

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Bagian ini memuat kajian dan uraian sistematis tentang informasi hasil penelitian yang pernah dilakukan peneliti lain dalam pustaka dan menghubungkannya dengan masalah penelitian. Tinjauan pustaka membahas tentang metode yang digunakan dalam penelitian aplikasi. Hal tersebut digunakan sebagai dasar membuat struktur landasan teori. Peneliti telah melakukan studi literatur terhadap 10 jurnal yang berkaitan dengan tema penelitian yang dikerjakan.

Penelitian pertama dengan judul “Pengembangan Aplikasi Pemesanan Tiket *Travel* Berbasis *Web* dengan Optimasi Jalur Penjemputan Penumpang (Studi Kasus: Beruang *Travel*)” oleh Della Fauziah, Fajar Pradana, dan Achmad Arwan memiliki permasalahan pada pengujian kebutuhan non-fungsional dari aplikasi pemesanan tiket *travel* berbasis *web* dengan optimasi pemilihan jalur penjemputan menggunakan algoritma *dijkstra*. Tujuan dari penelitian ini adalah mengembangkan sistem dalam usaha untuk meningkatkan kualitas pelayanan pihak *travel* dan memudahkan konsumen dalam mendapatkan informasi jadwal keberangkatan dengan cepat dan jelas, serta memberikan kenyamanan penjemputan dan pengantaran bagi konsumen dalam hal efisiensi waktu. Metode yang digunakan adalah *waterfall*. Hasil dari penelitian ini adalah pengujian validasi dihasilkan angka 100% *valid* pada setiap kebutuhan beserta kondisi alternatifnya. Dalam pengujian *compatibility* didapatkan hasil bahwa terdapat 2 *critical issues* pada 2 *browser* yang berbeda [1].

Penelitian kedua dengan judul “Implementasi Arsitektur *Microservice* Pada Aplikasi *Online Travel* Tourinc” oleh Muhamad Danil Rafiqi, Eko Subyantoro, dan Dewi Kania W. memiliki permasalahan pada PT Tourinc membutuhkan aplikasi yang kompleks untuk menjalankan bisnisnya di dalam

penjualan tiket pesawat, hotel, dan kereta api. Aplikasi tersebut harus mengakses *REST API* dari berbagai perusahaan seperti PT. KAI dan maskapai penerbangan, serta mengotomatisasi proses konfirmasi pembayaran. Tujuan dari penelitian ini adalah mempermudah *monitoring* pendapatan dari penjualan tiket *bus* secara *real-time* dengan implementasi *web service*. Metode yang digunakan adalah *WSIM (Web Service Implementation Methodology)*. Hasil dari penelitian ini adalah telah berhasil menyelesaikan *microservice online travel tourinc* dengan melakukan *Blackbox* testing menggunakan aplikasi *Insomnia* dan metode *boundary value analysis* [8].

Penelitian ketiga dengan judul “*Booking Tickets Online At PT CBB Using The Laravel Framework*” oleh Edy Rakhmat, Raden Kania, dan Yarham Fitrohiman Pahlawi. memiliki permasalahan pada PT Cahaya Berkah Bersama adalah pemesanan tiket wisata tidak efektif karena pelanggan harus menghubungi atau datang langsung ke kantor untuk memesan paket wisata dan mengetahui rincian paket wisata yang mereka inginkan, panggilan dari pelanggan seringkali terlewat karena admin layanan hanya satu orang, *admin* sering terlambat mengirimkan tiket wisata kepada pelanggan. Tujuan dari penelitian ini adalah mengatasi masalah kurang efektifnya pemesanan tiket wisata saat ini di PT Cahaya Berkah Bersama. Metode yang digunakan adalah *Waterfall*. Hasil dari penelitian ini adalah pengujian *Blackbox* diperoleh tingkat keberhasilan 80-90% lebih mudah, berhasil guna dan tepat waktu dibanding dengan pemesanan tiket sebelumnya saat dilakukan pengujian [9].

Penelitian keempat dengan judul “Perancangan Sistem Informasi Reservasi Kamar Hotel Berbasis *Web* Dengan Metode *RUP (Rational Unified Process)*” oleh Ita Dewi Sintawati, dan Suminten. memiliki permasalahan pada hotel yang masih menggunakan sistem reservasi manual, yang mengakibatkan banyak masalah seperti tamu tidak mendapatkan kamar yang diinginkan, bahkan harus membatalkan reservasi, dan tidak ada *back-up* pencatatan reservasi. Tujuan dari penelitian ini adalah untuk merancang sistem informasi reservasi hotel berbasis *online* yang dapat memudahkan

tamu melakukan pemesanan secara online tanpa harus datang langsung ke hotel. Metode yang digunakan adalah *RUP (Rational Unified Process)*. Hasil dari penelitian ini adalah Berhasil merancang sistem informasi dengan metode *RUP (Rational Unified Process)* [10].

Penelitian kelima dengan judul “Pengembangan Aplikasi Pengolahan Data Pemesanan Tiket Pesawat” oleh Pina Aprillia. memiliki permasalahan pada PT Elendra Tour and Travel masih menggunakan sistem manual dalam pencatatan data customer dan pemesanan tiket pesawat. Hal ini dapat menyebabkan kesalahan dan kehilangan data. Tujuan dari penelitian ini adalah merancang dan mengembangkan sebuah sistem informasi pengolahan data pemesanan tiket pesawat pada PT Elendra Tour and Travel. Sistem yang dirancang akan meminimalkan kesalahan dan kehilangan data dengan mengeliminasi kegiatan pencatatan ulang dalam laporan dan dengan menyimpan data dalam sebuah database. Metode yang digunakan adalah *Agile Extreme Programming* Hasil dari penelitian ini diuji menggunakan *Blackbox* hasilnya adalah sistem ini dapat memudahkan dalam pencatatan data pemesanan tiket pesawat dimana tidak perlu pencatatan ulang untuk laporannya serta adanya penyimpanan di *database* dapat meminimalisir kehilangan dan kerusakan data jika suatu saat diperlukan kembali [3].

