analisis Perancangan

Dalam tahapan ini, melibatkan perancangan arsitektur sistem untuk pengembangan *backend* aplikasi wisata Maluku, dengan meliputi perancangan skema *database*, menentukan fitur-fitur yang diimplementasikan, format data yang dikirimkan, hingga kepada tahapan dalam penyusunan daftar prioritas berdasarkan dari hasil wawancara secara (*online*). Semua perancangan ini akan menjadi pondasi dalam membangun pengembangan alur kerja *backend* secara handal dan efisien.

3.1.3 Implementasi

Pada tahapan ini, pemrosesan implementasi akan dilakukan secara sistematis dan terstruktur didasarkan prioritas yang telah ditetapkan dalam daftar prioritas pengguna, melibatkan proses pembuatan kebutuhan pengguna menggunakan bahasa pemrograman *javascript* dengan *runtime Node.JS* dan kerangka kerja menggunakan *Express.JS*.

3.1.4 Pengujian

Pada tahapan pengujian, proses dari implementasi dilakukan secara sistematis dengan terstruktur didasarkan prioritas yang telah ditetapkan dalam pengembangan. Hal tersebut melibatkan dalam kebutuhan dari pengguna hingga pembuatan *backend* menggunakan *runtime Node.JS* dan kerangka kerja menggunakan *Express.JS*.

3.1.5 Kesimpulan

Tahapan dari kesimpulan merupakan tahapan akhir yang akan berfokus kepada penyimpulan hasil temuan. Penelitian akan memamparkan solusi yang telah diimplementasikan untuk mengatasi permasalahan komunikasi informasi di wisata Maluku. Dari kesimpulan yang akan diberikan memuat evaluasi terkait efektivitas solusi dalam penanganan permasalahan, dan pencapaian tujuan yang telah ditetapkan sebelumnya.

3.2 Rancangan Penelitian

Rancangan penelitian merupakan penjelasan tentang rencana yang dilakukan dalam penelitian. Hal tersebut akan menampilkan tentang pendekatan dalam mengatasi tantangan yang akan dihadapi, jenis penelitian yang akan dipilih, metode yang akan dipilih dalam pengumpulan data, serta penjelasan dari metode implementasi dan evaluasi mengenai sistem informasi/*tools* yang telah digunakan.

3.2.1 Jenis Penelitian

Penelitian yang dilakukan, termasuk kedalam kategori Penelitian Pengembangan (*R&D*). Metode penelitian *R&D* merupakan penelitian yang sangat cocok dipilih dalam mengatasi permasalahan dengan menemukan solusi inovatif dalam pengembangan produk atau teknologi yang baru. Penelitian tersebut akan melibatkan identifikasi dari akar permasalahan, analisis dari kebutuhan, dan solusi perancangan yang efektif.

Dalam studi kasus yang diangkat dalam penelitian ini adalah rancang bangun *backend* aplikasi Maluku Explore. Penelitian tersebut melibatkan analisis dari kebutuhan pengguna, perancangan *backend* aplikasi wisata, implementasi, pengujian, dan evaluasi. Adapun luaran dari penelitian ini sebagai berikut :

1. Pengembangan *backend* aplikasi *web* yang mampu memenuhi kebutuhan aplikasi *web*.
2. Penerapan teknologi *Node.JS* sebagai kerangka kerja pengembangan dalam memastikan kinerja aplikasi berjalan dengan baik.

3. Integrasi konsep *Node.JS* dalam meningkatkan skalabel, reponsif, dan efisien dari aplikasi.
4. Penilaian evaluasi yang dianalisis sejauh mana implementasi *Express.JS* dan integrasi *Node.JS* berhasil mencapai tujuan dalam membangun *backend* aplikasi. Evaluasi ini akan memberikan sebuah gambaran mengenai efektivitas dari solusi yang diberikan dalam merancang serta membangun *backend* aplikasi.

3.2.2 Metode Analisis

Penelitian ini menggunakan metode analisis kualitatif untuk mengolah data yang diperoleh dari wawancara *online* dan observasi. Pemilihan metode kualitatif dilakukan dengan pertimbangan matang, karena metode ini memungkinkan penggalan informasi yang lebih mendalam dan kaya tentang pengalaman pengguna.

Metode analisis kualitatif yang digunakan dalam penelitian ini, dengan berfokus pada wawancara *online* dan observasi, memungkinkan pengumpulan data yang kaya dan mendalam tentang pengalaman pengguna. Penggunaan platform online yang tepat dan analisis data yang cermat akan menghasilkan temuan penelitian yang berkualitas tinggi dan bermakna.

3.2.3 Metode Pengumpulan Data

Dalam penelitian ini, data dan informasi yang telah dikumpulkan melalui beberapa metode, yaitu :

1. Wawancara *online* yang dilakukan oleh penulis, memungkinkan peneliti untuk mendapatkan berbagai perspektif dari responden. Bertujuan untuk mendapatkan informasi yang kontekstual tentang pengalaman yang didapatkan, kendala yang dialami, dan fitur-fitur yang diharapkan pengguna dalam sistem.
2. Studi kepustakaan menjadi langkah awal yang krusial dalam pengembangan *backend* untuk aplikasi wisata Maluku Explore. Hal tersebut dilakukan dengan mengumpulkan informasi serta pengetahuan yang terkait dari berbagai sumber terpercaya, seperti jurnal, buku, artikel, dan makalah. Dalam proses yang dilakukan secara cermat dan sistematis ini, memastikan relevansi dan

kredibilitas informasi yang akan diperoleh, serta berguna dalam mendukung pengembangan *backend* aplikasi yang efektif dan optimal.

3. Observasi dilakukan terhadap wisata wilayah Maluku untuk mengamati secara tidak langsung (*online*) perilaku, interaksi, dan penyampaian proses informasi yang telah dilakukan disana. Dengan bertujuan untuk mendapatkan informasi yang akurat serta mendalam tentang bagaimana *website* Maluku Explore dapat beroperasi dengan baik.

3.2.4 Metode Implementasi dan Evaluasi

Pada tahapan implementasi merupakan langkah dalam membangun *backend* aplikasi Maluku Explore berdasarkan hasil dari tahapan perancangan. Sistem yang akan dibangun berbasis *website* yang dimana akan diuraikan melalui tampilan berupa *screen shoot* dari hasil tersebut. Tahapan evaluasi akan dilakukan dengan merumuskan kesimpulan tentang efektivitas rancangan dan mengidentifikasi keperluan yang harus diperbaiki.

3.3 Metode Pengujian

Berdasarkan dari uraian sebelumnya, dalam penelitian ini menerapkan desain pengujian guna memperoleh hasil dari evaluasi yang sesuai dengan penelitian dan menunjukkan bahwa pengembangan *backend* aplikasi untuk sistem Maluku Explore sesuai dari ekspektasi penulis. Desain pengujian yang digunakan dalam perancangan *backend* aplikasi ini menggunakan metode *black box testing*. Hasil pengujian yang dilakukan diharapkan dapat menjadi bahan pertimbangan bagi pengguna dalam menggunakan sistem informasi wisata Maluku Explore.

Pada penelitian ini, penulis menggunakan metode *Black Box* dalam menguji fungsionalitas dari fitur-fitur yang dirancang. Pengujian akan dilakukan oleh 4 penguji yang berbeda dengan menggunakan aplikasi Postman, dengan memungkinkan pengujian fungsionalitas dari fitur *backend* yang dikembangkan melalui penggunaan permintaan *HTTP*.

Tabel 3. 1 Black Box Testing

NO.	Fitur	Tahap Pengujian	HTTP Method/Endpoint	Hasil yang diharapkan	Kesimpulan	Penguji
1	User	Register	Post : localhost:3000/auth/register	Akan menghasilkan output: response (201) & message 'User Berhasil Dibuat!'	Sesuai atau tidak sesuai	Penguji 1
		Login	Post: localhost:3000/auth/login	Akan menghasilkan output: response (201)	Sesuai atau tidak sesuai	Penguji 1
		Logout	Delete : localhost:3000/auth/logout	Akan menghasilkan output : response(200), 'Logout Berhasil !'	Sesuai atau tidak sesuai	Penguji 1
2	Halaman Blog website	Create	Post:localhost:3000/blog/create/	Akan menghasilkan output : response (201) &	Sesuai atau tidak sesuai	Penguji 2

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Penguji
				message 'Data Berhasil Ditambahkan !'		
		Read	Get:localhost:3000/blog/get/	Akan menghasilkan output : response (200)	Sesuai atau tidak sesuai	Penguji 2
			Get:localhost:3000/blog/get/byId			Penguji 2
		Change	Put:localhost:3000/blog/update/byId	Akan menghasilkan output : response (200) & message 'Blog Berhasil Diubah !'	Sesuai atau tidak sesuai	Penguji 2
		Delete	Delete:localhost:3000/blog/delete/byId	Akan menghasilkan output : response (200) & message 'Data	Sesuai atau tidak sesuai	Penguji 2

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Penguji
				Berhasil Dihapus !'		
3	Halaman tour wisata	Create	<i>Post:localhost:3000/tour/create</i>	Akan menghasilkan output : response (201) & message 'Data Berhasil Ditambah !'	Sesuai atau tidak sesuai	Penguji 3
		Read	<i>Get:localhost:3000/tour/get</i>	Akan menghasilkan output : response (200)	Sesuai atau tidak sesuai	Penguji 3
			<i>Get:localhost:3000/tour/get/byId</i>	Akan menghasilkan output : response (200)	Sesuai atau tidak sesuai	Penguji 3
			Change	<i>Put:localhost:3000/tour/update/byId</i>	Akan menghasilkan output : response (200) & message	Sesuai atau tidak sesuai

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Penguji
				'Data Berhasil Diubah !'		
		Delete	Delete:localhost:3000/delete/byId	Akan menghasilkan output : response (200) & message 'Data Berhasil Dihapus !'	Sesuai atau tidak sesuai	Penguji 3
4	Pemesanan Tiket	Create	Post:localhost:3000/order/create	Akan menghasilkan output : response (201) & message 'Data Berhasil Ditambahkan !'	Sesuai atau tidak sesuai	Penguji 4
		Read	Get:localhost:3000/order/get/	Akan menghasilkan output : response (200)	Sesuai atau tidak sesuai	Penguji 4

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Penguji
			<i>Get:localhost:3000/order/get/byId</i>	Akan menghasilkan <i>output : response (200)</i>	Sesuai atau tidak sesuai	
		<i>Change</i>	<i>Put:localhost:3000/order/update/byId</i>	Akan menghasilkan <i>output : response (200) & message 'Data Berhasil Diubah !'</i>	Sesuai atau tidak sesuai	Penguji 4
		<i>Delete</i>	<i>Delete:localhost:3000/order/byId</i>	Akan menghasilkan <i>output : response (200) & message 'Data Berhasil Dihapus !'</i>	Sesuai atau tidak sesuai	Penguji 4

3.4 Lingkungan Pengembangan

Lingkungan pengembangan merupakan tempat di mana penelitian yang akan dilakukan, alat yang digunakan, perangkat lunak, dan sumber daya lainnya yang digunakan oleh pengembang perangkat lunak atau aplikasi dalam membuat,

menguji, dan menyusun kode. Hal tersebut mencakup dari semua elemen yang diperlukan dalam mendukung proses pengembangan perangkat lunak dari awal hingga akhir.

3.4.1 Lokasi Penelitian

Penelitian ini memilih wilayah wisata Maluku sebagai tempat penelitian dikarenakan adanya kekurangan dalam penyampaian informasi wisata di wilayah tersebut. Diharapkan dalam penelitian ini akan berkontribusi dalam meningkatkan jumlah wisatawan dan proses penyampaian wisata di wilayah tersebut.

3.4.2 Alat dan Bahan

Penelitian ini menggunakan berbagai alat dan bahan dalam mengumpulkan data dengan cermat. Alat-alat tersebut membantu dalam proses penelitian, seperti:

1. Kebutuhan Alat

Penelitian tugas akhir ini menggunakan laptop sebagai perangkat keras utama.

Berikut merupakan spesifikasi laptop yang digunakan :

- a. Laptop MSI GF63Thin
- b. Windows 10 Home 64-bit
- c. RAM 16GB
- d. Intel(R) Core(TM) i5-10200H CPU @ 2.40GHz – 2.40 GHz

2. Kebutuhan Bahan

Penelitian ini menggunakan beberapa perangkat lunak dalam spesifikasi tertentu guna membantu proses penelitian dan pencapaian tujuan yang diinginkan.

Berikut beberapa perangkat lunak yang digunakan :

- a. *Tex Editor* : Visual Studio Code
- b. *Node.JS* : V20.12.2
- c. *API Testing Tools* : Postman
- d. Desain diagram : app.diagrams.net

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Analisis dan Perancangan Sistem

Dalam tahapan analisis dan perancangan sistem pada proyek ini akan berfokus kepada pemahaman dalam permasalahan informasi wisata yang di wilayah wisata Maluku. Hal tersebut dilakukan untuk mengetahui kebutuhan sistem secara komprehensif dalam konteks penelitian. Proses ini mencakup beberapa tahapan, mulai dari analisis *user stories*, identifikasi aktor, *product backlog*, dan pembuatan diagram seperti : *use case diagram*, *class diagram*, dan *activity diagram*.

4.1.1 User Stories

Berdasarkan dari hasil analisis dari berbagai sumber seperti laporan industri, artikel media, dan wawancara yang dipublikasi, penulis berhasil mengidentifikasi beberapa cerita pengguna (*user stories*) dengan menggambarkan kebutuhan serta harapan pengguna terhadap aplikasi Maluku Explore:

Tabel 4. 1 User Stories

No	Sebagai	Saya Ingin	Sehingga
1	Admin	Input data staff dan pengunjung	Saya dapat dengan mudah mengelola data staff dan pengunjung.
2	Admin dan Staff	Input data blog wisata	Saya dapat dengan mudah mengelola data blog wisata dengan mudah.
3	Admin, staff, dan pengunjung	Melihat daftar semua blog wisata	Saya dapat mengetahui informasi tentang blog wisata dengan mudah.
4	Admin dan Staff	Merubah data blog wisata	Saya dapat merubah informasi tentang blog wisata dengan mudah.
5	Admin dan Staff	Menghapus data blog wisata	Saya dapat menghapus tentang blog wisata dengan mudah.

No	Sebagai	Saya Ingin	Sehingga
6	Admin dan Staff	Input data <i>tour</i> wisata	Saya dapat dengan mudah mengelola data <i>tour</i> wisata.
7	Admin dan Staff	Melihat data <i>tour</i> wisata	Saya dapat dengan mudah melihat data <i>tour</i> wisata.
8	Admin dan Staff	Merubah data <i>tour</i> wisata	Saya dapat dengan mudah mengubah data <i>tour</i> wisata.
9	Admin dan Staff	Menghapus data <i>tour</i> wisata	Saya dapat dengan mudah menghapus data <i>tours</i> wisata.
10	Admin dan Staff	Input data order	Saya dapat dengan mudah mengelola data order wisata.
11	Admin dan Staff	Melihat data order wisata	Saya dapat dengan mudah melihat data order wisata.
12	Admin dan Staff	Merubah data order wisata	Saya dapat dengan mudah mengubah data order wisata.
13	Admin dan Staff	Menghapus data order wisata	Saya dapat dengan mudah menghapus data order wisata.
14	Pengunjung	Melihat daftar blog	Saya dapat mengetahui informasi wisata.
15	Pengunjung	Melihat daftar <i>tour</i>	Saya dapat mengetahui informasi tempat wisata.
16	Pengunjung	Melakukan order	Saya dapat memesan tiket tempat wisata.

