

BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian Kuantitatif

Metode penelitian kuantitatif merupakan metode yang dilakukan secara sistematis dan menggunakan model yang bersifat matematis. Metode penelitian ini bersifat objektif dan analitis. Untuk memperoleh data-data umumnya dilakukan tes, pengujian dan wawancara terstruktur. Data penelitian yang didapat berupa angka, skala, maupun sebuah grafik yang dapat dihitung. Oleh karena itu penulis menggunakan metode ini, karena data yang akan didapat dari hasil pengujian pada performa *framework* Laravel dan Codeigniter, merupakan data angka yang dapat dihitung.

3.2 Metode Pengumpulan Data

Metode pengumpulan data adalah cara yang digunakan oleh peneliti untuk memperoleh data yang diperlukan. Pada umumnya selalu ada hubungan antara metode pengumpulan data dengan masalah yang ingin dipecahkan. Karena masalah memberikan sebuah arah yang dapat mempengaruhi metode pengumpulan data. Pada penelitian ini penulis menggunakan metode observasi dan studi literatur untuk mendapatkan data yang diperlukan.

3.7.1 Observasi

Observasi merupakan metode pengumpulan data yang dilakukan melalui pengamatan langsung. Pada metode ini peneliti menggunakan pancaindranya sendiri untuk mengamati objek penelitian, yang berarti peneliti dianggap sebagai pengamat. Dalam mengumpulkan data menggunakan metode ini, peneliti dapat menggunakan catatan, rekaman atau media lainnya yang dapat membantu.

3.7.2 Studi Literatur

Studi literatur merupakan metode pengumpulan data dengan mempeleajari dan melakukan peninjauan dari berbagai sumber seperti buku, jurnal, tesis, internet, dan sumber lainnya yang memiliki relevansi dengan penelitian yang sedang dilakukan, sehingga dapat menjadi dasar teori dan bahan perbandingan.

3.3 Metode Analisis Statistik Deskriptif

Pada umumnya data yang kita dapatkan dari proses pengumpulan data (data mentah) belum memiliki makna atau arti apapun. Oleh karena itu diperlukan proses analisis data agar data tersebut menjadi informasi yang berguna. Pada penelitian ini penulis menggunakan metode analisis statistik deskriptif yang merupakan metode analisis untuk menggambarkan karakteristik dasar, informasi, dan ringkasan data dengan cara rasional. Metode ini mendeskripsikan data secara kuantitatif. Statistik deskriptif dapat mencakup angka, bagan, tabel, grafik, dan jenis visualisasi data lainnya yang dapat digunakan untuk menyajikan data mentah.

3.4 Alat Dan Bahan

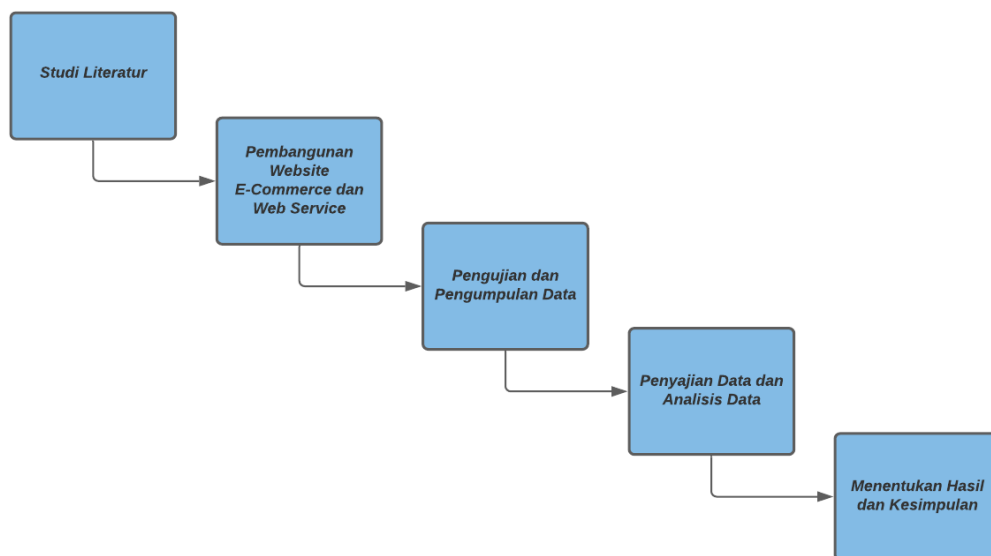
Dalam pelaksanaan penelitian ini dibutuhkan alat dan bahan dalam bentuk perangkat keras (*hardware*) maupun perangkat lunak (*software*), sebagai berikut :