Penelitian keenam dengan judul “Rancang Bangun Sistem Informasi *Travel* Berbasis *Web* Pada *Jaya Abadi Travel*” oleh Sintia Ika Anggraini. memiliki permasalahan pada *Jaya Abadi Tour & Travel* yang sering ditemui yaitu bilamana membutuhkan data *customer* selalu membutuhkan waktu yang lama, hal ini dikarenakan data tersebut belum tersimpan pada *database* tetapi masih berupa dokumen arsip sehingga dapat menghambat kinerja perusahaan. Selain itu, *customer* harus datang ke *Jaya Abadi Tour & Travel* untuk perihal pemesanan hingga transaksi pembayaran baik untuk *customer* yang dari dalam atau luar kota Sragen. Tujuan dari penelitian ini adalah untuk mengembangkan sistem informasi *Jaya Abadi Tour & Travel* yang kurang efisien. Metode yang digunakan adalah Pengembangan. Hasil dari penelitian ini adalah pelanggan dapat melakukan pemesanan atau transaksi lain selama

24 jam sehari sepanjang tahun dari nyaris setiap lokasi dan *Website* Sistem Informasi *Travel* Berbasis *Web* ini membuat transaksi lebih efisien dari segi biaya, tenaga dan waktu [11].

Penelitian ketujuh dengan judul “Rancang Bangun *Backend* Sistem Manajemen *Gymship* Berbasis *Website* Pada *Alterra Academy*” oleh Muhammad Azril Wijaya, Muhammad Faqih Dzulqarnain, Safri Adam. Memiliki permasalahan di *Alterra Academy* memerlukan sistem informasi terkomputerisasi untuk mengurangi kesalahan manusia dan meningkatkan efisiensi. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun backend sistem manajemen *Gymship* berbasis website sebagai media untuk mendapatkan data dan informasi keanggotaan bagi *member* dan *admin*. tujuan dari penelitian ini adalah membantu masyarakat yang ingin berolahraga dengan mudah mendapatkan jadwal latihan melalui pendaftaran member di website *Gymship*. Penelitian ini juga menggunakan *Entity Relationship Diagram* untuk merancang data yang dibutuhkan, serta *Swagger API Documentation* sebagai dokumentasi data yang dapat dibaca oleh *programmer*. Selain itu, penelitian ini juga melibatkan pengujian data dengan menggunakan *test case* yang dirancang berdasarkan *Entity Relationship Diagram*. hasil dari penelitian ini adalah pembuatan website bernama *Gymship*, yang bertujuan untuk mempermudah perusahaan dalam mempromosikan produk jasa mereka untuk mendapatkan pelanggan baru [4].

Penelitian kedelapan dengan judul “Pemanfaatan *UML (Unified Modeling Language)* Dalam Perencanaan Sistem Penyewaan Baju Adat Berbasis *Website*” oleh Elis dan Apriade Voutama memiliki permasalahan pada bagaimana memudahkan penyewaan baju adat dengan memanfaatkan teknologi dan komunikasi yang efisien, sehingga konsumen tidak perlu datang langsung untuk mengecek ketersediaan baju adat. Tujuan dari penelitian ini adalah untuk mengembangkan perangkat lunak yang dapat memfasilitasi penyewaan baju adat secara *online*, memudahkan komunikasi antara penyewa dan penyedia, serta memberikan rekomendasi diagram-diagram *UML* yang dapat digunakan oleh pengembang perangkat lunak

terkait usaha jasa penyewaan. Metode yang digunakan adalah menggunakan *SDLC (System Development Lifecycle)* Hasil dari penelitian ini adalah pengembangan perangkat lunak yang memungkinkan penyewaan baju adat secara efisien melalui sistem informasi *online* [12].

Penelitian kesembilan dengan judul “Pengujian *Blackbox* Dan *Whitebox* Sistem Informasi Parkir Berbasis *Web*” oleh Anisya Caty Praniffa, Alfi Syahri, Fitriani Fariha, Qhoiril Aldi Giansyah, Muhammad Luthfi Hamzah memiliki permasalahan pada kurangnya pengaturan dan pengendalian yang baik terhadap sistem parkir di Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA RIAU). Selain itu, juga belum ada data terkontrol mengenai kendaraan mahasiswa, dosen, dan *staff* yang dapat digunakan untuk evaluasi dalam pengelolaan parkir. Satpam yang bertugas juga menghadapi kendala dalam mengatur keluar masuknya kendaraan di kampus. Tujuan dari penelitian ini adalah merancang dan membuat sistem parkir berbasis *web* yang tertata dengan baik dan terkontrol di UIN SUSKA RIAU. Selain itu, penelitian ini juga bertujuan untuk melakukan pengujian terhadap sistem tersebut menggunakan *blackbox testing* dan *whitebox testing* sebelum diimplementasikan. Metode yang digunakan adalah *Blackbox testing* dilakukan untuk menguji fungsionalitas dan kecocokan sistem dari perspektif pengguna, sedangkan *whitebox testing* dilakukan untuk menguji keandalan dan struktur *internal* sistem. Hasil dari penelitian ini adalah pengujian *blackbox testing* dan *whitebox testing* pada sistem informasi parkir berbasis website di UIN SUSKA RIAU menunjukkan bahwa tidak ditemukan permasalahan pada sistem yang dibangun. Sistem yang telah dirancang dan dibuat berhasil dan *valid* sesuai dengan kebutuhan dan fungsi yang diharapkan. Setiap pengguna sistem memiliki hak akses sesuai dengan perancangan sistem [13].