4.1.2 Identifikasi Aktor

Berdasarkan dari hasil uraian *user stories* sebelumnya, disimpulkan bahwa beberapa aktor penting yang berperan dalam pengembangan sistem aplikasi informasi wisata Maluku. Aktor tersebut ialah :

Tabel 4. 2 Identifikasi Aktor

No	Aktor	Deskripsi
1	Admin	Pengelola utama aplikasi dengan akses penuh untuk mengelola konten, pengguna, dan sistem.
2	Staff	Staff merupakan personel yang membantu admin dalam mengelola informasi wisata Maluku.
3	Pengunjung	Calon wisatawan yang menggunakan aplikasi untuk mendapatkan informasi dan layanan wisata di Maluku Explore.

4.1.3 Product Backlog

Berdasarkan *user stories*, peneliti merangkum fitur-fitur yang telah diperlukan dalam aplikasi *backend* Maluku Explore. Rangkuman ini akan menjelaskan berbagai *task* yang harus dilakukan untuk mengembangkan aplikasi, fitur prioritas, dan *story point* yang digunakan dalam mengukur kompleksibilitas atau waktu dalam pengerjaan setiap *task*.

Dalam pengembangan aplikasi, *task* diberikan nilai prioritas berdasarkan kebutuhannya. Kompleksitas *task* diukur dengan skala *story points*, nilai 1 setara dengan 2-7 jam pengerjaan, dan 2 nilai setara dengan 8-12 jam. Aplikasi juga disertakan deskripsi dalam memperjelas tujuan dan fungsinya. Informasi ini diharapkan dapat membantu dalam pengembangan sistem informasi wisata yang terstruktur dan efisien.

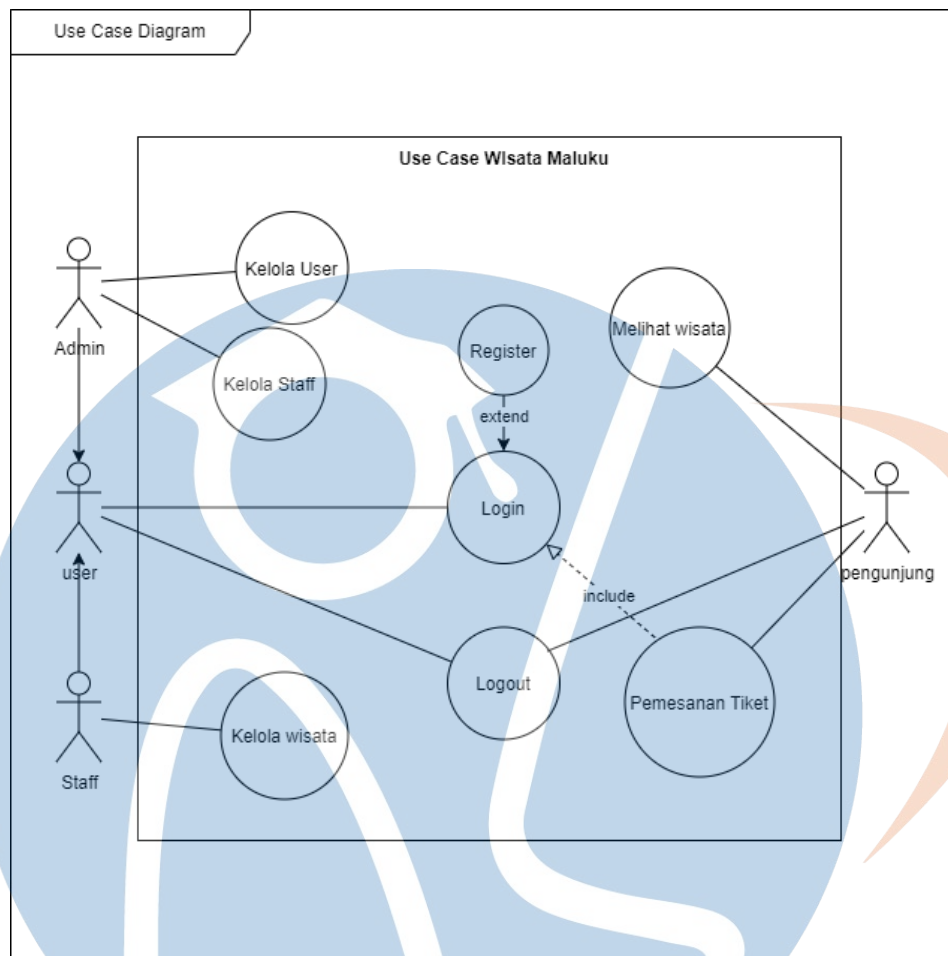
Tabel 4. 3 Product Backlog

No	Task	Prioritas	Story Points	Deskripsi
1	Pembuatan <i>user requirement</i>	Tinggi	1	Merancang serta mencatat kebutuhan pengguna dengan detail dalam mengenali fitur inti dari aplikasi.
2	Pembuatan skema <i>database</i>	Tinggi	1	Merancang skema <i>database</i> dalam penyimpanan data sistem wisata.

No	Task	Prioritas	Story Points	Deskripsi
3	<i>Migration database table</i>	Tinggi	1	Mengkonfirmasi struktur <i>database</i> sudah sesuai dengan desain dalam melakukan migrasi skema <i>database</i> dalam sistem basis data.
4	Pembuatan <i>API login, logout dan register</i>	Tinggi	2	Pembuatan <i>API</i> dengan memungkinkan pengguna untuk <i>login, logout, dan register</i> dengan menggunakan <i>JWT</i> .
5	Pembuatan <i>API blog</i>	Tinggi	2	Pembuatan <i>API</i> dalam mengelola data <i>blog</i> memungkinkan pengguna untuk dapat melihat informasi wisata Maluku
6	Pembuatan <i>API tour</i>	Tinggi	2	Pembuatan <i>API</i> dalam mengelola data <i>tour</i> memungkinkan pengguna dapat melihat tempat wisata Maluku
7	Pembuatan <i>API order</i>	Tinggi	2	Pembuatan <i>API</i> dalam mengelola data <i>order</i> memungkinkan pengguna dapat melakukan pemesanan tiket tempat wisata Maluku

4.1.4 Use Case Diagram

Use case diagram untuk pengembangan *backend* Maluku Explore yang menggambarkan interaksi antara aktor (pengguna) dengan sistem melalui dari berbagai kasus penggunaan. Diagram ini akan memberikan gambar visual dari interaksi pengguna dengan sistem. Hal tersebut sangat dibutuhkan untuk mengetahui bagaimana sistem akan berkerja, dengan bagaimana pengguna dapat menggunakan untuk mendapatkan informasi yang dibutuhkan.



Gambar 4. 1 Use Case Diagram

Pada gambar 4.1 *Use Case Diagram* menggambarkan sebuah interaksi antar 3 aktor (Admin, Staff, dan Pengunjung) dengan aplikasi *backend* Maluku Explore.

Para aktor memiliki peran dengan akses yang berbeda dalam sistem tersebut seperti:

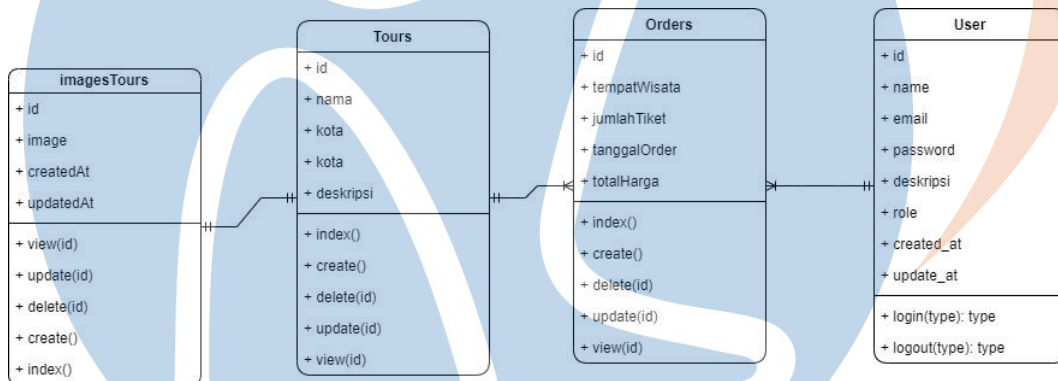
1. Admin memiliki hak penuh pengelolaan data *user*. Admin dapat menambah, melihat, mengubah data pengguna. Hal tersebut untuk memastikan keakuratan data didalam sistem.
2. Staff memiliki peran dalam mengelola data wisata Maluku. Staff memiliki hak atas menambahkan, melihat, mengubah, dan menghapus data blog, dan tour wisata.

3. Pengguna memiliki akses yang terbatas pada sistem. Pengguna hanya dapat melihat informasi wisata yang telah dimasukan oleh staff, seperti tours, dan blog wisata.

Use Case Diagram menggambarkan bagaimana *backend* aplikasi Maluku Explor memberikan interaksi yang terstruktur antara Admin, Staff, dan Pengguna untuk mengelola dan mengakses informasi wisata dengan mudah dan efisien.

4.1.5 Class Diagram

Class Diagram informasi Maluku Explore menggambarkan hubungan antar kelas, objek, atribut, dan operasi dalam sistem. Diagram ini akan membantu proses desain sistem dan spesifikasi sistem aplikasi menjadi terstruktur.



Gambar 4. 2 Class Diagram

Class Diagram pada Gambar 4.2 memiliki 4 kelas dengan atribut dan operasi yang menggambarkan kualitas dan interaksi kelas dalam sebuah sistem. Setiap kelas tersebut memiliki atribut dalam mendefinisikan karakteristiknya, dan operasi untuk memungkinkan kelas berinteraksi dengan data dan kelas lain. Diagram ini akan sangat membantu dalam memahami struktur dan fungsionalitas sistem informasi wisata Maluku Explore.

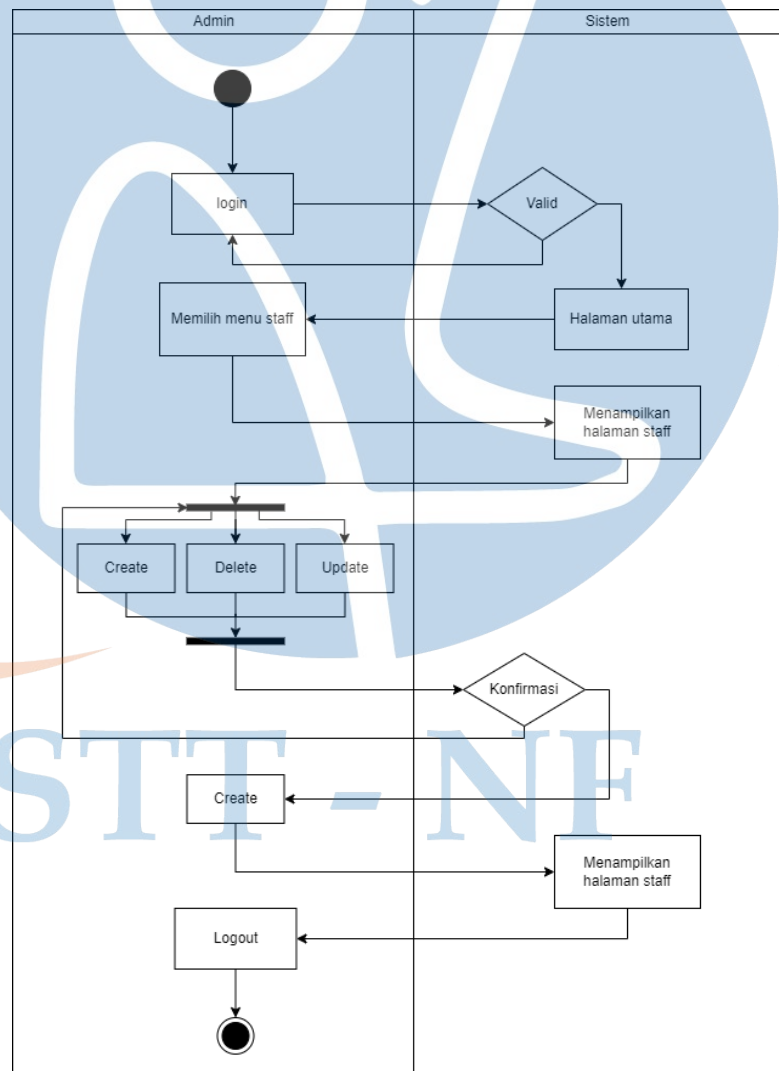
4.1.6 Activity Diagram

Activity diagram adalah gambaran diagram *UML* yang digunakan dalam mevisualkan alur kerja atau proses dalam suatu sistem. Diagram tersebut akan menunjukkan langkah-langkah yang terlibat dalam proses, dan bagaimana langkah tersebut saling terkait. Hal tersebut dapat membantu dalam analisis sistem,

pengembangan, mengidentifikasi inefisiensi, dan meningkatkan sistem informasi wisata.

1. Activity Diagram Mengelola Data Staff

Activity Diagram ini mengilustrasikan alur kerja Admin dalam melakukan pengolahan data staff sistem wisata Maluku Explore. Diagram ini menampilkan langkah demi langkah yang dilalui oleh admin, dimulai dengan *login* sampai *logout*, serta bagaimana kerja sebuah sistem dalam melakukan verifikasi data, menyimpan perubahan data, dan menampilkan informasi yang relevan.



Gambar 4. 3 Activity Diagram Mengelola Data Staff

Pada gambar 4.3 Activity Diagram Mengelola Data Staff, admin pertama kali melakukan proses *login* ke dalam sistem wisata Maluku Explore, kemudian sistem

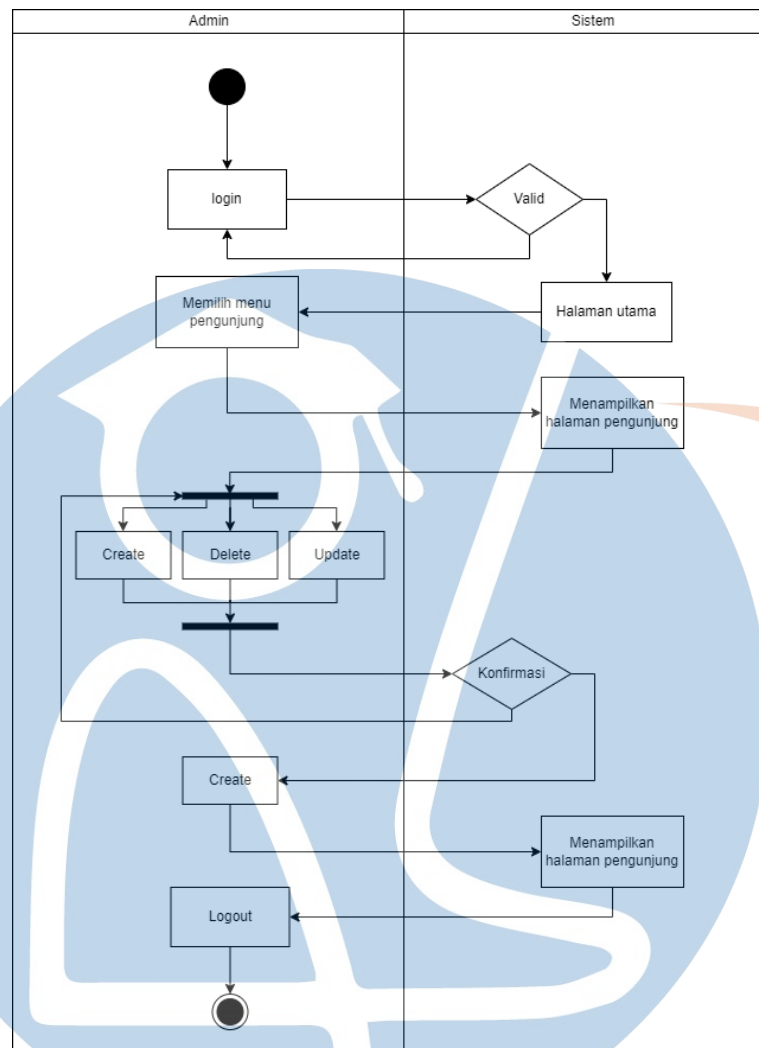
akan melakukan verifikasi data *login* yang terinput, jika data tersebut valid, maka sistem akan menampilkan halaman utama. Pada halaman utama, admin memilih menu staff untuk mengakses halaman kelola data staff. Kemudian admin dapat memilih salah satu dari beberapa aktivitas seperti menambahkan, melihat, mengubah, dan menghapus data staff. Setelah admin menyelesaikan pilihan aktivitas tersebut, sistem akan melakukan konfirmasi untuk menyimpan perubahan kedalam *database*. Kemudian admin akan diarahkan kembali kepada halaman staff, apabila sudah selesai, admin dapat melakukan *logout* dari sistem.