1. Perangkat Keras (*hardware*)
Berikut ini merupakan spesifikasi perangkat keras yang digunakan penulis dalam melaksanakan penelitian ini :
 - a) Laptop dengan *processor* AMD ryzen 5 3500U
 - b) Memory RAM 10 GB DDR4
 - c) SSD 500 GB

2. Perangkat Lunak (*software*)
Berikut ini merupakan perangkat lunak yang digunakan penulis dalam melaksanakan penelitian ini :
 - a) Windows 10 *Home*
 - b) Microsoft *Office* 2019
 - c) XAMPP
 - d) Visual Studio Code
 - e) Google Chrome
 - f) Site24x7
 - g) Apache *Benchmark*

3.5 Prosedur Penelitian

Prosedur penelitian merupakan hal yang sangat penting dalam suatu proses penelitian, agar penelitian tersebut dapat berjalan sesuai dengan alur yang telah ditetapkan. Dan dengan adanya prosedur penelitian ini diharapkan dapat memudahkan pembaca untuk memahami proses-proses penelitian yang dilakukan. Berikut ini adalah gambar dari prosedur penelitian :



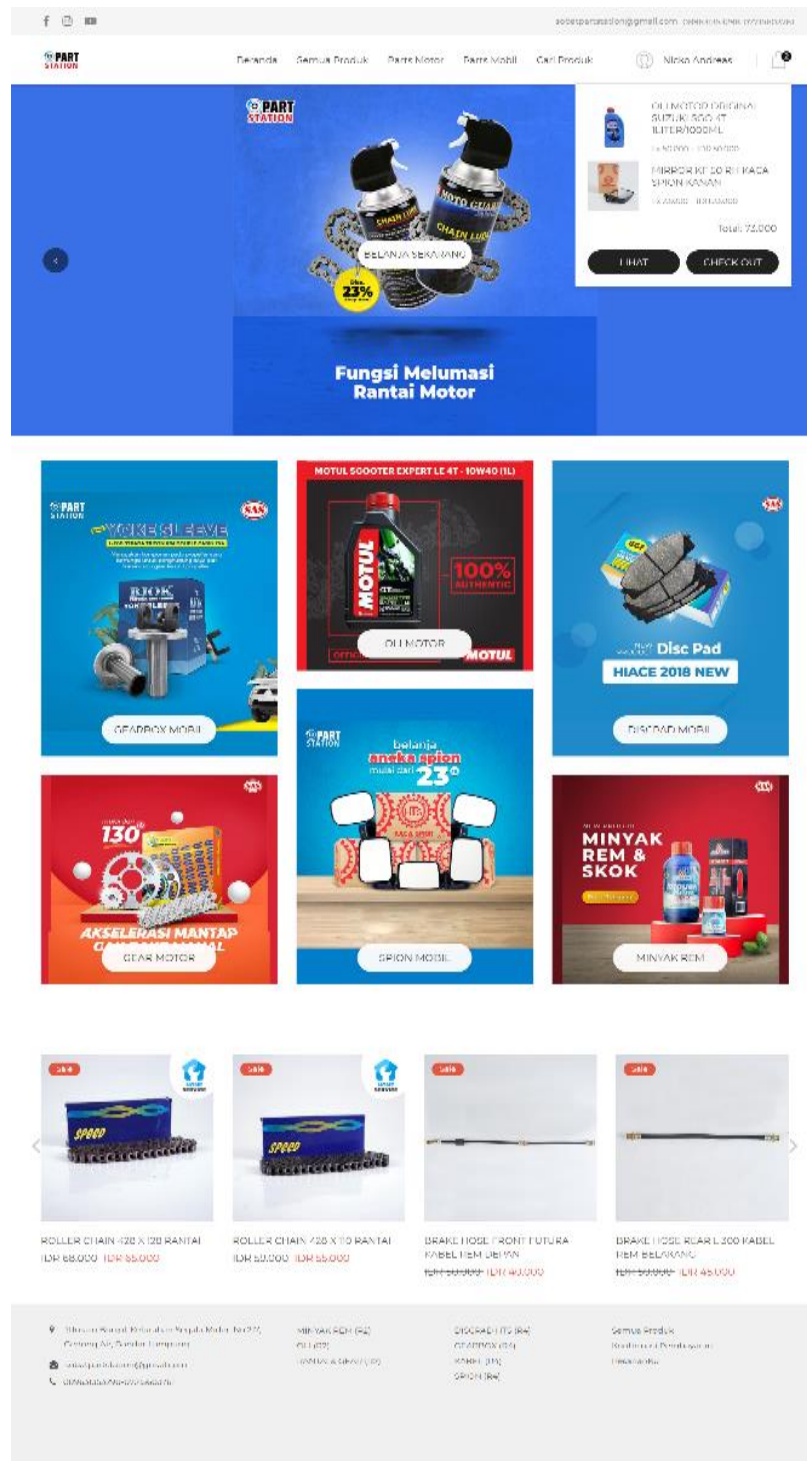
Gambar 3.1 Prosedur Penelitian.