Tabel 2.1 Penelitian Terdahulu

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
1.	Pengembangan Aplikasi Pemesanan Tiket <i>Travel</i> Berbasis <i>Web</i> dengan Optimasi Jalur Penjemputan Penumpang (Studi Kasus: Beruang <i>Travel</i> ) [1]	Penelitian ini membandingkan penggunaan metode <i>waterfall</i> dalam pengembangan aplikasi dan penggunaan <i>database MySQL</i> serta bahasa <i>PHP</i> dengan <i>framework CodeIgniter</i> dalam tahap implementasi.	Penelitian ini menyoroti adanya peningkatan persaingan dalam dunia bisnis akibat kemajuan teknologi <i>web</i> , terutama dalam bidang layanan transportasi.	Diskusi mengenai keterbatasan atau tantangan yang mungkin dihadapi dalam pengembangan dan implementasi aplikasi pemesanan tiket <i>travel</i> kurang dibahas.	Penggunaan metode <i>waterfall</i> , bersama dengan <i>database MySQL</i> , bahasa <i>PHP</i> , dan <i>framework CodeIgniter</i> , memastikan pendekatan yang terstruktur dan terorganisir dalam pengembangan dan implementasi aplikasi.	Studi ini mengembangkan aplikasi pemesanan tiket <i>travel</i> berbasis <i>web</i> menggunakan metode <i>waterfall</i> , dengan mengaplikasikan algoritma <i>Dijkstra</i> untuk merekomendasikan lokasi penjemputan penumpang. Fungsionalitas

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
						aplikasi diuji menggunakan pengujian <i>unit</i> dan integrasi, yang menghasilkan hasil yang <i>valid</i> .
2.	Implementasi Arsitektur <i>Microservice</i> Pada Aplikasi <i>Online Travel</i> Tourinc [8]	Penelitian ini membandingkan penggunaan arsitektur <i>microservice</i> dalam pengembangan aplikasi PT Tourinc untuk menjalankan proses bisnisnya.	Penelitian ini kontras antara metode pengembangan yang digunakan, yaitu <i>Web Service Implementation Methodology (WSIM)</i> , dan bahasa pemrograman yang digunakan,	Penelitian ini tidak menjelaskan secara rinci mengenai bagaimana aplikasi dapat mengotomatisasi proses konfirmasi pembayaran.	Penelitian ini menyimpulkan bahwa arsitektur <i>microservice</i> merupakan solusi yang tepat untuk mengatasi kompleksitas dalam pengembangan aplikasi PT Tourinc, terutama	Penelitian ini merekomendasikan penggunaan arsitektur <i>microservice</i> dalam pengembangan aplikasi tersebut. Metode pengembangan yang digunakan adalah <i>Web</i>

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
			yaitu <i>PHP</i> dengan <i>framework Laravel</i> dan <i>javascript</i> dengan <i>framework express.js</i> .		terkait akses ke <i>REST API</i> dari berbagai perusahaan.	<i>Service Implementation Methodology (WSIM)</i> , dengan menggunakan bahasa pemrograman <i>PHP</i> dan <i>framework Laravel</i> serta <i>javascript</i> dengan <i>framework express.js</i> .
3.	<i>Booking Tickets Online At PT CBB Using The Laravel Framework</i> [9]	Penelitian ini membandingkan penggunaan aplikasi pemesanan tiket	Penelitian ini menyoroti perbedaan antara proses pemesanan tiket <i>travel</i> yang	Penelitian ini kurangnya pembahasan mengenai pengalaman	Penelitian ini menyimpulkan bahwa pengembangan aplikasi	Penelitian dari pengembangan aplikasi ini adalah menciptakan aplikasi



No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
		<i>travel</i> berbasis <i>web</i> dengan proses pemesanan tiket <i>travel</i> yang dilakukan secara konvensional di PT Cahaya Berkah Bersama.	tidak efektif di PT Cahaya Berkah Bersama saat ini dengan harapan pengembangan aplikasi pemesanan tiket <i>travel</i> berbasis <i>web</i> yang lebih efisien dan praktis.	pelanggan dan respon pengguna terhadap aplikasi pemesanan tiket <i>travel</i> berbasis <i>web</i> yang dikembangkan.	pemesanan tiket <i>travel</i> berbasis <i>web</i> di PT Cahaya Berkah Bersama bertujuan untuk mengatasi masalah ketidakefektifan dalam pemesanan tiket <i>travel</i> dan meningkatkan aksesibilitas bagi pelanggan.	pemesanan tiket <i>travel</i> yang dapat dilakukan secara online, menghemat waktu, mudah diakses oleh pelanggan, dan memudahkan pemilik untuk melihat laporan transaksi pemesanan.
4.	Perancangan Sistem Informasi Reservasi Kamar Hotel Berbasis <i>Web</i> Dengan Metode <i>RUP</i>	Penelitian ini membandingkan sistem reservasi hotel secara online dengan sistem	Penelitian ini menyoroti perbedaan antara kemudahan dan efektivitas sistem reservasi hotel	Penelitian ini kurangnya pembahasan mengenai pengalaman pengguna,	Penelitian ini menyimpulkan bahwa perancangan sistem informasi reservasi hotel	Penelitian ini memungkinkan tamu untuk memeriksa ketersediaan kamar secara

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	<i>(Rational Unified Process)</i> [10]	reservasi hotel secara manual.	secara <i>online</i> dengan masalah yang muncul dalam sistem reservasi hotel secara manual.	keamanan data, dan tantangan dalam mengembangkan dan mengimplementasikan sistem informasi reservasi hotel berbasis <i>online</i> .	berbasis <i>online</i> bertujuan untuk mempermudah tamu dalam melakukan pemesanan secara <i>online</i> tanpa harus datang langsung ke hotel.	<i>online</i> dan memudahkan pihak hotel, terutama <i>administrator</i> , dalam mengelola data dan membuat laporan.
5.	Pengembangan Aplikasi Pengolahan Data Pemesanan Tiket Pesawat [3]	Penelitian ini membandingkan pengolahan data pemesanan tiket pesawat menggunakan sistem manual (pencatatan buku	Penelitian ini menyoroti perbedaan antara pengolahan data manual yang memerlukan pencatatan ulang dan berisiko	Penelitian ini kurang memberikan analisis yang mendalam terkait masalah yang dihadapi PT Elendra Tour and	Penelitian ini menggunakan pengujian sistem, metode <i>blackbox</i> digunakan untuk mengevaluasi kemampuan sistem dalam	Penelitian ini memperbaiki masalah dengan mengimplementasikan sistem informasi berbasis komputer menggunakan