2. *Activity Diagram* Mengelola Data pengunjung

Activity Diagram ini akan mengilustrasikan alur kerja Admin dalam melakukan pengolahan data pengunjung sistem wisata Maluku Explore. Diagram ini menampilkan langkah demi langkah yang dilalui oleh admin, dimulai dengan *login* sampai *logout*, serta bagaimana kerja sebuah sistem dalam melakukan verifikasi data, menyimpan perubahan data, dan menampilkan informasi yang relevan.



STT - NF



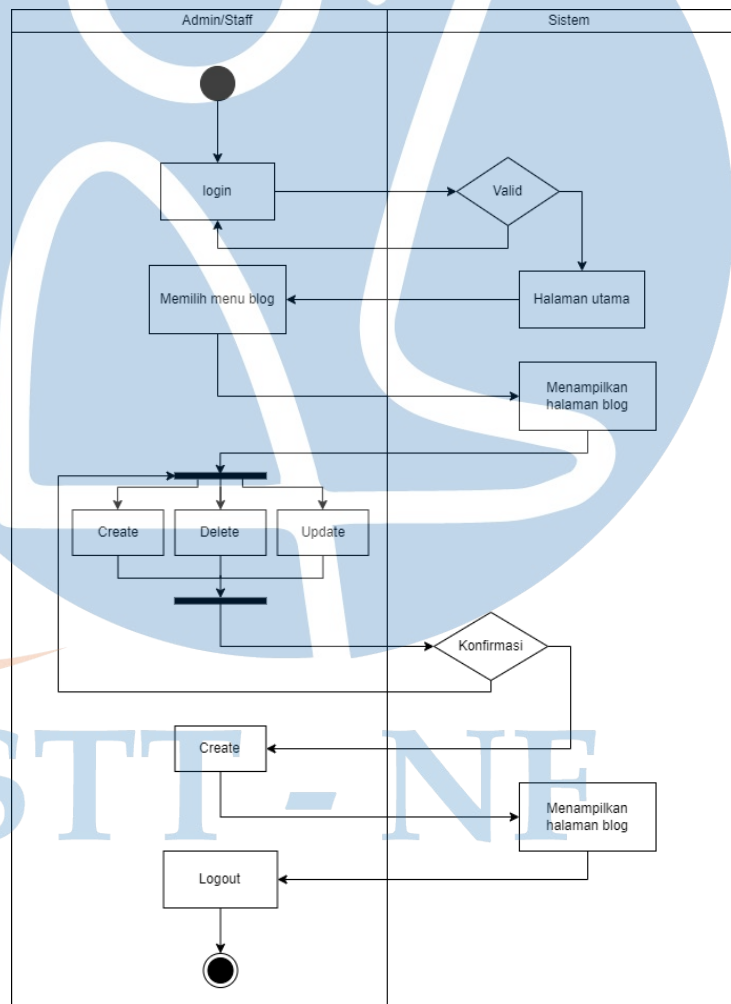
Gambar 4. 4 Activity Diagram Mengelola Data Pengunjung

Pada gambar 4.4 Activity Diagram Mengelola Data pengunjung, admin pertama kali melakukan proses *login* ke dalam sistem wisata Maluku Explore, kemudian sistem akan melakukan verifikasi data *login* yang terinput, jika data tersebut valid, maka sistem akan menampilkan halaman utama. Pada halaman utama, admin memilih menu pengunjung untuk mengakses halaman kelola data pengunjung. Kemudian admin dapat memilih salah satu dari berberapa aktivitas seperti menambahkan, melihat, mengubah, dan menghapus data pengunjung. Setelah admin menyelesaikan pilihan aktivitas tersebut, sistem akan melakukan konfirmasi untuk menyimpan perubahan kedalam *database*. Kemudian admin akan diarahkan

kembali kepada halaman pengunjung, apabila sudah selesai, admin dapat melakukan *logout* dari sistem.

3. Activity Diagram Mengelola Data Blog

Activity Diagram ini akan mengilustrasikan alur kerja Admin dan Staff dalam melakukan pengolahan data blog sistem wisata Maluku Explore. Diagram ini menampilkan langkah demi langkah yang dilalui oleh admin dan staff, dimulai dengan *login* sampai *logout*, serta bagaimana kerja sebuah sistem dalam melakukan verifikasi data, menyimpan perubahan data, dan menampilkan informasi yang relevan.



Gambar 4.5 Activity Diagram Mengelola Data Blog

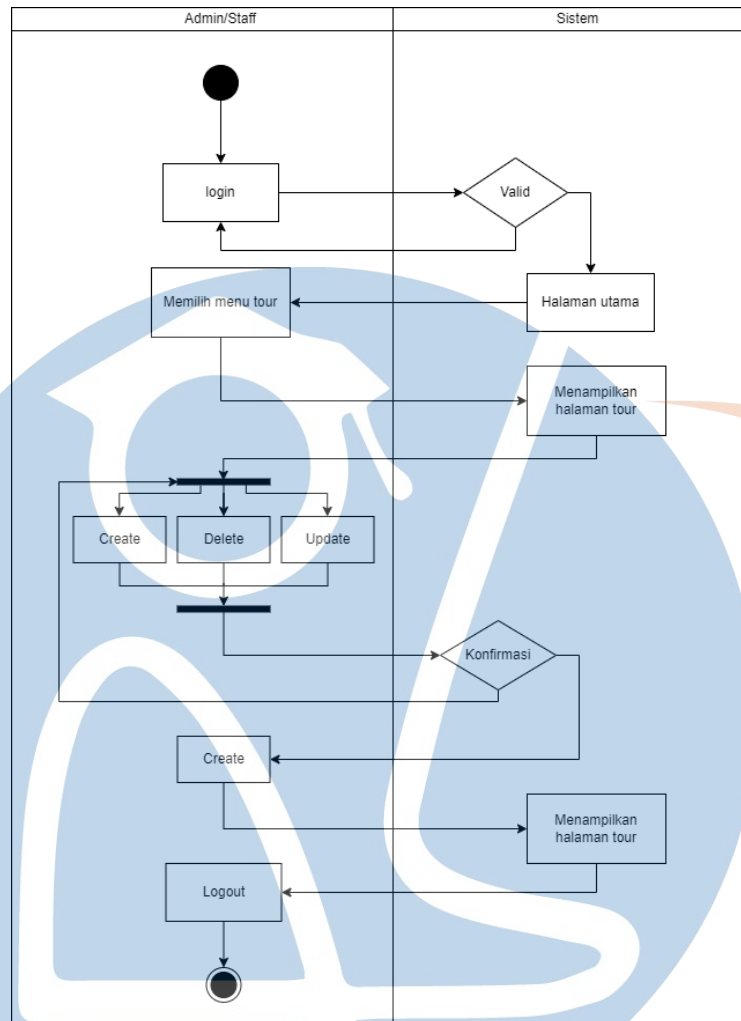
Pada gambar 4.5 *Activity Diagram Mengelola Data Blog*, admin dan staff pertama kali melakukan proses *login* ke dalam sistem wisata Maluku Explore,

kemudian sistem akan melakukan verifikasi data *login* yang terinput, jika data tersebut valid, maka sistem akan menampilkan halaman utama. Pada halaman utama, admin memilih menu blog untuk mengakses halaman kelola data blog. Kemudian admin dapat memilih salah satu dari berberapa aktivitas seperti menambahkan, melihat, mengubah, dan menghapus data blog. Setelah admin menyelesaikan pilihan aktivitas tersebut, sistem akan melakukan konfirmasi untuk menyimpan perubahan kedalam *database*. Kemudian admin akan diarahkan kembali kepada halaman blog, apabila sudah selesai, admin dapat melakukan *logout* dari sistem.

4. *Activity Diagram* Mengelola Data *Tour* (REVISI GAMBAR)

Activity Diagram ini akan mengilustrasikan alur kerja Admin dan Staff dalam melakukan pengolahan data *tour* sistem wisata Maluku Explore. Diagram ini menampilkan langkah demi langkah yang dilalui oleh admin dan staff, dimulai dengan *login* sampai *logout*, serta bagaimana kerja sebuah sistem dalam melakukan verifikasi data, menyimpan perubahan data, dan menampilkan informasi yang relevan.

STT - NF



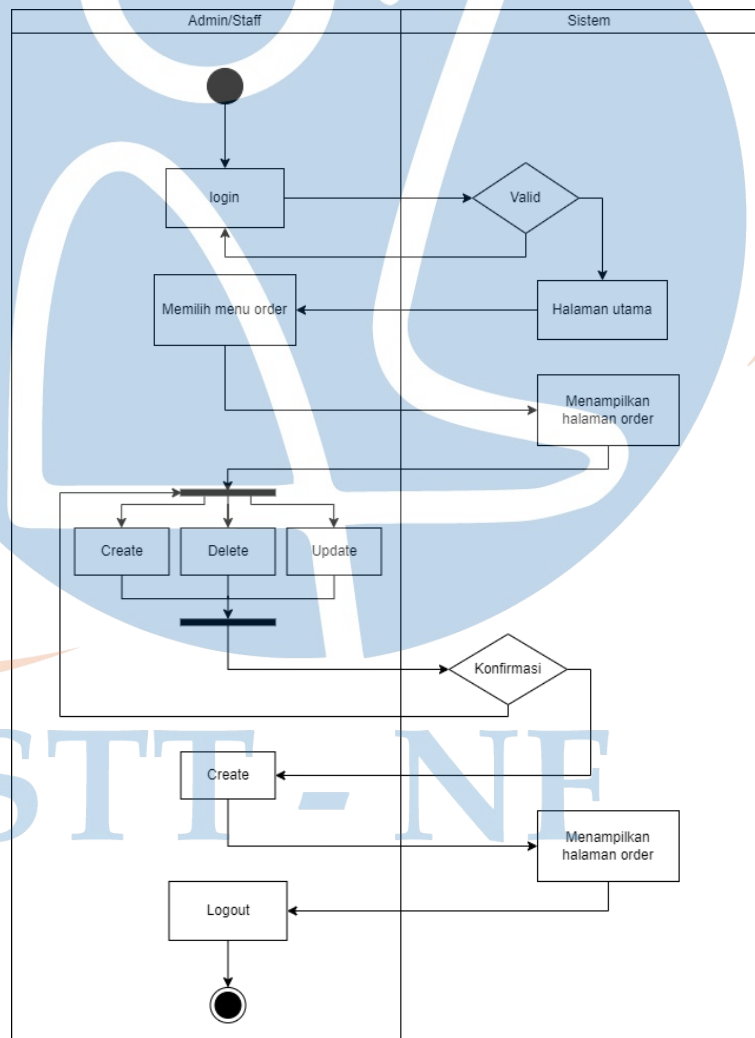
Gambar 4. 6 Activity Diagram Mengelola Data Tour

Pada gambar 4.6 *Activity Diagram Mengelola Data Tour*, admin dan staff pertama kali melakukan proses *login* ke dalam sistem wisata Maluku Explore, kemudian sistem akan melakukan verifikasi data *login* yang terinput, jika data tersebut valid, maka sistem akan menampilkan halaman utama. Pada halaman utama, admin memilih menu *tour* untuk mengakses halaman kelola data *tour*. Kemudian admin dapat memilih salah satu dari berberapa aktivitas seperti menambahkan, melihat, mengubah, dan menghapus data *tour*. Setelah admin menyelesaikan pilihan aktivitas tersebut, sistem akan melakukan konfirmasi untuk menyimpan perubahan kedalam *database*. Kemudian admin akan diarahkan

kembali kepada halaman *tour*, apabila sudah selesai, admin dapat melakukan *logout* dari sistem.

5. Activity Diagram Mengelola Data Order

Activity Diagram ini akan mengilustrasikan alur kerja Admin dan Staff dalam melakukan pengolahan data *tour* sistem wisata Maluku Explore. Diagram ini menampilkan langkah demi langkah yang dilalui oleh admin dan staff, dimulai dengan *login* sampai *logout*, serta bagaimana kerja sebuah sistem dalam melakukan verifikasi data, menyimpan perubahan data, dan menampilkan informasi yang relevan.



Gambar 4. 7 Activity Diagram Mengelola Data Order

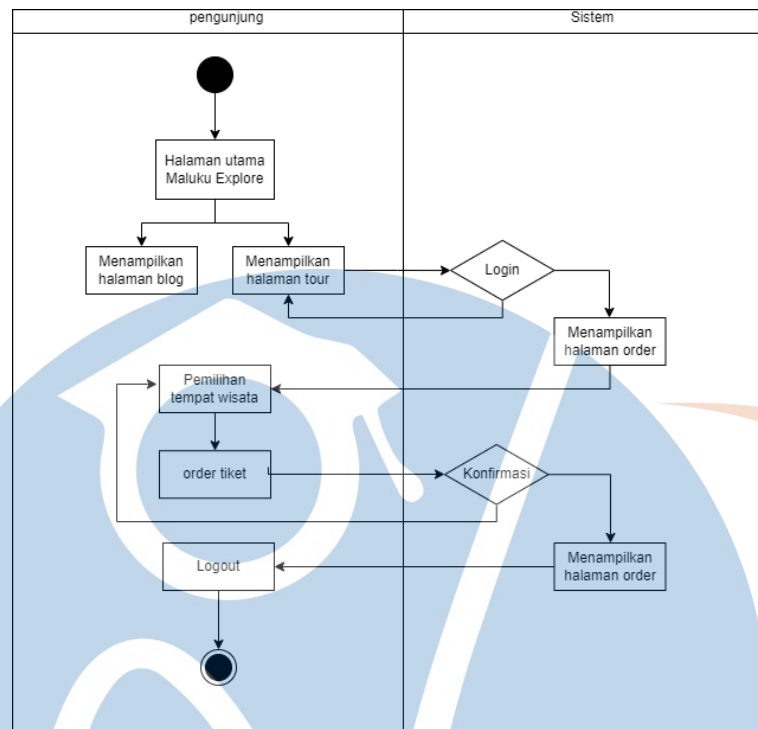
Pada gambar 4.7 *Activity Diagram* Mengelola Data Order, admin dan staff pertama kali melakukan proses *login* ke dalam sistem wisata Maluku Explore,

kemudian sistem akan melakukan verifikasi data *login* yang terinput, jika data tersebut valid, maka sistem akan menampilkan halaman utama. Pada halaman utama, admin memilih menu order untuk mengakses halaman kelola data order. Kemudian admin dapat memilih salah satu dari beberapa aktivitas seperti menambahkan, melihat, mengubah, dan menghapus data order. Setelah admin menyelesaikan pilihan aktivitas tersebut, sistem akan melakukan konfirmasi untuk menyimpan perubahan kedalam *database*. Kemudian admin akan diarahkan kembali kepada halaman order, apabila sudah selesai, admin dapat melakukan *logout* dari sistem.