Dari gambar 3.1 dapat dilihat bahwa tahap pertama pada penelitian ini adalah melakukan studi literatur. Studi literatur dilakukan pada jurnal, artikel, tesis, dan buku yang memiliki relevansi dengan penelitian ini. Selanjutnya pada tahap kedua yaitu melakukan pembangunan *website e-commerce* menggunakan *framework* Laravel dan Codeigniter. Pembangunan yang dilakukan hanya pada fungsi-fungsi dasar *website e-commerce*, mengingat pembangunan ini diperlukan untuk pengujian performa. Dan pada tahap ini juga penulis membangun sebuah *web service* yang mengolah data *user* untuk digunakan pada pengujian waktu eksekusi transaksi data menggunakan arsitektur REST. Selanjutnya pada tahap ketiga dilakukan pengujian performa pada *website e-commerce* yang telah dibangun. Pada tahap ini juga penulis sebagai peneliti melakukan observasi pada pengujian dan melakukan pengumpulan data dari hasil pengujian tersebut.

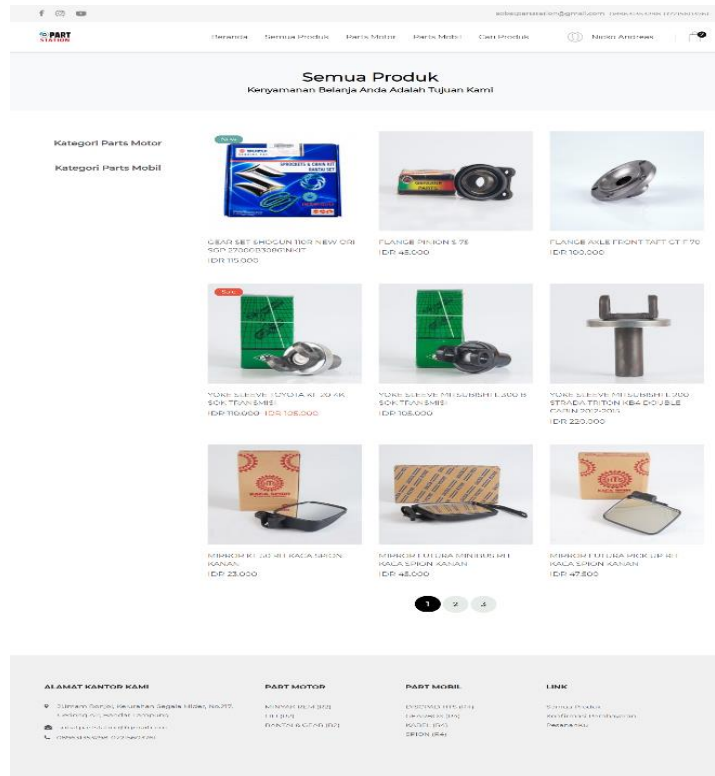
Selanjutnya tahap keempat, dimana data mentah yang telah dikumpulkan dari hasil pengujian performa akan dilakukan visualisasi data, sehingga data-data tersebut menjadi informasi yang berguna. Setelah mendapatkan informasi, penulis akan melakukan analisis performa pada *framework* Laravel dan Codeigniter. Selanjutnya tahap terakhir, dimana pada tahap ini penulis menuntukan hasil analisis dan perbandingan, serta kesimpulan yang didapatkan pada penelitian ini.