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
		dan lembar kerja <i>Spreadsheet</i> ) dengan penggunaan sistem informasi berbasis komputer ( <i>JavaNetbeans</i> dan <i>MySQL</i> ).	kehilangan atau kerusakan data dengan pengolahan data menggunakan sistem informasi yang dapat meminimalisir kesalahan dan memudahkan penyimpanan data.	Travel dalam pengolahan data pemesanan tiket pesawat secara manual dan bagaimana sistem informasi yang diusulkan dapat secara efektif mengatasi masalah tersebut.	memudahkan pencatatan data pemesanan tiket pesawat dan meminimalkan risiko kehilangan atau kerusakan data.	<i>JavaNetbeans</i> dan <i>MySQL</i> sebagai <i>database</i> .
6.	Rancang Bangun Sistem Informasi <i>Travel</i> Berbasis <i>Web</i> Pada Jaya Abadi Travel [11]	Penelitian ini membandingkan penggunaan sistem informasi yang kurang efisien pada Jaya Abadi <i>Tour &amp; Travel</i>	Penelitian ini Kontras terhadap penggunaan sistem manual yang mengharuskan <i>customer</i> datang ke lokasi Jaya	Penelitian ini kurangnya pembahasan mengenai pengalaman pengguna, keamanan data,	Penelitian ini menyimpulkan bahwa pengembangan sistem informasi Jaya Abadi <i>Tour &amp; Travel</i> melalui	Diharapkan aplikasi <i>web</i> ini dapat meningkatkan efisiensi biaya, tenaga, dan waktu dalam pengelolaan

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
		dengan penggunaan <i>website</i> yang diusulkan untuk meningkatkan efisiensi dalam pengelolaan pemesanan <i>travel</i> .	Abadi <i>Tour &amp; Travel</i> untuk pemesanan dan pembayaran dengan penggunaan <i>website</i> yang memungkinkan <i>customer</i> melakukan pemesanan dan transaksi kapan saja dan dari mana saja.	dan kelebihan aplikasi <i>web</i> yang diusulkan dibandingkan dengan sistem yang ada.	pembuatan aplikasi <i>web</i> bertujuan untuk meningkatkan efisiensi dalam pengelolaan pemesanan <i>travel</i> .	pemesanan dan transaksi <i>travel</i> di Jaya Abadi <i>Tour &amp; Travel</i> .
7.	Rancang Bangun <i>Backend</i> Sistem Manajemen <i>Gymship</i> Berbasis	Penelitian ini memiliki kesamaan dengan penelitian lain	Penelitian ini berfokus pada pembangunan <i>backend</i> sistem	Penelitian ini tidak memberikan informasi yang cukup <i>detail</i>	Penelitian ini bertujuan untuk membantu masyarakat dalam	Penelitian ini bertujuan untuk mengembangkan <i>backend</i> sistem

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	<i>Website</i> Pada Alterra Academy [4]	yang bertujuan untuk mengembangkan sistem informasi untuk mendukung bisnis penyewaan gym.	manajemen <i>Gymship</i> berbasis <i>website</i> .	tentang bagaimana sistem informasi terkomputerisasi dapat mengurangi risiko kesalahan manusia.	mendapatkan jadwal latihan dengan mudah melalui pendaftaran <i>member</i> di <i>website Gymship</i> .	manajemen <i>Gymship</i> berbasis <i>website</i> serta mempermudah promosi produk jasa dan pendaftaran <i>member</i> , serta memudahkan pemilihan instruktur dan jadwal latihan.
8.	Pemanfaatan <i>UML (Unified Modeling Language)</i> Dalam Perencanaan Sistem Penyewaan Baju Adat	Penelitian ini membandingkan perkembangan teknologi yang pesat dengan kebutuhan	Penelitian ini menyoroti kemajuan teknologi yang pesat dengan kesulitan	Penelitian ini kurang memberikan analisis yang mendalam tentang tantangan dan	Hasil penelitian ini mencakup pengembangan perangkat lunak untuk penyewaan baju adat dan	Hasil penelitian ini mengimplementasikan perangkat lunak menggunakan SDLC dan UML

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	Berbasis Website [12]	perusahaan dalam memanfaatkan teknologi untuk memfasilitasi pekerjaan mereka.	konsumen dalam mengecek ketersediaan baju adat secara langsung di tempat.	hambatan yang dihadapi dalam pengembangan perangkat lunak untuk penyewaan baju adat.	rekomendasi diagram <i>UML</i> yang dapat digunakan oleh pengembang perangkat lunak terkait.	dengan bahasa pemrograman Java dan MySQL. Hasil penelitian mencakup pengembangan perangkat lunak untuk penyewaan baju adat dan rekomendasi diagram <i>UML</i> yang berguna bagi pengembang perangkat lunak.
9.	Pengujian <i>Blackbox</i> Dan <i>Whitebox</i> Sistem	Penelitian ini membandingkan sistem parkir yang belum tertata	Penelitian ini terdapat kontras antara sistem parkir yang belum	Penelitian ini tidak memberikan informasi yang cukup mengenai	Penelitian ini dilakukan untuk mengatasi masalah sistem parkir yang	Penelitian ini pada pengujian sistem menggunakan <i>blackbox testing</i>

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	Informasi Parkir Berbasis Web [13]	dengan sistem parkir yang diimplementasikan melalui perancangan dan pembuatan sistem berbasis <i>web</i> .	terkontrol dengan baik sebelumnya dengan sistem parkir yang diharapkan dapat mengatur kendaraan mahasiswa, dosen, dan <i>staff</i> secara terkontrol.	metode yang digunakan dalam perancangan dan pembuatan sistem parkir berbasis <i>web</i> .	belum tertata dan kurang terkontrol di Universitas Islam Negeri Sultan Syarif Kasim Riau melalui perancangan dan pembuatan sistem parkir berbasis <i>web</i> .	dan <i>whitebox testing</i> menunjukkan bahwa sistem yang dibangun berhasil dan <i>valid</i> sesuai dengan kebutuhan dan fungsi. <i>User</i> sistem diberikan hak akses berdasarkan perancangan sistem yang telah dilakukan.

## 2.2 Landasan Teori

Berikut ini merupakan dasar teori yang berkaitan atau berhubungan menjadi landasan penelitian ini yang bersumber dari buku dan jurnal terdahulu.