6. *Activity Diagram* Pengguna

Activity Diagram ini mengilustrasikan alur kerja Pengguna dalam melakukan pengolahan data staff sistem wisata Maluku Explore. Diagram ini menampilkan langkah demi langkah yang dilalui oleh pengguna, dimulai dengan *login* sampai *logout*, serta bagaimana kerja sebuah sistem dalam melakukan verifikasi data, menyimpan perubahan data, dan menampilkan informasi yang relevan.

STT - NF



Gambar 4. 8 Activity Diagram Pengguna

Pada gambar 4.8 *Activity Diagram* Pengguna, Pengguna pertama kali akan melihat tampilan dari halaman utama wisata Maluku Explore yaitu halaman blog dan halaman tour. Pada halaman tour jika pengguna ingin melakukan order tiket tempat wisata pada halaman order, pengguna harus melakukan *login* terlebih dahulu kedalam sistem informasi wisata Maluku Explore, jika data tersebut valid maka sistem akan menampilkan halaman order. Pada halaman order, pengguna memilih menu order tiket untuk mengakses halaman order tiket. Kemudian pengguna dapat melakukan pemesanan tiket wisata yang ingin telah dipilih dan data tersebut akan terinput kedalam order tiket. Setelah melakukan pemesanan tiket, sistem akan melakukan konfirmasi untuk menyimpan perubahan kedalam *database*. Kemudian pengguna akan diarahkan kembali kepada halaman order, apabila sudah selesai, pengguna dapat melakukan *logout* dari sistem.

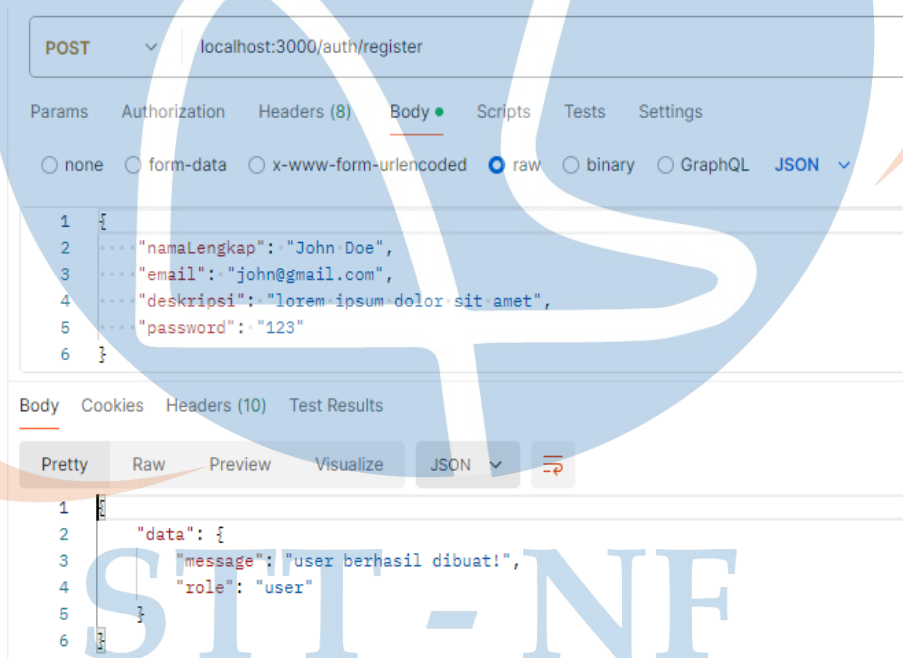
4.2 Implementasi Sistem Maluku Explore

Dalam tahap implementasi, *API* sistem informasi Maluku Explore merupakan upaya dalam memanfaatkan teknologi untuk pengelolaan informasi wisata sehingga dapat diakses dengan mudah, kapan saja, dan dimana saja. Hal tersebut merupakan langkah realisasi dari perancangan yang telah dibuat, menjadi sebuah aplikasi *backend* yang dikembangkan menggunakan *Node.js*.

4.2.1 Antarmuka Kelola Data Pengguna

Pada tahapan implementasi antarmuka kelola data pengunjung ini, peneliti akan menyajikan uji implementasi sistem layanan *REST API* dalam mengelola data pengunjung. Penjelasan akan difokuskan kepada *request* yang dimasukan dan *response* yang akan diterima saat pengguna berinteraksi dengan sistem.

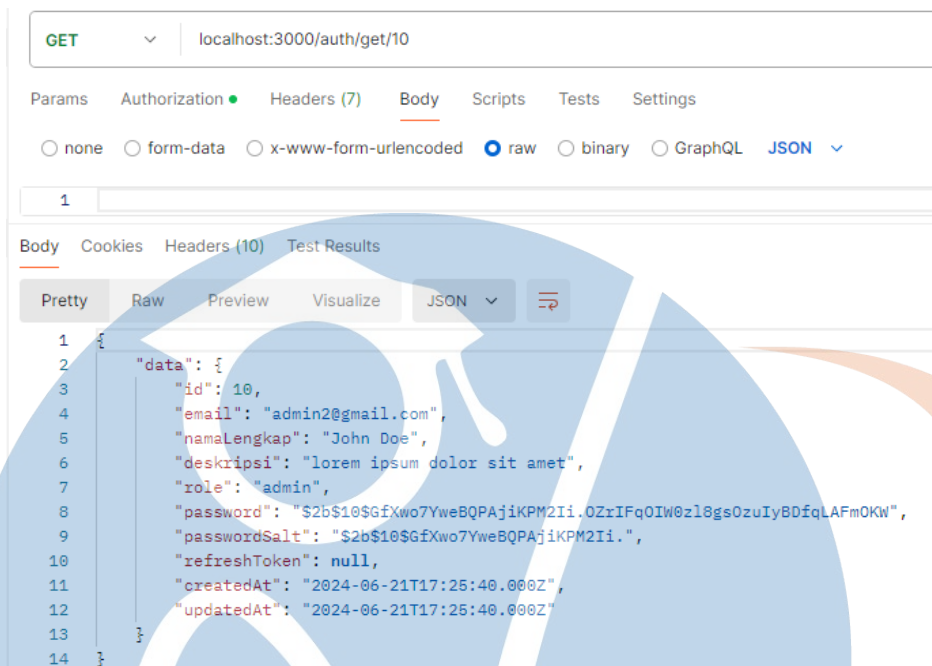
1. Tampilan *input* Data User



Gambar 4. 9 Tampilan Input Data User.

Pada gambar 4.9 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menambahkan data *user* akan menghasilkan *response* “user berhasil dibuat!”.

2. Tampilan Dalam Melihat Data User Berdasarkan ID

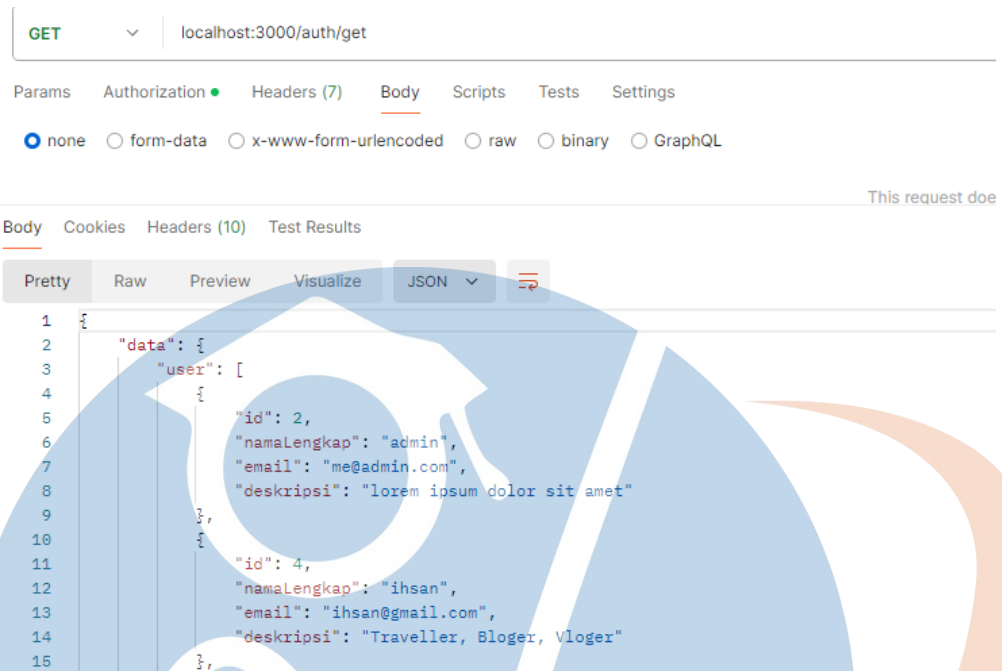


Gambar 4. 10 Tampilan Melihat Data User Berdasarkan Id.

Pada gambar 4.10 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam melihat data *user* berdasarkan id akan menghasilkan *response* dengan menampilkan data *user* berdasarkan id.

3. Tampilan Dalam Melihat Seluruh Data *User*

STT - NF

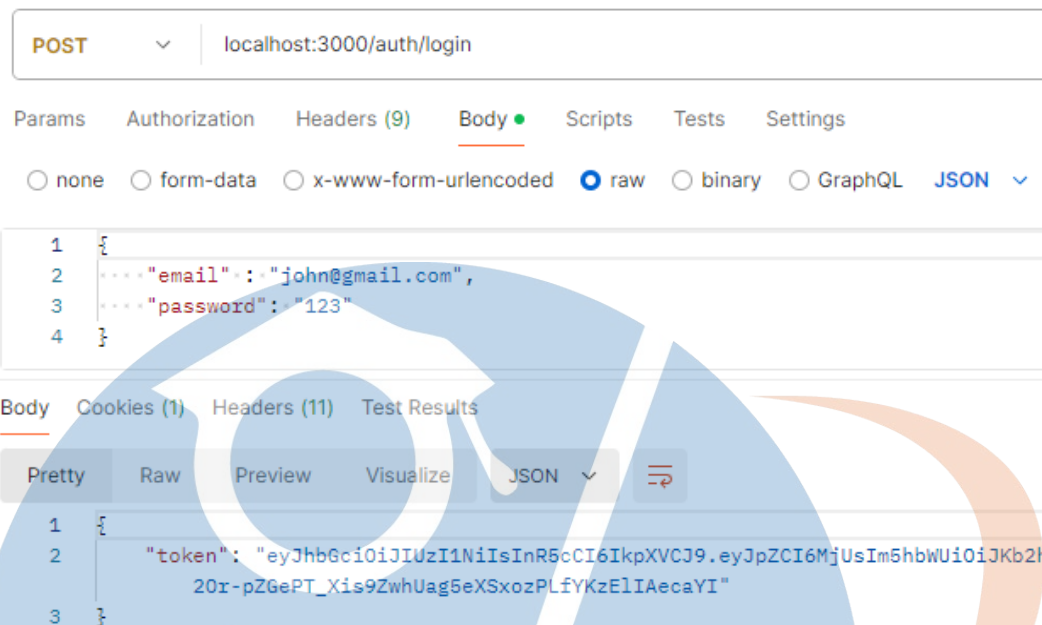


Gambar 4. 11 Tampilan Melihat Seluruh Data User.

Pada gambar 4.11 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin untuk memperlihatkan seluruh data *user* akan menghasilkan *response* dengan menampilkan seluruh data *user*.

4. Tampilan Log-in User

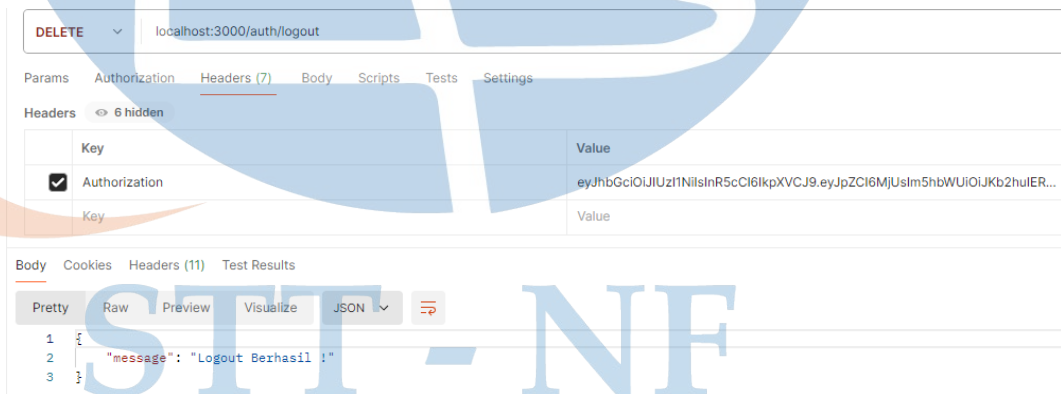
STT - NF



Gambar 4. 12 Tampilan Log-in User

Pada gambar 4.12 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh *user* saat ingin melakukan *log-in* dengan memasukkan *email* dan *password*, akan menghasilkan *response* token *jwt*.

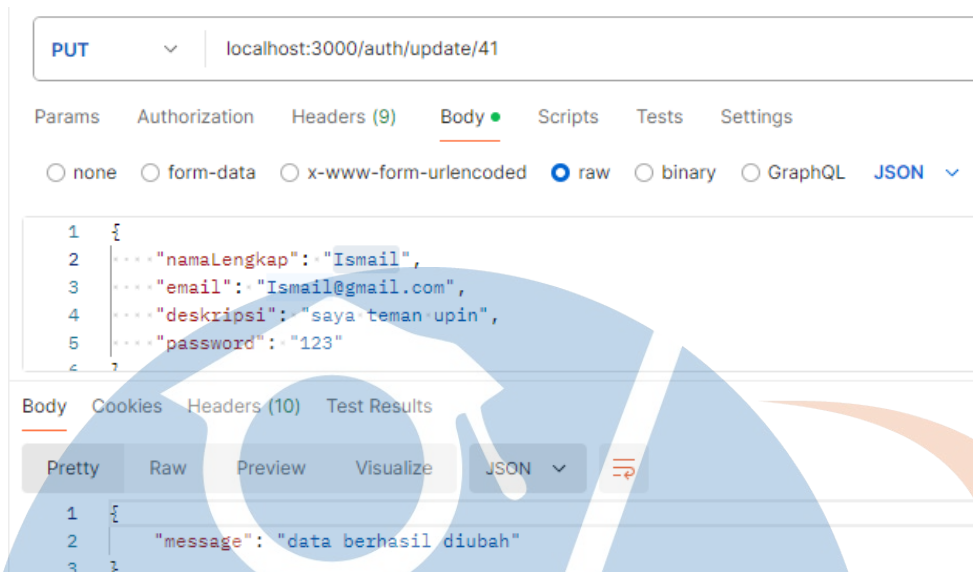
5. Tampilan Log-out User



Gambar 4. 13 Tampilan Log-out User

Pada gambar 4.13 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan *user* saat ingin melakukan *log-out* dengan memasukkan token *jwt* akan menghasilkan *response* “Logout Berhasil !”.