3.6 Objek Penelitian (*Website E-Commerce Laravel dan Codeigniter*)

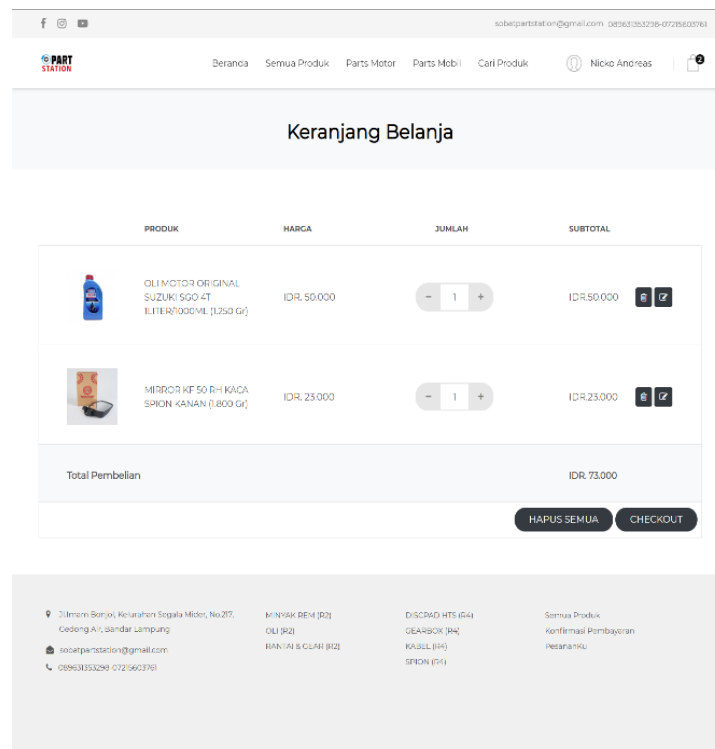
Pada subbab ini penulis akan menampilkan hasil dari pembangunan *website e-commerce* yang menggunakan *framework* Laravel dan Codeigniter. Namun karena antarmuka kedua *website e-commerce* ini identik, penulis hanya akan menampilkan satu *website e-commerce* untuk mewakili keduanya. Kedua *website e-commerce* ini juga telah di *hosting* sehingga dapat diakses secara *online*, untuk alamat *website e-commerce* yang dibangun dengan *framework* Laravel adalah www.laravel.sasparts-ci.my.id, sedangkan untuk alamat *website e-commerce* yang dibangun menggunakan *framework* Codeigniter adalah www.codeigniter.sasparts-ci.my.id. Berikut ini adalah tampilan antarmukanya.



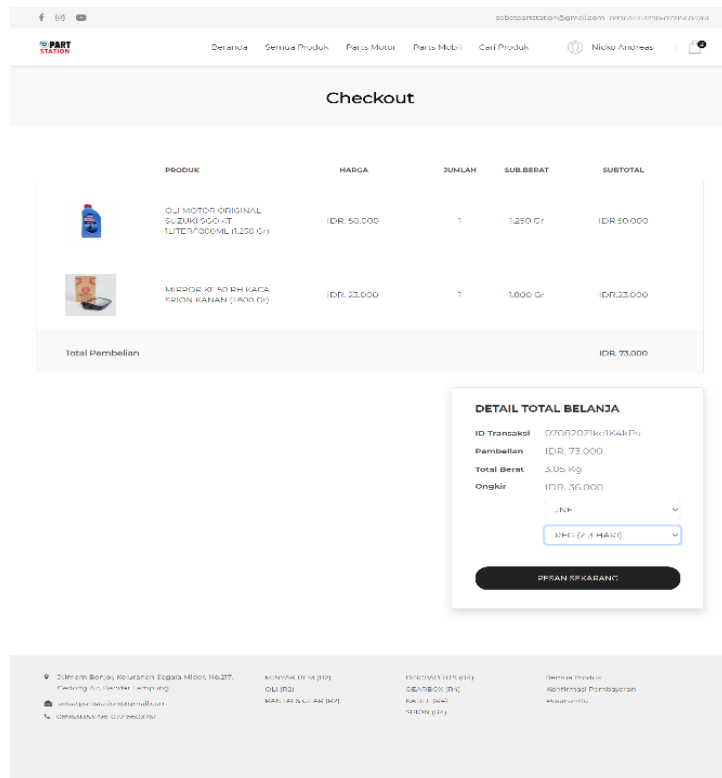
Gambar 3.2 Halaman Home.