### 2.2.1 *Travel*

*Travel* merujuk pada aktivitas perpindahan individu antara lokasi-lokasi yang jauh secara geografis. Aktivitas ini dapat dilakukan dengan berjalan kaki, menggunakan sepeda, mobil, kereta, perahu, bus, pesawat, kapal, atau sarana transportasi lainnya. Perusahaan *travel* beroperasi di industri jasa transportasi, akomodasi, dan layanan terkait lainnya. Bisnis *travel* menuntut pelayanan yang baik dan responsif terhadap konsumen, sehingga memerlukan pemanfaatan teknologi informasi yang efektif. [1].

### 2.2.2 Pemesanan Tiket

Pemesanan tiket merupakan suatu sistem yang melibatkan kesepakatan antara konsumen dan penyedia layanan untuk memesan jasa transportasi dalam periode waktu tertentu. Fungsi-fungsi dari sistem pemesanan tiket *travel* meliputi memberikan informasi terkini tentang jadwal keberangkatan dan menyediakan layanan pemesanan tiket kepada konsumen [1].

### 2.2.3 Alterra

Alterra Academy (PT Marka Kreasi Persada) Anak perusahaan yang melayani kebutuhan bisnis spesialisasi dalam pengembangan, distribusi, dan pemeliharaan perangkat lunak. Alterra Academy sebelumnya dikenal sebagai Alpha Tech Academy, adalah sebuah inkubator talenta teknologi yang menyediakan kesempatan bagi individu dari berbagai latar belakang, termasuk non-IT, untuk menjadi tenaga kerja profesional di bidang teknologi.

### 2.2.4 Golang

Bahasa pemrograman Golang, atau disebut juga *Google Language*, dibuat oleh Google pada tahun 2009 bersama dengan Ken



Thompson, Robert Griesemer, dan Rob Pike. Bahasa ini dibuat dengan tujuan untuk menjadi bahasa pemrograman yang cepat, handal, *scalable*, dan mudah digunakan. Golang memadukan keamanan dan efisiensi dalam pengembangan sistem *open source* dan juga mendukung konkurensi dengan implementasi yang sederhana. Golang memiliki sistem *garbage collection* yang baik dengan bantuan *built-in garbage collector process (Go-routines)* sehingga dapat menangani sistem dengan skala besar. Selain itu, Golang merupakan bahasa pemrograman yang andal dan cepat dalam skala besar, serta di desain dengan *clean code* agar tidak membebani sistem [5].

#### 2.2.5 Framework Echo

*Echo* adalah sebuah *web framework* pada bahasa pemrograman Golang yang memiliki performa yang tinggi, *extensibility* yang baik, dan desain minimalis yang sangat cocok digunakan pada sistem dengan skala besar. *Framework echo* juga memiliki beberapa keunggulan lainnya seperti *optimized router*, dukungan untuk *HTTP/2*, kemampuan untuk melakukan *scaling*, *data binding* dan *data rendering*, *middleware*, *automatic TLS*, dan kemampuan untuk melakukan *error handling* yang dapat di-*customize*. Oleh karena itu, *framework echo* sangat berguna dalam pembuatan *web service* [14].

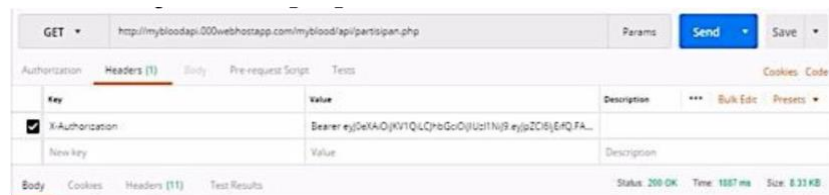
#### 2.2.6 JSON Web Token (JWT)

*JSON Web Token (JWT)* adalah sebuah token berupa string yang terdiri dari tiga bagian yaitu *header*, *payload*, dan *signature*, yang digunakan untuk proses otentikasi dan pertukaran informasi. Terdapat dua jenis *token*, yaitu *token* pembawa dan *token* pemegang kunci. Selain itu, terdapat juga dua skema berdasarkan tujuannya, yaitu *token* identitas dan *token* akses. *JWT* bekerja dengan cara yang mirip dengan *password*. Setelah pengguna berhasil *login*, *server* akan memberikan *token* yang disimpan di *local storage* atau *cookies browser*. *Token* tersebut digunakan untuk mengakses halaman tertentu, dan pengguna

akan mengirimkan *token* tersebut sebagai bukti bahwa ia sudah berhasil *login* [6].

### 1. Pengujian *Response HTTP Server*

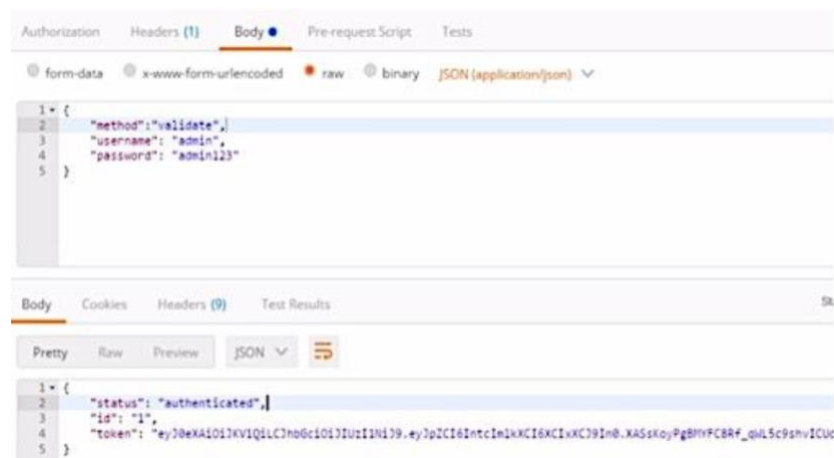
Sebelum dilakukan pengujian, beberapa tahapan perlu disiapkan terlebih dahulu seperti membuat atau menambahkan *data dummy* yang terdiri dari pengguna, *operator*, *admin*, *event* kegiatan, dan lain-lain. Setelah itu, uji coba dilakukan dengan melakukan login dari berbagai peran seperti *admin*, *operator*, dan pengguna. Selanjutnya, selama proses *login*, *token JWT* yang dihasilkan serta sumber daya yang diperoleh akan diamati untuk memastikan bahwa hak akses yang dimiliki oleh peran tersebut sesuai dengan apa yang diperoleh. Proses *parsing data JWT* menggunakan beberapa alat, salah satunya adalah aplikasi Postman. Contoh isi header untuk menguji proses *login* dengan hak akses sebagai *admin* dapat dilihat pada Gambar 2.1[6].