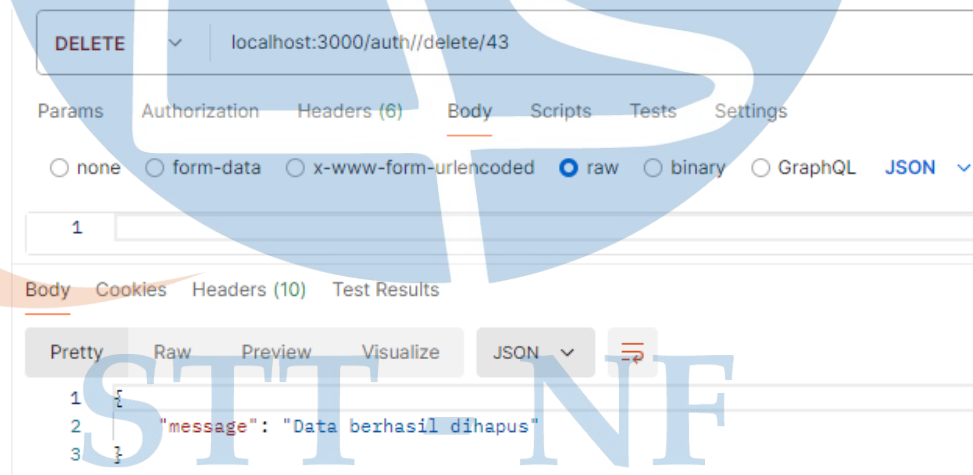
6. Tampilan Mengubah Data User



Gambar 4. 14 Tampilan Mengubah Data User.

Pada gambar 4.14 merupakan tahapan implementasi *request* dan *response*. dari *request API* yang dilakukan oleh admin dalam mengubah data *user* berdasarkan id akan menghasilkan *response* “data berhasil diubah”.

7. Tampilan Menghapus Data User



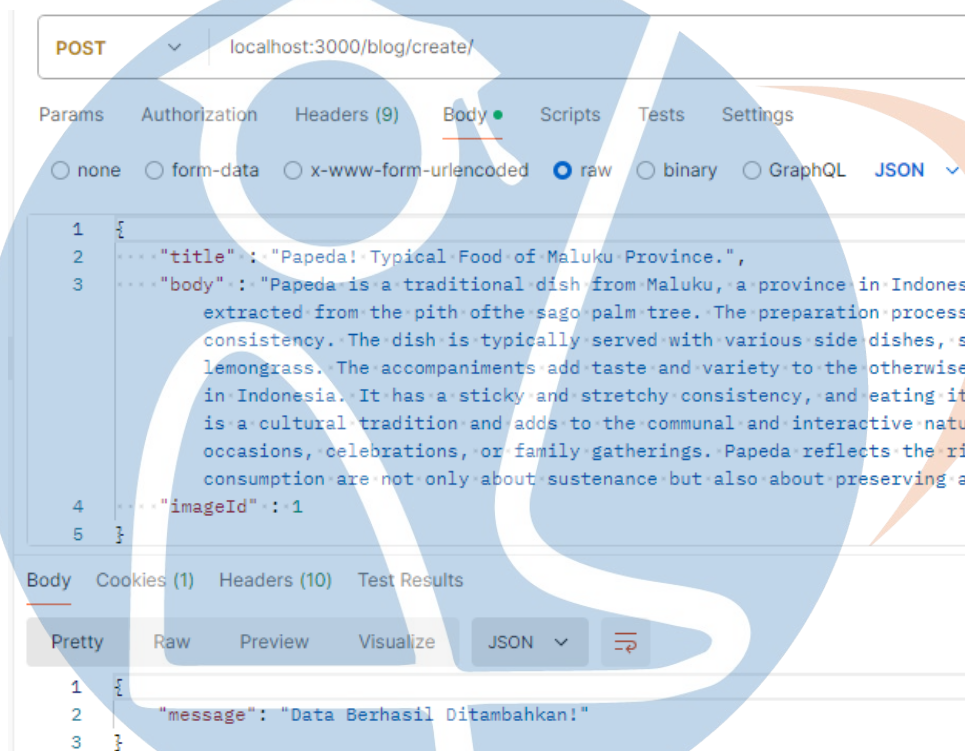
Gambar 4. 15 Tampilan Menghapus Data User.

Pada gambar 4.15 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menghapus data *user* akan menghasilkan *response* “Data berhasil dihapus”.

4.2.2 Antarmuka Kelola Data Blog

Pada tahapan implementasi antarmuka kelola data blog ini, peneliti akan menyajikan uji implementasi sistem layanan *REST API* dalam mengelola data blog. Penjelasan akan difokuskan kepada *request* yang dimasukan dan *response* yang akan diterima saat pengguna berinteraksi dengan sistem.

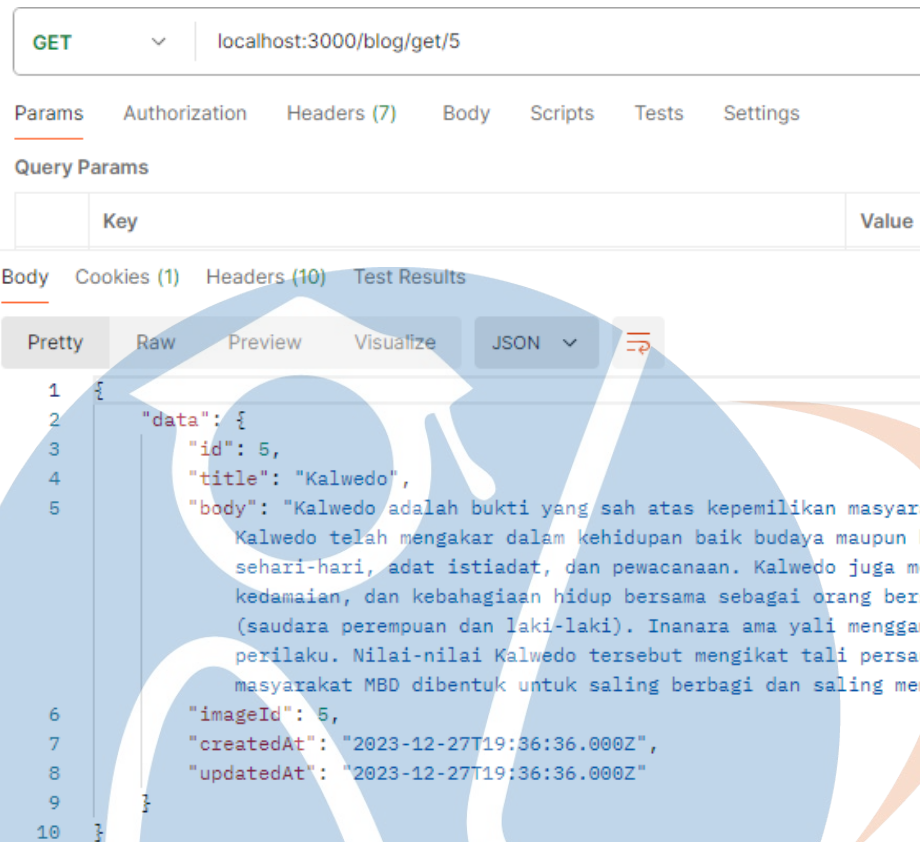
1. Tampilan *Input* Data Blog



Gambar 4. 16 Tampilan *Input* Data Blog.

Pada gambar 4.16 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menambahkan data blog akan menghasilkan *response* “Data Berhasil Ditambahkan!”.

2. Tampilan Dalam Melihat Data Blog Berdasarkan ID



Gambar 4. 17 Tampilan Melihat Data Blog Berdasarkan Id.

Pada gambar 4.17 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menampilkan data blog berdasarkan id akan memberikan *response* dengan menampilkan data blog berdasarkan id.

3. Tampilan Dalam Melihat Seluruh Data Blog

STT - NF

```
GET localhost:3000/blog/get/

Params Authorization Headers (7) Body Scripts Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL

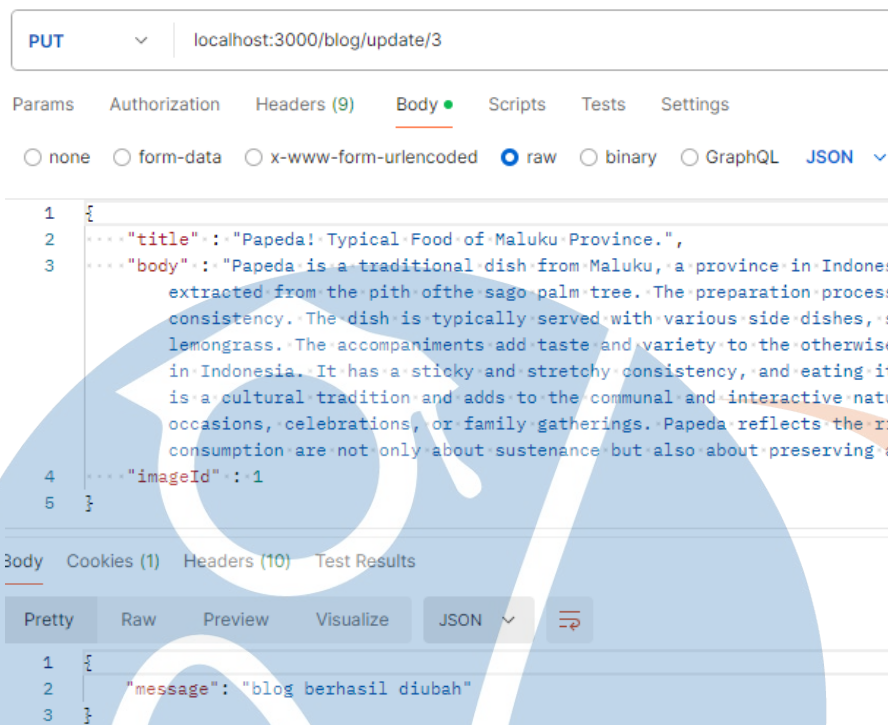
Body Cookies (1) Headers (10) Test Results
Pretty Raw Preview Visualize JSON

1 {
2   "data": [
3     {
4       "id": 2,
5       "title": "Nasi Lapola! Makanan Khas Maluku.",
6       "body": "Nasi Lapola adalah hidangan tradisional yang berasal dari Maluku sehingga memberikan aroma khas pada nasi tersebut. Nasi Lapola biasanya rempah-rempah khas Maluku. Selain itu, beberapa bahan tambahan seperti segar dan pedas. Nasi Lapola memiliki rasa yang lezat dan unik, sering disajikan dalam acara-acara adat, perayaan, atau sebagai hidangan istimewa."
7       "imageId": 2,
8       "createdAt": "2023-12-27T19:32:37.000Z",
9       "updatedAt": "2023-12-27T19:32:37.000Z"
10    },
11   {
12     "id": 3,
13     "title": "Sambal Colo-Colo, Sambal Khas Maluku.",
14     "body": "Sambal Colo Colo adalah sambal khas dari Maluku, Indonesia, yang menggunakan bahan-bahan seperti cabe merah, cabe putih, tomat, dan bahan-bahan lain yang dihaluskan. Sambal Colo Colo sering dijadikan pelengkap dalam hidangan tradisional Maluku dan juga menjadi favorit di kalangan masyarakat Maluku dan juga pengunjung."
15     "imageId": 3,
16     "createdAt": "2023-12-27T19:33:24.000Z",
17     "updatedAt": "2023-12-27T19:33:24.000Z"
18   },
19 ]
20 }
```

Gambar 4.18 Tampilan Menampilkan Seluruh Data Blog.

Pada gambar 4.18 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menampilkan seluruh data blog akan menghasilkan *response* dengan menampilkan seluruh data blog.

4. Tampilan Mengubah Data Blog

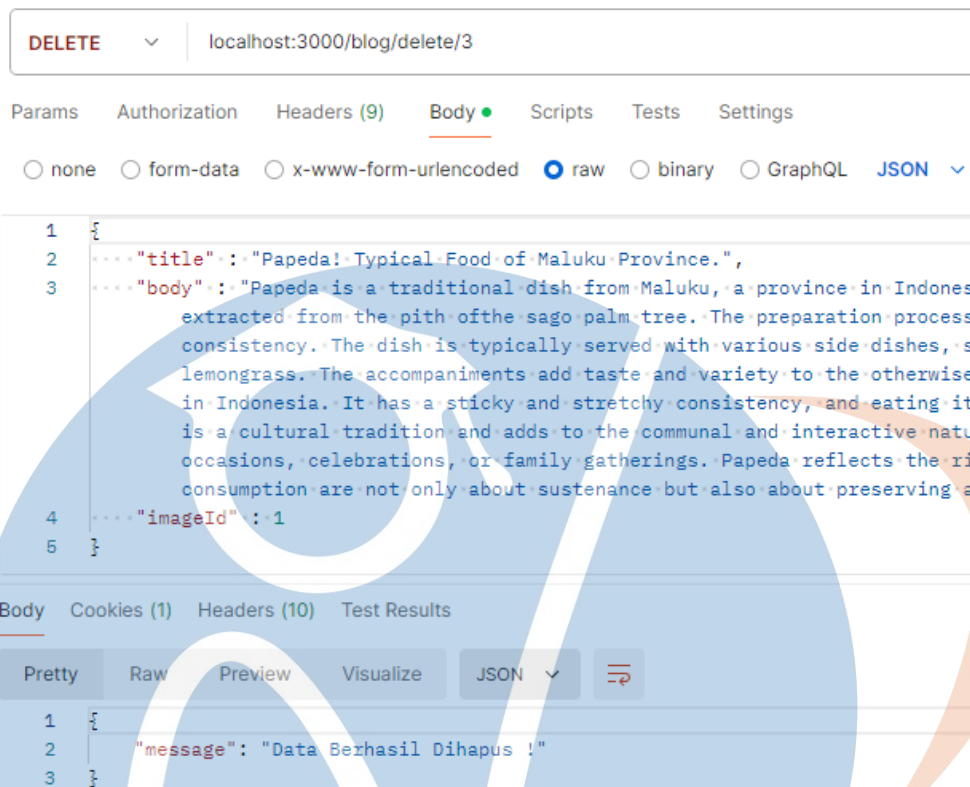


Gambar 4.19 Tampilan Mengubah Data Blog.

Pada gambar 4.19 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam mengubah data blog akan menampilkan *response* “blog berhasil diubah”.

5. Tampilan Menghapus Data Blog

STT - NF



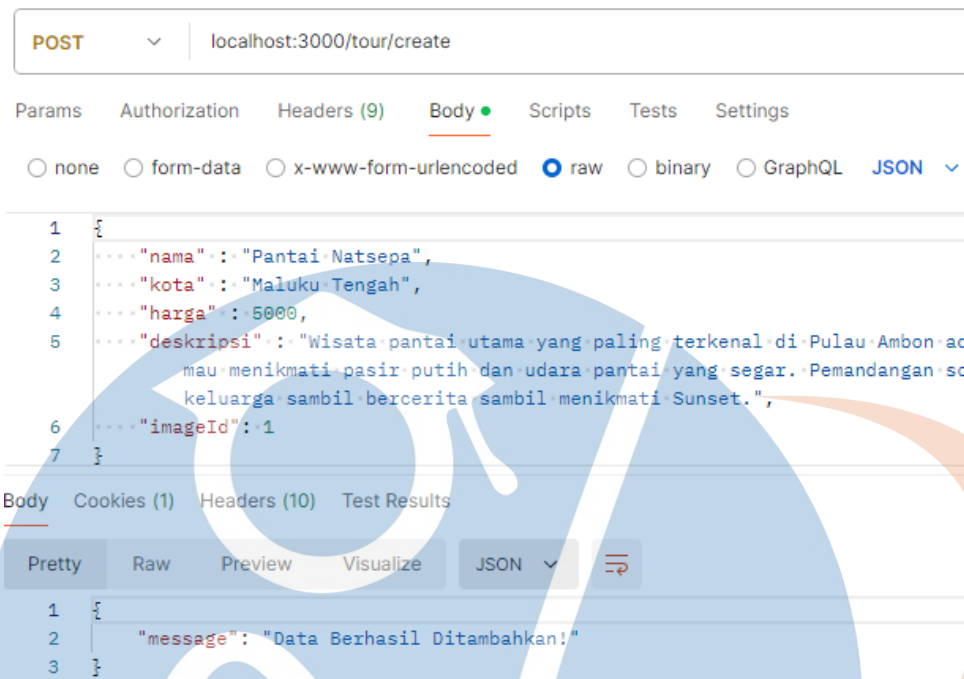
Gambar 4. 20 Tampilan Menghapus Data Blog.

Pada gambar 4.20 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menghapus data blog akan menghasilkan *response* “Data Berhasil Dihapus !”.

4.2.3 Antarmuka Kelola Data Tour

Pada tahapan implementasi antarmuka kelola data *tour* ini, peneliti akan menyajikan uji implementasi sistem layanan *REST API* dalam mengelola data *tour*. Penjelasan akan difokuskan kepada *request* yang dimasukan dan *response* yang akan diterima saat pengguna berinteraksi dengan sistem.

1. Tampilan *input* Data *Tour*

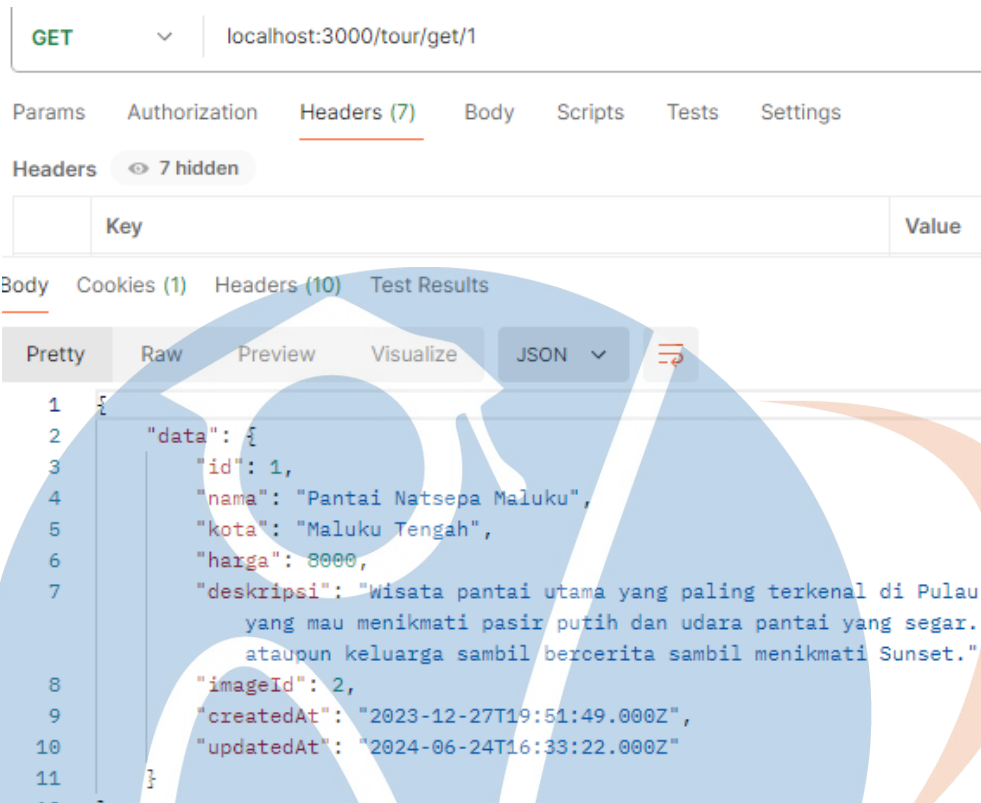


Gambar 4. 21 Tampilan Input Data Tour.

Pada gambar 4.21 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menambahkan data *tour* akan menghasilkan *response* “Data Berhasil Ditambahkan!”.

2. Tampilan Dalam Melihat Data *Tour* Berdasarkan ID

STT - NF

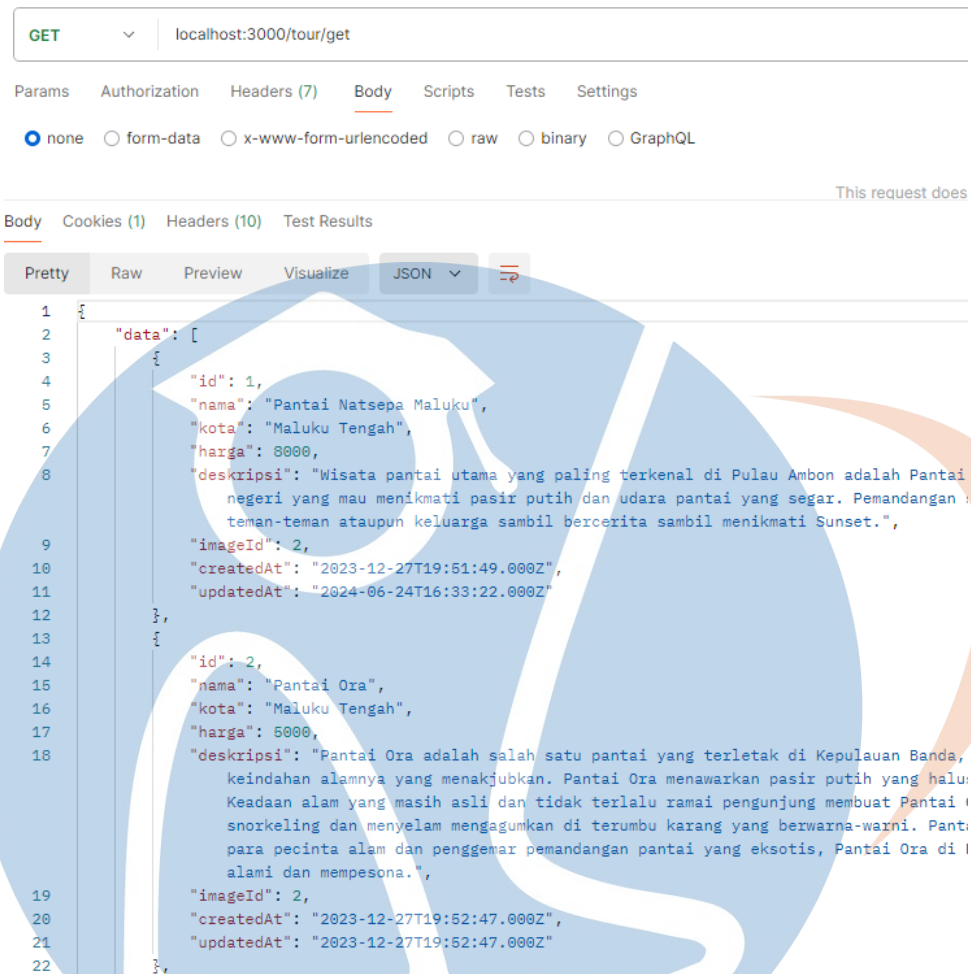


Gambar 4. 22 Tampilan Menampilkan Data Tour Berdasarkan Id.

Pada gambar 4.22 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menampilkan data *tour* berdasarkan id akan menghasilkan *response* dengan menampilkan data *tour* berdasarkan id.

3. Tampilan Dalam Melihat Seluruh Data *Tour*

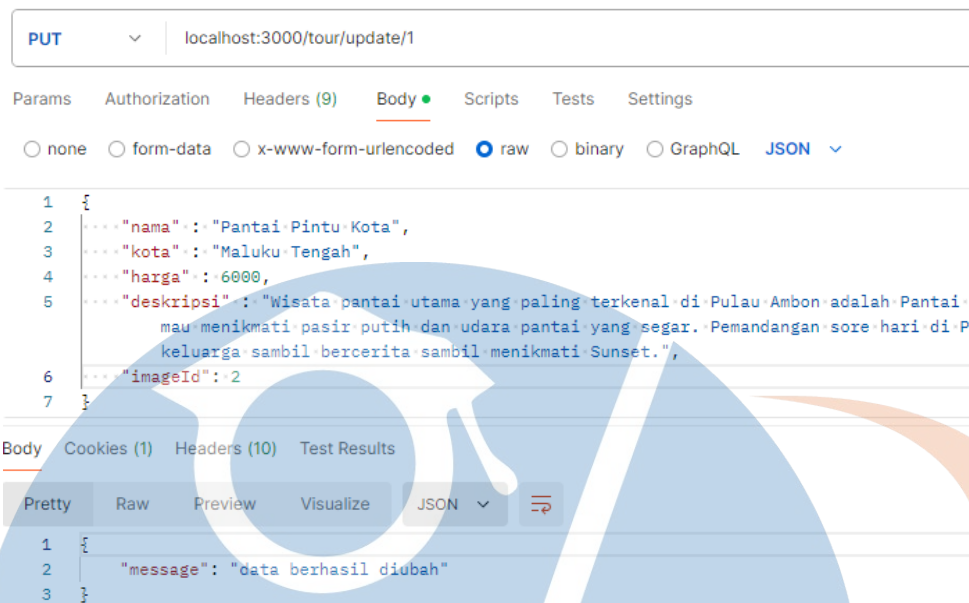
STT - NF



Gambar 4. 23 Tampilan Menampilkan Seluruh Data Tour.

Pada gambar 4.23 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menampilkan seluruh data *tour* akan menghasilkan *response* dengan menampilkan seluruh data *tour*.

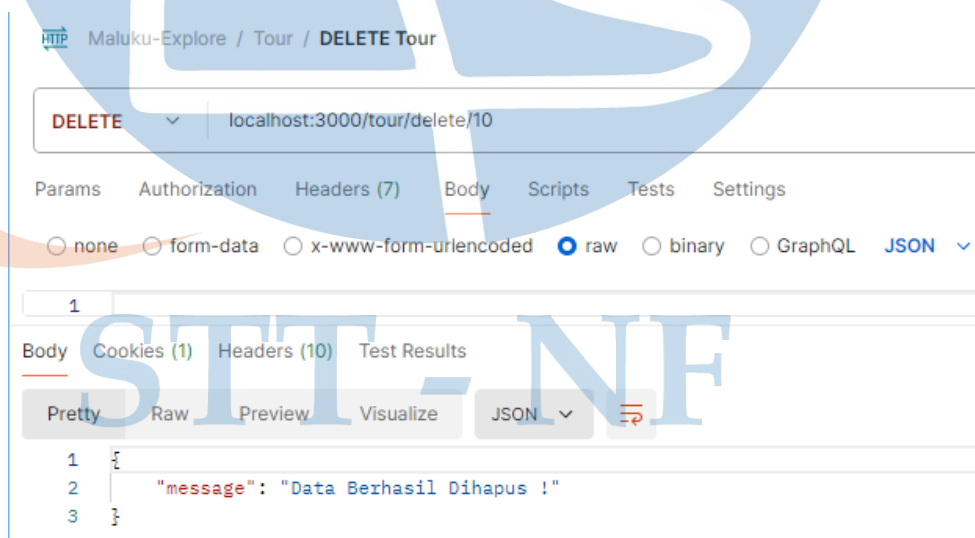
4. Tampilan Mengubah Data Tour



Gambar 4. 24 Tampilan Mengubah Data Tour.

Pada gambar 4.24 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam mengubah data *tour* akan menghasilkan *response* “data berhasil diubah”.

5. Tampilan Menghapus Data Tour



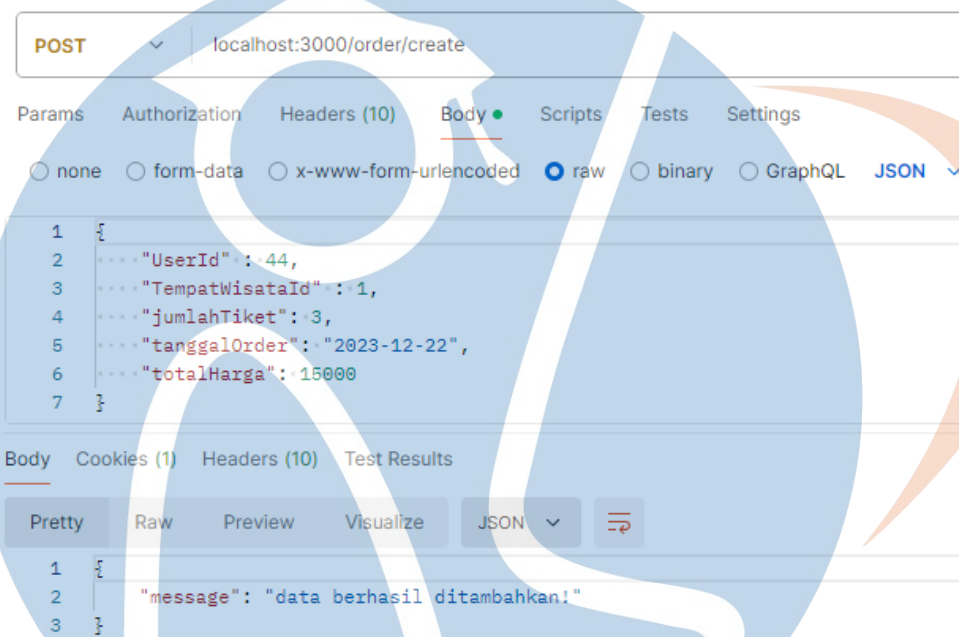
Gambar 4. 25 Tampilan Menghapus Data Tour.

Pada gambar 4.25 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh admin dalam menghapus data *tour* akan menghasilkan *response* “Data Berhasil Dihapus !”.

4.2.4 Antarmuka Kelola Data Order

Pada tahapan implementasi antarmuka kelola data order ini, peneliti akan menyajikan uji implementasi sistem layanan *REST API* dalam mengelola data order. Penjelasan akan difokuskan kepada *request* yang dimasukan dan *response* yang akan diterima saat pengguna berinteraksi dengan sistem.

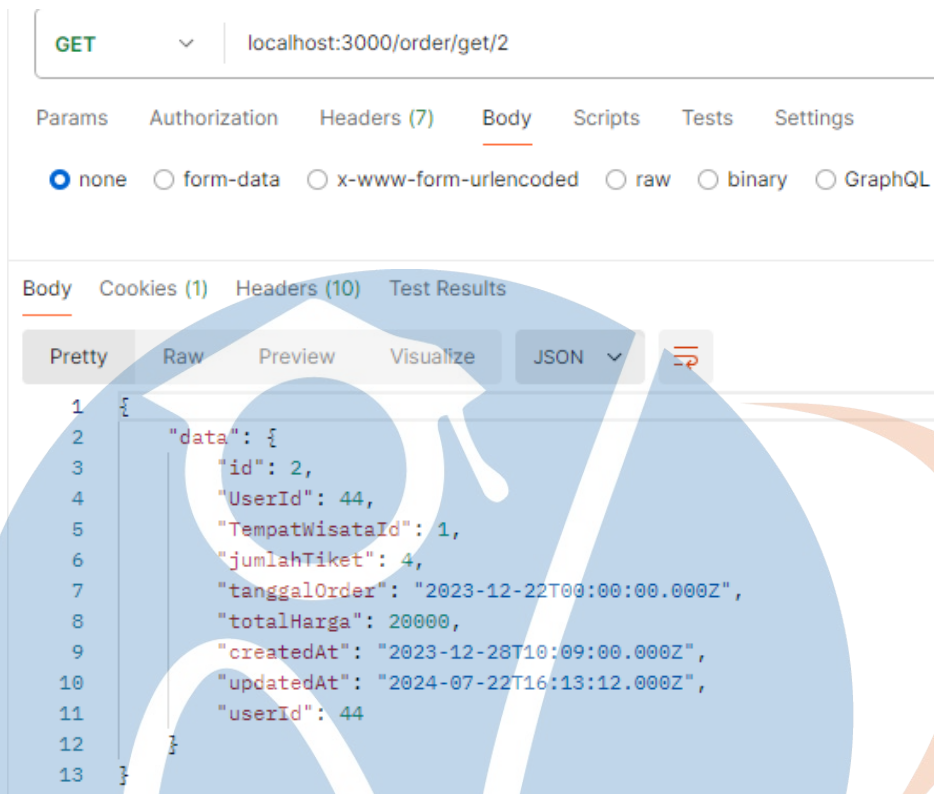
1. Tampilan *input* Data Order



Gambar 4. 26 Tampilan input Data Order.