Gambar 2.1 Pengujian Response HTTP Server

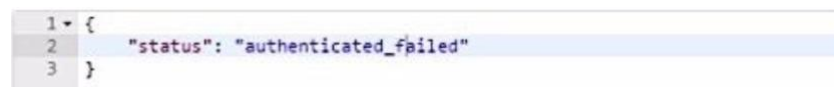
### 2. Pengujian *Authentication* dengan *JWT*

Konten *login* yang dikirim ke *server* berbentuk *JSON* yang dikemas dalam tipe konten *HTTP body*. *JSON* tersebut terdiri dari *username* dan *password*. Setelah itu, permintaan tersebut dikirim ke layanan *web*, dan *server* akan memberikan *response* seperti yang terlihat pada Gambar 2.2[6].



Gambar 2.2 Pengujian Authentication dengan JWT

Respon yang diterima berbentuk *JSON* dan terdiri dari *token* dan *data HTTP*. Nilai *token* akan berbeda-beda karena *token* tersebut memiliki *data* pengguna dan hak akses yang berbeda. Pengujian selanjutnya dapat dilihat pada Gambar 2.3 dengan memasukkan kredensial *login* yang salah [6].



Gambar 2.3 Response Login Gagal

Percobaan berikutnya menguji sumber daya yang dimiliki oleh *token*. Hasilnya, layanan *web* akan memberikan respons dari *server* jika konten tersebut tersedia [6], seperti yang terlihat pada Gambar 2.4.



yang menggunakan protokol *HTTP* [6]. Arsitektur *REST* terdiri dari beberapa komponen sebagai berikut:

1. *URI design*

*REST API* menggunakan protokol *HTTP* untuk mengakses sumber daya. Untuk mengakses sumber daya pada *API*, diperlukan *Uniform Resource Identifier (URI)* atau yang biasa disebut *endpoint* pada *API*. Contohnya adalah pemanggilan *endpoint* seperti “*users*”, “*users/125*”, “*users/125/orders*”, dan lain sebagainya [6].

2. *HTTP verbs*

Komponen ini dikenal juga sebagai metode yang digunakan saat *client* melakukan permintaan pada *endpoint* ke *server*. Ada berbagai metode yang tersedia pada *HTTP*, namun beberapa metode yang sering digunakan adalah *GET*, *POST*, *PUT*, dan *DELETE* [6].

3. *HTTP response code*

Komponen ini merupakan kode standar pada protokol *HTTP* yang digunakan saat *client* melakukan permintaan [6]. Terdapat tiga jenis kode yang biasa digunakan pada *REST API*, yaitu:

- a. *2XX*, menandakan bahwa permintaan *client* berhasil.
- b. *4XX*, menandakan terdapat kesalahan pada *client* saat melakukan permintaan.
- c. *5XX*, menandakan terdapat kesalahan pada *server* saat melakukan permintaan.

4. *Format response*

Setiap permintaan yang dilakukan oleh *client*, *server* akan memberikan balasan berupa data atau informasi. Terdapat dua *format* respon yang umum digunakan pada *REST API*, yaitu *Extensible Markup Language (XML)* atau *JavaScript Object Notation (JSON)* [6].

### 2.2.8 *Swagger*

*Swagger* merupakan *software* yang berfungsi untuk membuat dokumentasi *REST API*. Dengan *Swagger*, dokumen *API* dapat dibagikan antara manajer proyek, pengujian dan pengembang. Selain itu, *Swagger* juga dapat diintegrasikan dengan berbagai tools untuk mengotomatisasi proses terkait *API*. *Swagger* bekerja secara *real-time*, sehingga klien dan sistem dokumentasi bergerak dengan kecepatan yang sama dengan *server*. Deskripsi dari *method*, *parameter*, dan *model* terintegrasi secara erat ke dalam kode *server*, sehingga sinkronisasi antara *API* dan dokumentasinya dapat terjaga dengan baik [4].

### 2.2.9 *Microservices Architecture*

*Microservices architecture* adalah suatu pendekatan dalam mengembangkan perangkat lunak dengan memecah aplikasi besar menjadi beberapa layanan kecil yang berdiri sendiri, masing-masing menjalankan fungsi bisnis yang spesifik. Setiap layanan dapat dikembangkan secara independen dan dapat di-*deploy* secara terpisah. Komunikasi antara layanan biasanya dilakukan melalui protokol jaringan seperti *HTTP* atau *message queue*. Pendekatan ini bertujuan untuk mempermudah pengembangan dan *deployment* perangkat lunak, memungkinkan skalabilitas yang lebih baik, dan memperbaiki kehandalan aplikasi dengan mengurangi *single point of failure*. Karena masing-masing layanan memiliki *codebase* dan *database* yang terpisah, sehingga memungkinkan untuk menggunakan teknologi yang berbeda-beda untuk setiap layanan [15].

### 2.2.10 *Software Development Life Cycle (SDLC)*

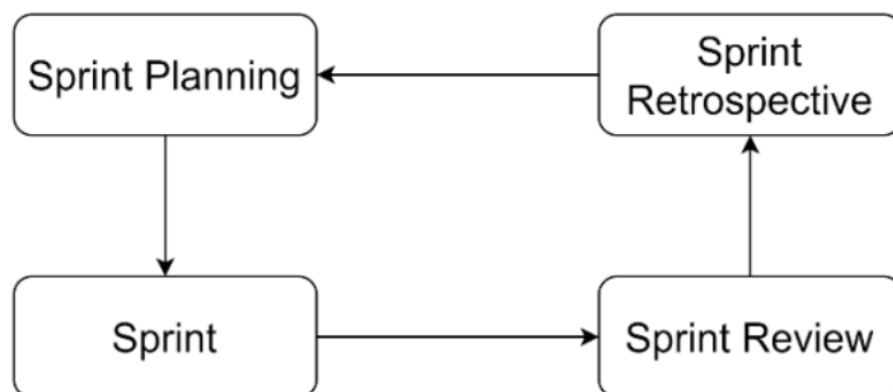
*SDLC* adalah suatu proses merancang sistem yang selalu bergerak seperti roda yang melalui langkah-langkah seperti perencanaan, analisis, desain, implementasi, dan pemeliharaan. Kemudian sistem akan kembali ke tahap perencanaan jika dianggap tidak efisien untuk diimplementasikan [16].

### 2.2.11 Agile

*Agile* adalah teknik kolaboratif antara pendekatan iteratif dan evolusioner dengan menggunakan dokumen formal terkait dalam membangun perangkat lunak yang berkualitas baik dalam hal biaya dan tepat waktu sesuai dengan kebutuhan yang berubah [16].

### 2.2.12 Scrum

*Scrum* adalah sebuah *framework* untuk pengembangan yang termasuk dalam kategori “*Agile Methodologies*”. *Scrum* bersifat fleksibel dan adaptif, dimana dapat diterapkan pada proyek dengan berbagai tingkat kompleksitas. Secara singkat, *scrum* adalah sebuah *framework* yang membantu tim atau organisasi mencapai tujuan pada proyek yang kompleks dengan menggunakan pendekatan iteratif yang menghasilkan solusi yang adaptif [16].



Gambar 2.6 Scrum Events

Tujuan utama dari aktivitas dalam *scrum* adalah untuk memeriksa dan menyesuaikan artefak yang dibuat. Terdapat lima aktivitas yang idealnya terjadi dalam satu iterasi *scrum*, yaitu *sprint*, perencanaan *sprint*, *daily scrum*, tinjauan *sprint*, dan retrospektif *sprint* [16].

#### 1. The Sprint

Inti dari metode *scrum* adalah *sprint*, yang merupakan proses untuk mengubah ide menjadi solusi yang bernilai. Selain itu, *sprint* juga menjadi tempat untuk menjalankan kegiatan *scrum* lain seperti *sprint planning*, *daily scrum*, *sprint review*, dan

*sprint retrospective*. Durasi *sprint* selalu sama, yakni antara dua hingga empat minggu. Namun, selama *sprint* berlangsung ada beberapa hal yang harus diperhatikan seperti tetap mempertahankan *sprint goal*, menjaga kualitas, menyesuaikan *product backlog*, serta mengklarifikasi cakupan *sprint* jika ada temuan baru yang muncul selama *sprint* berlangsung [16].

## 2. *Sprint Planning*

Sebelum dimulai *sprint*, dilakukan kegiatan perencanaan yang disebut *sprint planning* sebagai langkah awal. *Scrum guide* menjelaskan bahwa saat *sprint planning* dilaksanakan, beberapa topik akan dibahas, seperti menetapkan *sprint goal* yang memiliki nilai tambah bagi produk untuk meningkatkan nilai keseluruhan produk, dan memilih item-item yang terdapat pada *product backlog* yang akan dikerjakan pada *sprint* dan dirangkum dalam *sprint backlog* [16].

## 3. *Daily Scrum*

Menurut *scrum guide*, kegiatan *daily scrum* bertujuan untuk memeriksa proses pencapaian *sprint goal* dan menyesuaikan *sprint backlog* sesuai kebutuhan. Kegiatan ini dilakukan setiap hari pada waktu yang sama dengan durasi sekitar 15 menit. Tim pengembang bebas memilih teknik atau cara apa pun untuk mencapai *sprint goal*. Selama kegiatan ini, anggota tim akan merencanakan tugas untuk hari berikutnya, sehingga dapat meningkatkan manajemen diri tim secara langsung [16].

## 4. *Sprint Review*

Kegiatan ini bertujuan untuk memeriksa hasil dari *sprint* dan menentukan perubahan yang diperlukan di masa depan. Tim *scrum* akan mempresentasikan hasil kerja *sprint* dan progres menuju *product goal* yang telah dicapai. *Stakeholder* di luar tim akan mengetahui apa yang telah dicapai selama *sprint* dan dampaknya pada lingkungan mereka. secara *ideal*, waktu yang



diperlukan untuk kegiatan ini adalah empat jam dalam satu sprint bulanan, tetapi jika *sprint* lebih pendek, waktu kegiatan ini juga akan lebih singkat [16].

#### 5. *Sprint Retrospective*

Kegiatan ini bertujuan untuk meningkatkan efektivitas dan kualitas *sprint* selanjutnya dengan melakukan inspeksi pada beberapa aspek kegiatan *sprint* sebelumnya. Tim akan mengevaluasi temuan yang berguna, permasalahan yang dihadapi, dan solusi yang telah dilakukan. Kemudian, tim akan mengidentifikasi hal-hal yang paling penting untuk meningkatkan efektivitas *sprint* dan menerapkannya pada *sprint* selanjutnya. Durasi ideal kegiatan ini adalah tiga jam dalam satu bulan *sprint*, namun waktu yang diperlukan dapat disesuaikan dengan durasi *sprint* yang lebih pendek [16].

### 2.2.13 Artefak Scrum

Dalam metode *scrum*, artefak digunakan untuk mewakili hasil kerja tim atau organisasi selama proses pengembangan proyek. Artefak dirancang untuk menjadi transparan sehingga semua anggota tim memahami informasi yang sama dan dapat melakukan adaptasi sesuai kebutuhan. Terdapat tiga jenis artefak dalam *scrum* yaitu *product backlog*, *sprint backlog*, dan *increment*, yang masing-masing memiliki peran dan fungsi yang berbeda dalam pengembangan produk [16].