Pada gambar 4.26 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh *user* dalam menambahkan data order apabila *user* sudah melakukan *log-in* akan menghasilkan *response* “data berhasil ditambahkan!”.

2. Tampilan Dalam Melihat Data Order Berdasarkan ID

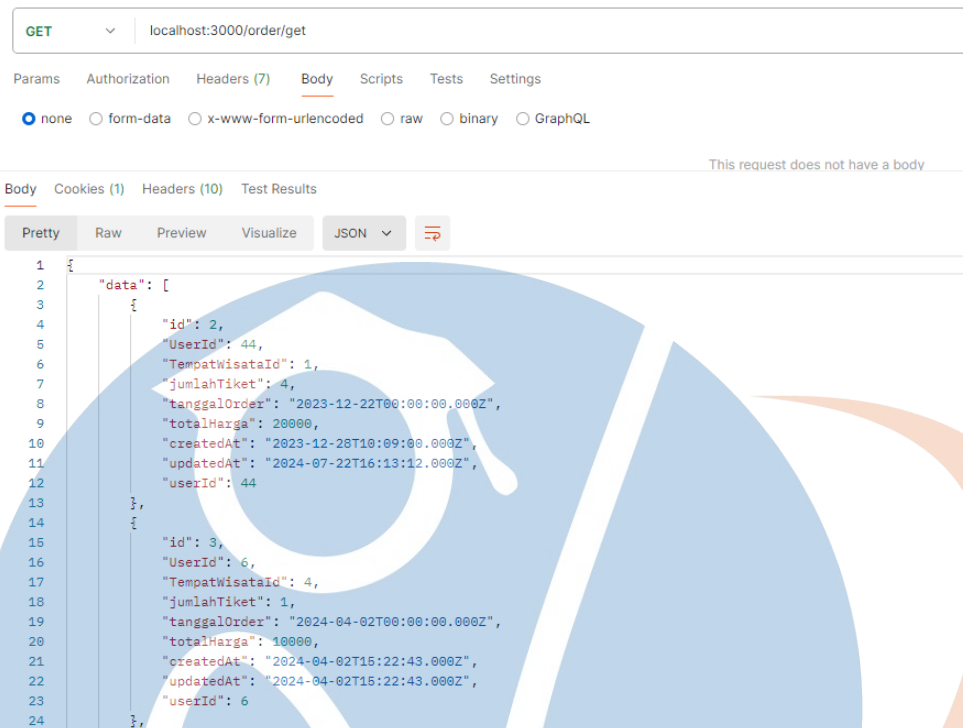


Gambar 4. 27 Tampilan Menampilkan Data Orde Berdasarkan id.

Pada gambar 4.27 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh *user* dalam melihat data order berdasarkan id apabila *user* sudah melakukan *log-in* dan telah meng-order, *response* akan menampilkan data order berdasarkan id.

3. Tampilan Dalam Melihat Seluruh Data Order

STT - NF

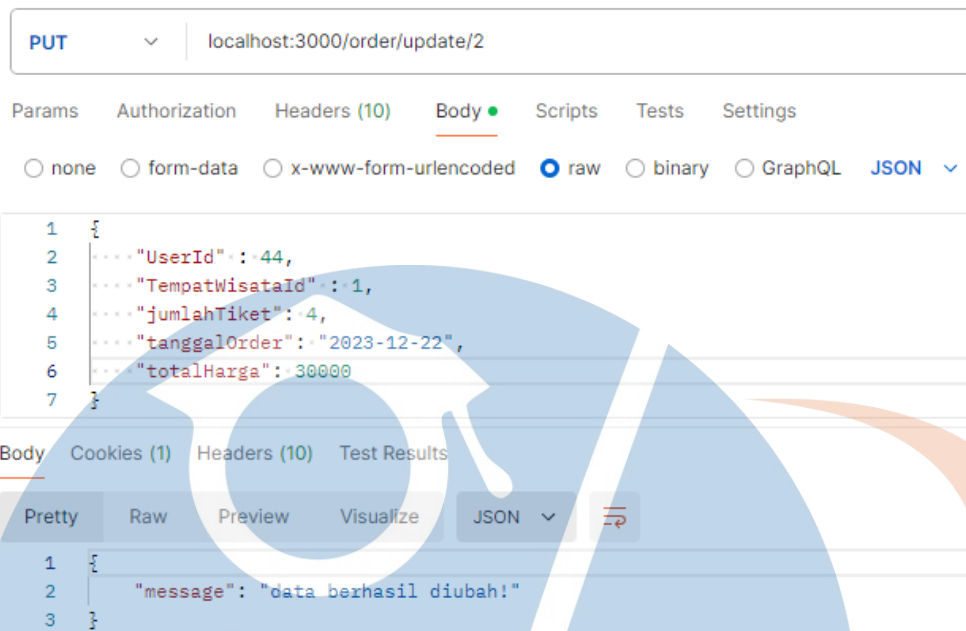


Gambar 4. 28 Tampilan Menampilkan Seluruh Data Order.

Pada gambar 4.28 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh *user* dalam melihat seluruh data order, apabila *user* sudah melakukan *log-in* dan telah meng-order, *response* akan menampilkan seluruh data order.

4. Tampilan Mengubah Data Order

STT - NF

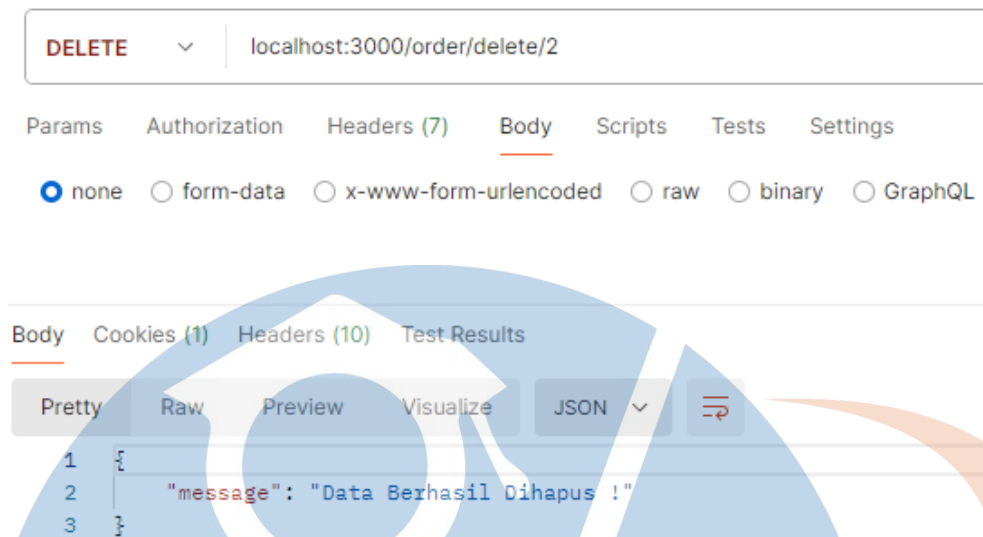


Gambar 4. 29 Tampilan Mengubah Data Order.

Pada gambar 4.29 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh *user* dalam mengubah data order apabila *user* sudah melakukan login dan meng-order, akan menghasilkan *response* “*data berhasil diubah!*”.

5. Tampilan Menghapus Data Order

STT - NF



Gambar 4. 30 Tampilan Menghapus Data Order.

Pada gambar 4.30 merupakan tahapan implementasi *request* dan *response*, dari *request API* yang dilakukan oleh *user* dalam menghapus data order apabila *user* sudah melakukan *log-in*, akan menghasilkan *response* “Data Berhasil Dihapus !”.

4.3 Pengujian Sistem

Pengujian sistem merupakan evaluasi sistem pengujian dimana langkah-langkah untuk menguji dan menilai sistem yang telah dirancang dengan maksud memastikan bahwa aplikasi dapat beroperasi berdasarkan persyaratan yang ditetapkan.

4.3.1 Black Box Testing

Dibawah ini adalah table pengujian yang dilakukan dengan menggunakan *black box testing* yang diuji oleh peneliti berdasarkan sistem wisata Maluku Explore.

Tabel 4. 4 Pengujian Black Box Testing.

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Pengujian
1	User	Register	Post : localhost:3000/auth/register	Akan menghasilkan output: response (201) & message 'User Berhasil Dibuat!'	Sesuai	Penguji 1
		Login	Post: localhost:3000/auth/login	Akan menghasilkan output: response (201)	Sesuai	Penguji 1
		Logout	Delete : localhost:3000/auth/logout	Akan menghasilkan output : response(200), 'Logout Berhasil !'	Sesuai	Penguji 1
2	Halaman Blog website	Create	Post:localhost:3000/blog/create/	Akan menghasilkan output : response (201) & message 'Data Berhasil Ditambahkan !'	Sesuai	Penguji 2

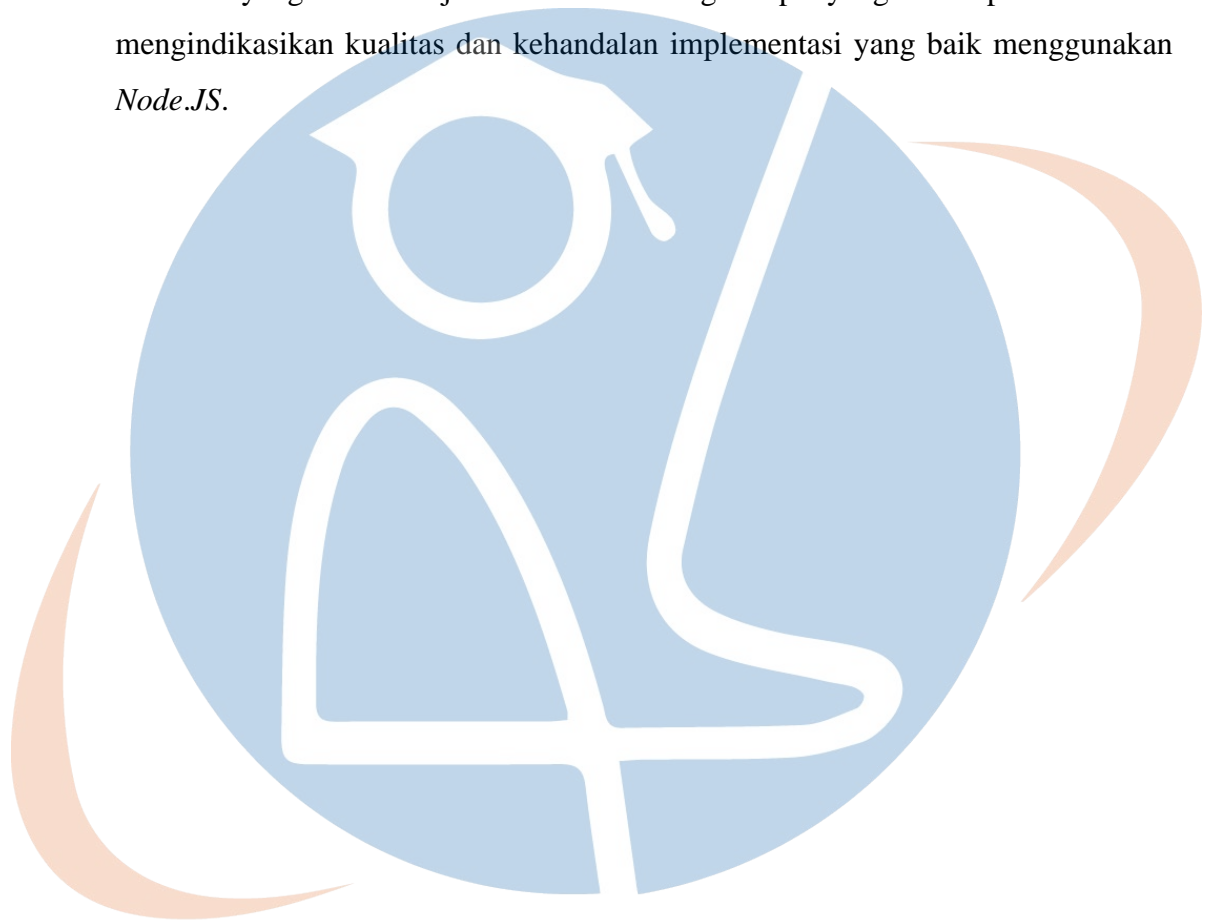
NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Pengujian
		Read	<i>Get:localhost:3000/blog/get/</i>	Akan menghasilkan output : response (200)	Sesuai	Penguji 2
			<i>Get:localhost:3000/blog/get/byId</i>			Penguji 2
		Change	<i>Put:localhost:3000/blog/update/byId</i>	Akan menghasilkan output : response (200) & message 'Blog Berhasil Diubah !'	Sesuai	Penguji 2
		Delete	<i>Delete:localhost:3000/blog/delete/byId</i>	Akan menghasilkan output : response (200) & message 'Data Berhasil Dihapus !'	Sesuai	Penguji 2
3	Halaman wisata	Create	<i>Post:localhost:3000/tour/create</i>	Akan menghasilkan output : response (201) & message 'Data Berhasil Ditambah !'	Sesuai	Penguji 3

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Pengujian
		Read	<i>Get:localhost:3000/tour/get</i>	Akan menghasilkan output : response (200)	Sesuai	Penguji 3
			<i>Get:localhost:3000/tour/get/byId</i>	Akan menghasilkan output : response (200)	Sesuai	Penguji 3
		Change	<i>Put:localhost:3000/tour/update/byId</i>	Akan menghasilkan output : response (200) & message 'Data Berhasil Diubah !'	Sesuai	Penguji 3
		Delete	<i>Delete:localhost:3000/delete/byId</i>	Akan menghasilkan output : response (200) & message 'Data Berhasil Dihapus !'	Sesuai	Penguji 3
4	Pemesanan Tiket	Create	<i>Post:localhost:3000/order/create</i>	Akan menghasilkan output : response (201) & message	Sesuai	Penguji 4

NO.	Fitur	Tahap Pengujian	HTTP Method/End point	Hasil yang diharapkan	Kesimpulan	Pengujian
				'Data Berhasil Ditambahkan !'		
		Read	<i>Get:localhost:3000/order/get/</i>	Akan menghasilkan output : <i>response (200)</i>	Sesuai	Penguji 4
			<i>Get:localhost:3000/order/get/byId</i>	Akan menghasilkan output : <i>response (200)</i>	Sesuai	Penguji 4
		Change	<i>Put:localhost:3000/order/update/byId</i>	Akan menghasilkan output : <i>response (200)</i> & <i>message</i> 'Data Berhasil Diubah !'	Sesuai	Penguji 4
		Delete	<i>Delete:localhost:3000/order/byId</i>	Akan menghasilkan output : <i>response (200)</i> & <i>message</i> 'Data Berhasil Dihapus !'	Sesuai	Penguji

Hasil dari pengujian *black box testing* yang dilakukan oleh 4 pengujian menunjukkan bahwa sistem yang telah dikembangkan menggunakan *Node.js* telah

berhasil dijalankan dengan baik saat di uji fungsionalitasnya. Fitur-fitur utama seperti pengelolaan data user, blog, tour, *login*, *logout* dan lainnya, dapat digunakan sesuai dengan kebutuhan. Pengujian ini dilakukan secara menyeluruh oleh pengembang serta teman pengembang, dan hasilnya menunjukkan bahwa setiap skenario yang telah diuji telah sesuai dengan apa yang diharapkan. Hal ini mengindikasikan kualitas dan kehandalan implementasi yang baik menggunakan *Node.JS*.



STT - NF

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian tugas akhir dengan judul “Rancang Bangun *Backend* Aplikasi Maluku Explore Dengan *Node.JS*: Studi Kasus Wisata Maluku” , maka dapat disimpulkan sebagai berikut:

1. Rancangan pengembangan *backend* untuk sistem informasi wisata dari Maluku Explore mencakup serangkaian tahapan yang terinci, melibatkan analisis sistem yang berjalan, identifikasi permasalahan, analisis kebutuhan, perancangan arsitektur *API*, dan pemilihan teknologi yang tepat. Dengan penggunaan model *UML*, digunakan dalam menggambarkan secara visual struktur dan hubungan antar komponen dalam sistem. Kemudian, implementasi dari rancangan ini dilakukan dalam pengembangan *backend*, memanfaatkan bahasa pemrograman *javascript* dengan dukungan dari kerangka kerja *Express.js* dengan alat bantu *Node.js* sebagai *runtime server*. Sebagai langkah akhir, tahapan pengujian diatur dengan pembuatan rancangan pengujian menggunakan *Black Box Testing* yang diujikan dengan 4 penguji dengan hasil pengujian yang dilakukan pada aplikasi berhasil berjalan dengan baik untuk digunakan.
2. Berdasarkan hasil pengujian aplikasi Maluku Explore, fitur-fitur yang dirancang untuk memudahkan eksplorasi wisata di Maluku telah dievaluasi oleh empat penguji menggunakan metode *Black Box Testing*. Pengujian ini berfokus pada fungsionalitas utama, seperti :
 - a. Blog : Fitur ini dapat diakses oleh pengguna tanpa harus melakukan *log-in* terlebih dahulu. Dengan menampilkan informasi yang relevan dan terstruktur mengenai berbagai informasi destinasi wisata di Maluku, memudahkan pengguna untuk memahami dan merencanakan kunjungan mereka berdasarkan data yang disajikan.
 - b. Tour : Fitur ini memungkinkan pengguna untuk melihat lebih detail tempat wisata yang akan menjadi tempat tujuan dalam berwisata ke Maluku,

dengan informasi yang lebih detail mengenai tempat wisata, serta terdapat informasi mengenai harga tiket untuk masuk kedalam wisata tersebut.

- c. Order : Order adalah sebuah fitur pemesanan tiket yang dapat memudahkan pengguna dalam melakukan pemesanan tiket tempat wisata secara online tanpa harus melakukannya secara langsung, fitur ini juga dapat membantu pengguna untuk melakukan pemesanan tiket dari jumlah tiket yang ingin dipesan sampai tanggal yang bisa ditentukan oleh pengguna.

Pengujian menunjukkan bahwa aplikasi tersebut tidak hanya berfungsi sesuai spesifikasi tetapi juga memberikan kemudahan dalam penggunaan dan meningkatkan pengalaman eksplorasi wisata di Maluku. Hasil dari pengujian fitur aplikasi tersebut telah berhasil berjalan dengan baik, dari fitur blog yang dapat memberikan informasi kepada pengguna tentang macam macam wisata yang ada di wilayah Maluku, fitur *tour* yang dapat memberikan informasi mendetail tentang wisata yang ingin dikunjungi, dan fitur order yang dapat membantu pengguna dalam melakukan pemesanan tiket tempat wisata.

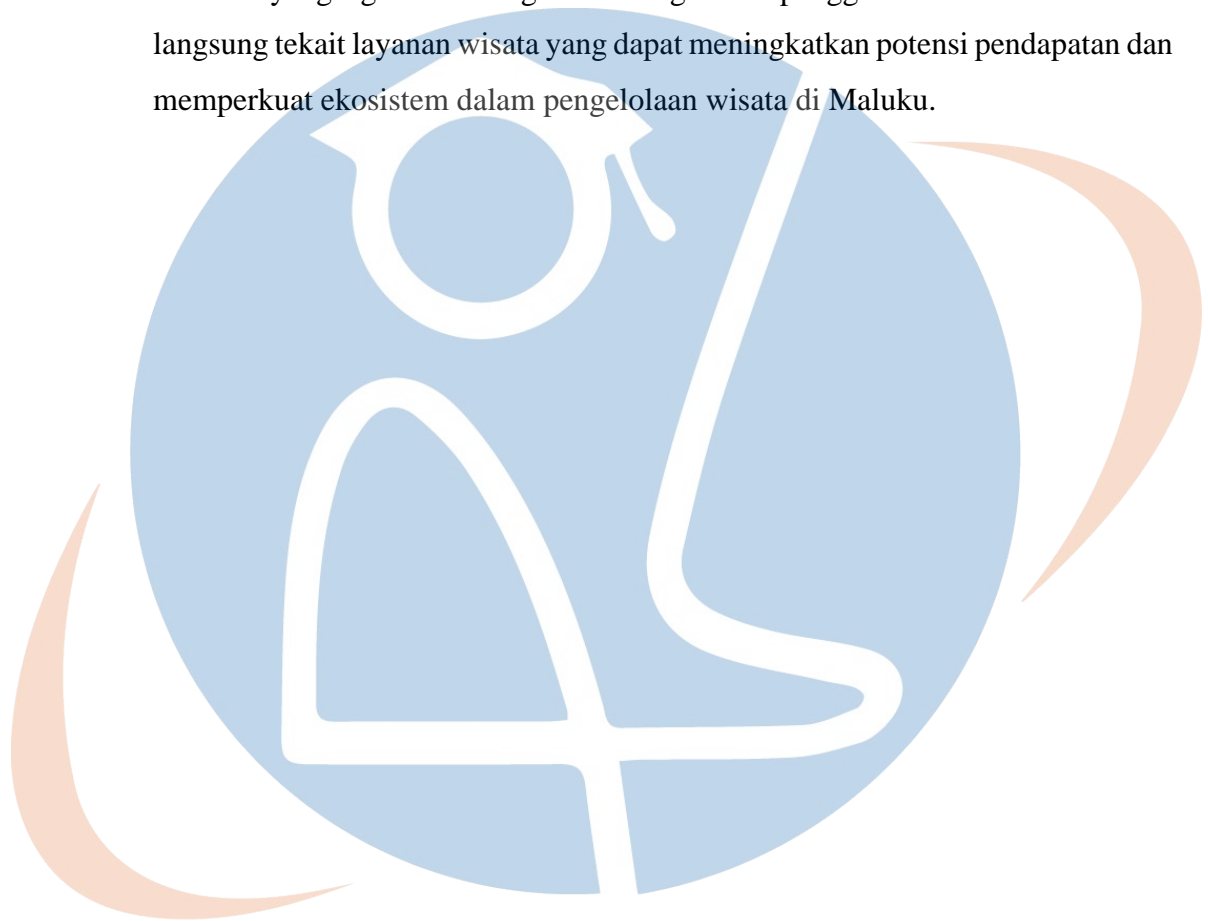
5.2 Saran

Dengan merujuk kepada hasil dari penelitian pengembangan *backend* untuk sistem informasi dari Maluku Explore, terdapat beberapa masukan yang bisa dipertimbangkan sebagai acuan dalam mengembangkan penelitian agar lebih baik. Masukan dari penlusi untuk pengembangan penelitian lebih lanjut sebagai berikut:

1. Saat ini pengembangan *backend* untuk sistem informasi dari Maluku Explore, aplikasi dapat diperluas dengan memasukan pengembangan dan evaluasi pada bagian *fortend*, hal ini akan memberikan manfaat besar bagi pengguna karena visualisasi dan interaksi yang lebih baik akan membuat informasi wisata lebih mudah dipahami.
2. Mempertimbangkan untuk mengembangkan versi *mobile* dari aplikasi ini di masa depan. Mengingat tingginya penggunaan perangkat *mobile* dalam mengakses informasi, meskipun aplikasi saat ini berbasis *website* memiliki keunggulan dalam aksesibilitas di berbagai perangkat. Namun, aplikasi *mobile* sering kali menawarkan pengalaman yang lebih mulus dan optimal, terutama

bagi para wisatawan yang sering menggunakan perangkat mobile dalam mencari informasi wisata.

3. Di masa mendatang, mempertimbangkan untuk mengintergrasikan fitur *payment gateway* dalam aplikasi. Karena fitur *payment gateway* akan memberikan nilai tambah yang signifikan dengan memungkinkan pengguna melakukan transaksi langsung terkait layanan wisata yang dapat meningkatkan potensi pendapatan dan memperkuat ekosistem dalam pengelolaan wisata di Maluku.



STT - NF

DAFTAR PUSTAKA

- [1] A. Surahmat, "RANCANG BANGUN APLIKASI SISTEM PENJUALAN PADA PERCETAKAN CUBIC ART," *JATI*, vol. 7, 2023.
- [2] M. Hamdan Romadhon and Y. Yudhistira, "Sistem Informasi Rental Mobil Berbasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus : CV Kopja Mandiri," 2021. [Online]. Available: www.journal.peradaban.ac.id
- [3] L. 'Rizkinaswara, "Pentingnya Teknologi dalam Sektor Pariwisata," *INDONESIA TERKONEKSI*.
- [4] Hasanuddin, H. Asgar, and B. Hartono, "RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN," *RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN*, vol. 4, pp. 8–14, 2021, doi: Prefix 10.51401.
- [5] S. Mariko, "APLIKASI WEBSITE BERBASIS HTML DAN JAVASCRIPT UNTUK MENYELESAIKAN FUNGSI INTEGRAL PADA MATA KULIAH KALKULUS," *Jurnal Inovasi Teknologi Pendidikan*, vol. 6, no. 1, pp. 80–91, 2019, doi: 10.21831/jitp.v6.1.22280.
- [6] O. Karlsson, "A Performance comparison Between ASP.NET Core and Express.js for creating Web APIs," *JONK " OPING*, 2021.
- [7] S. KUMARI Behera *et al.*, "MongoDB integration with Python and Node.js, Express.js," 2024, doi: 10.1109/ICAECT60202.2024.10460546.
- [8] R. E. P. Rizaldy, U. A. Ahmad, and B. Dirgantoro, "Implementasi REST API Pada Pengembangan Aplikasi Backend Untuk Platform Kursus Online (Growup)," *e-proceeding of Engineering*, vol. 10, no. 2355–9365, p. 404, 2023.
- [9] N. H. Arzt, "Application Programming Interface (API) for Immunization Information Interoperability Author CASE STUDY," 2020. [Online]. Available: <http://journals.ke-i.org/index.php/mra>

- [10] N. Hasanati, E. Khudzaeva, E. Nurmiyati, Elpawati, and N. Hakiem, “Dashboard of Quantitative Data for Accreditation of Undergraduate Level Study Program using Application Programming Interface (Case Study on University in Indonesia),” *European Union Digital Library*, 2020, doi: 10.4108.
- [11] M. Massé, *REST API Design Rulebook*. United States of America: O’Reilly Media, Inc., 2011. [Online]. Available: www.allitebooks.com
- [12] N. Barsoti and D. Gibertoni, “IMPACTO QUE O SEQUELIZE TRAZ PARA O DESENVOLVIMENTO DE UMA API CONSTRUÍDA EM NODE.JS COM EXPRESS.JS,” *Revista Interface Tecnológica*, vol. 17, no. 2, pp. 231–243, Dec. 2020, doi: 10.31510/infa.v17i2.964.
- [13] K. F. Khaliq, “PENGAMANAN DATA AKTA DENGAN METODE AES BERBASIS CLOUD COMPUTING.”
- [14] Hyams. Rebecca and Pilko. Tamara, “‘You could use the API!’: A Crash Course in Working with the Alma APIs using Postman,” no. 24, 2022.
- [15] J. Sains *et al.*, “Yayasan Insan Cipta Medan APLIKASI BUKU TAMU MENGGUNAKAN FITUR KAMERA DAN AJAX BERBASIS WEBSITE PADA KANTOR DISPORA KOTA MEDAN,” *SITek*, 2022.
- [16] T. Informatika, S. Tinggi, T. N. Fikri, S. Munir, S. Si, and M. Kom, “RANCANG BANGUN APLIKASI PENGELOLAAN PRAKTIK KERJA LAPANGAN BERBASIS WEB MENGGUNAKAN MVC FRAMEWORK STUDI KASUS SMK TARUNA BHAKTI DEPOK Sirin Mazaya Rochmah Shahab,” *Jurnal Teknologi Terpadu*, vol. 5, no. 1, 2019.
- [17] R. N. Thakur and U. S. Pandey, “The Role of Model-View Controller in Object Oriented Software Development,” *Nepal Journal of Multidisciplinary Research*, vol. 2, no. 2, pp. 1–6, Nov. 2019, doi: 10.3126/njmr.v2i2.26279.
- [18] Y. Bachtiar, D. Anjani, and D. Novianti, “RANCANG BANGUN E-RTDALAM UPAYA MENINGKATKAN KEAMANAN, KETERTIBAN, DAN KERUKUNAN HIDUP ANTAR WARGA,” *Jurnal Teknologi Terpadu*, vol. 8, pp. 78–85, 2022.

- [19] D. Wira, T. Putra, and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," vol. 7, no. 1, 2019.
- [20] F.- Sonata, "Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer," *Jurnal Komunika : Jurnal Komunikasi, Media dan Informatika*, vol. 8, no. 1, p. 22, Jun. 2019, doi: 10.31504/komunika.v8i1.1832.
- [21] M. Najib, D. Satria, M. Y. Ridho, and I. Mahfud, "SISTEM INFORMASI PEMERINGKATAN PRESTASI SISWA BERBASIS WEB PADA SMK PELITA PESAWARAN," *Journal of Technology and Social for Community Service (JTSCS)*, vol. 3, no. 1, pp. 114–122, 2022, [Online]. Available: <https://ejurnal.teknokrat.ac.id/index.php/teknoabdimas>
- [22] F. Arif Novianto and dan Hari Purwanto, "PERANCANGAN SISTEM INFORMASI LAND TRANSPORTATION ASSISTANCE TAXI PUSKOPAU PADA BANDARA XYZ."
- [23] D. Titian Lestari and D. Ayu Megawaty, "SISTEM INFORMASI PKK BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS: KAMPUNG PURWOEJO)," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 3, no. 2, pp. 244–253, 2022, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>
- [24] R. Sunantoro and D. Anubhakti, "ANALISA DAN RANCANGAN E-COMMERCE PADA TOKO ANGSANA," 2019.
- [25] D. Ahrizal, M. Khaerul Miftah, R. Kurniawan, and T. Zaelani, "Pengujian Perangkat Lunak Sistem Informasi Peminjaman PlayStation dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing," vol. 5, no. 1, 2020, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika73>
- [26] G. Putu Mahendra Putra, A. Tenriawaru, P. Studi Ilmu Komputer, F. Matematika dan Ilmu Pengetahuan Alam, and U. Halu Oleo, "Rancang

Bangun Virtual Assistant Chatbot Menggunakan Node.Js pada Layanan Sistem Informasi Akademik,” 2023.

- [27] L. Ariyanti, M. Najib, D. Satria, and D. Alita, “SISTEM INFORMASI AKADEMIK DAN ADMINISTRASI DENGAN METODE EXTREME PROGRAMMING PADA LEMBAGA KURSUS DAN PELATIHAN,” 2020. [Online]. Available:

<http://jim.teknokrat.ac.id/index.php/sisteminformasi>

- [28] N. Nugroho, R. Napianto, G. Adithama, F. Keguruan, I. Pendidikan, and U. Bina Bangsa, “Pengembangan Sistem E-Procurement Pada SMK Yadika Baturaja dengan Pendekatan Extreme Programming,” 2021.



STT - NF