#### 1. *Product Backlog*

Artefak ini adalah hal yang paling penting untuk mengarahkan pekerjaan tim *scrum*. Artefak ini memiliki sebuah istilah yaitu komitmen yang harus dipahami oleh seluruh anggota tim *scrum*, yaitu *product goal*. *Product goal* menjelaskan tentang kondisi produk yang menjadi target untuk tim *scrum* dalam membuat rencana kerja. *Product goal* adalah tujuan jangka panjang tim *scrum*, sehingga setiap tujuan harus dicapai atau dihapus agar tim dapat mencapai tujuan lainnya [16].

## 2. *Sprint Backlog*

Artefak ini mencakup *sprint goal*, setiap *item product backlog* yang dipilih untuk dikerjakan dalam *sprint*, dan rencana aksi untuk memberikan *delivery* terhadap *increment*. Artefak ini dibuat dan digunakan oleh *developers*, dengan visualisasi yang sangat detail dan mencerminkan rencana teknis yang menjadi acuan *developer* dalam mengerjakan tugas. Artefak ini dapat disesuaikan selama *sprint* berlangsung. *Sprint backlog* memiliki komitmen yang disebut *sprint goal*, yang diterapkan oleh *developer*. *Sprint goal* dibuat selama kegiatan *sprint planning* dan ditambahkan ke *sprint backlog*. Jika hasil pengerjaan tidak memenuhi ekspektasi, cakupan *sprint backlog* dapat dinegosiasikan dengan *product owner* tanpa mempengaruhi *sprint goal* [16].

## 3. *Increment*

Artefak ini berfungsi sebagai pijakan untuk mencapai *product goal*. Setiap *increment* yang ditambahkan merupakan akumulasi dari keseluruhan *increment* sebelumnya. Pada akhir *sprint*, *increment* dibuat sebagai dokumentasi hasil kerja tim yang telah diselesaikan sesuai dengan *definition of done*. Fokus pada *definition of done* adalah untuk memastikan bahwa *increment* sesuai dengan ekspektasi produk. *Increment* akan tercipta ketika setiap butir *product backlog* telah memenuhi *definition of done*. *Definition of done* memberikan transparansi dan pemahaman kepada setiap orang tentang pekerjaan yang telah selesai dan merupakan bagian dari *increment*[16].





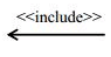
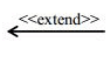
Penerapan *Scrum* tidak membutuhkan penerapan semua tuntunan di atas, namun jika hanya diterapkan secara parsial, itu tidak akan menghasilkan *Scrum*. *Scrum* adalah kerangka kerja yang dapat menampung teknik, metode, atau praktik lain di dalamnya [16].

### 2.2.14 *Unified Modeling Language (UML)*

Bahasa pemodelan *Unified Modeling Language (UML)* digunakan dalam pengembangan sistem perangkat lunak yang berbasis objek. *UML* digunakan untuk memodelkan dan mengkomunikasikan informasi mengenai sistem menggunakan skema dan skrip pendukung. Tujuan dari *UML* adalah menyediakan bahasa pemodelan yang independen dari berbagai bahasa pemrograman dan proses rekayasa. Selain itu, penggunaan *UML* mengikuti praktik terbaik yang telah distandarisasi dalam pemodelan dan menyediakan model yang siap digunakan. [12].

#### 1. *Use Case Diagram*







*Use Case Diagram* digunakan untuk menggambarkan bagaimana aktor berinteraksi dengan sistem dalam hal fungsionalitas yang ada. Tujuan dari penggunaan *use case* adalah untuk mencatat daftar *aktor* dan *use case* yang ada, serta mengidentifikasi aktor yang terlibat dalam setiap *use case* secara spesifik [12]. Pada Gambar 2.7 merupakan simbol dari *use case diagram*.

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 2.7 Simbol Use Case Diagram

## 2. Activity Diagram

*Activity Diagram* digunakan untuk menggambarkan rangkaian langkah-langkah prosedur dari suatu sistem dan menggambarannya berdasarkan aktivitas yang dilakukan oleh setiap aktor yang terlibat dalam sistem tersebut [12]. Pada Gambar 2.8 merupakan simbol dari *activity diagram*.

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2.8 Simbol Activity Diagram

### 2.2.15 BlackBox Testing

Pengujian *blackbox testing* juga dikenal sebagai pengujian perilaku, melibatkan pengujian perangkat lunak tanpa pengetahuan tentang struktur internal atau logika yang diuji oleh penguji. Pada pengujian ini, penguji mengacu pada spesifikasi kebutuhan dan tidak perlu menganalisis kode secara mendetail. Pendekatan pengujian *blackbox* dilakukan dengan mempertimbangkan perspektif pengguna akhir. Pengujian *program* dilakukan setelah penulisan kode *program* untuk memeriksa dan memastikan bahwa komponen-komponen berfungsi sesuai harapan. Tujuan dari pengujian ini adalah untuk menemukan kesalahan atau kelemahan yang mungkin terdapat dalam sistem [13].

### 2.2.16 *Load Testing*

Pengujian *load testing* adalah jenis pengujian kinerja yang digunakan untuk menentukan bagaimana sistem berperilaku di bawah beban kerja yang berat. Hal ini biasanya melibatkan simulasi penggunaan bersamaan oleh banyak pengguna untuk memeriksa bagaimana aplikasi berperilaku dalam kondisi normal dan pada puncaknya. *Load testing* penting untuk memverifikasi bahwa sistem yang dikembangkan atau dikelola dapat menangani beban yang diharapkan dalam produksi. Proses *load testing* melibatkan: [17]

1. Perencanaan

Mendefinisikan tujuan pengujian, menentukan metrik sukses, dan merencanakan skenario pengujian.

2. Pembuatan *Script*

Membuat *script* yang mensimulasikan aktivitas pengguna dalam aplikasi.

3. Pengaturan *Environment* Pengujian

Menyiapkan lingkungan yang menyerupai produksi sebisa mungkin.

4. Pelaksanaan

Menjalankan *script* di lingkungan pengujian dan memonitor metrik kinerja.

5. Analisis

Menganalisis hasil pengujian untuk mengidentifikasi *bottleneck* dan masalah kinerja lainnya.

6. Optimisasi

Membuat perubahan yang diperlukan pada konfigurasi sistem, kode aplikasi, atau infrastruktur.