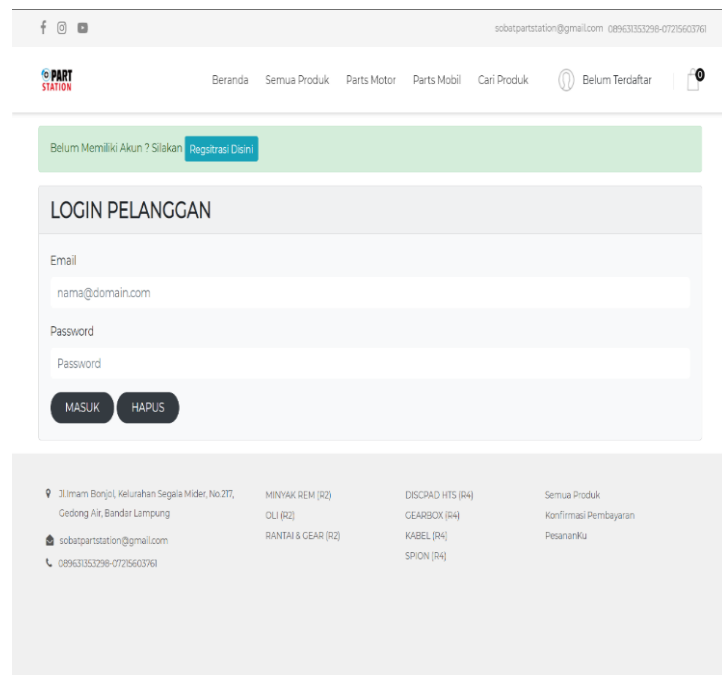
Gambar 3.3 Halaman Produk.



Gambar 3.4 Halaman Keranjang Belanja.



Gambar 3.5 Halaman Checkout.



Gambar 3.6 Halaman Login.

f @

sobatpartstation@gmail.com 08963153298-0725603761

PART STATION Beranda Semua Produk Parts Motor Parts Mobil Cari Produk

Sudah Memiliki Akun? Silakan [Login Disini](#)

REGISTRASI PELANGGAN

Nama
Nama Pengguna

Email
nama@domain.com

Password
Password

No Telepon
08xxxxxxxxxx

Provinsi
--Pilih Provinsi--

Kabupaten/Kota
--Pilih Kota--

Alamat Lengkap

Jl. Imam Bonjol, Kelurahan Segala Mider, No.27,
 Gedung Air, Bandar Lampung
 sobatpartstation@gmail.com
 08963153298-0725603761

MINYAK REM (R2)
 DLI (R2)
 HANJAI & CLEAR (R2)

DISCRAD HTS (R4)
 CLEARBOX (R4)
 KABEL (R4)
 SPONG (R4)

Semua Produk
 Konfirmasi Pembayaran
 Hesamanku

Gambar 3.7 Halaman Registrasi.

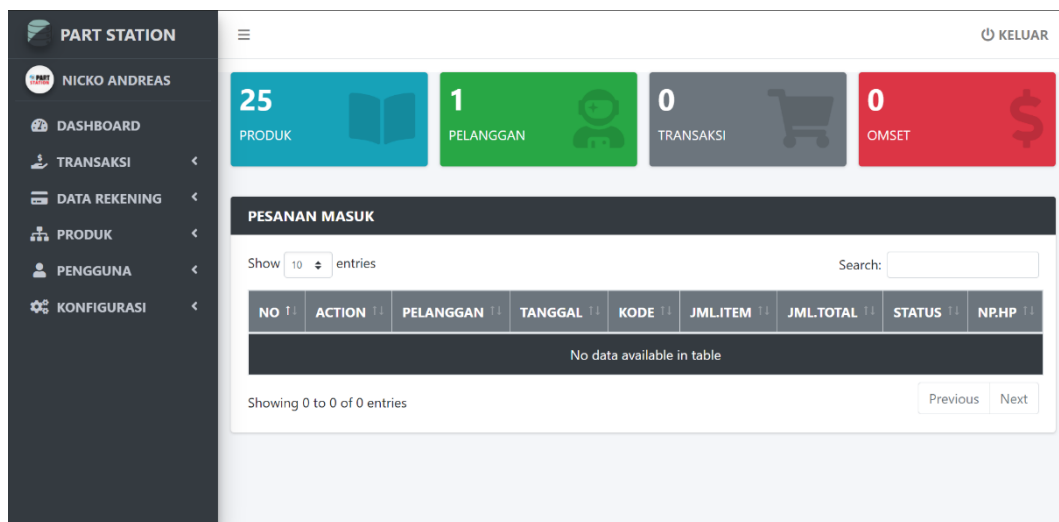
PART STATION

LOGIN ADMINISTRATOR

Username

Password

Gambar 3.8 Halaman Login Admin.



Gambar 3.9 Halaman *Dashboard Admin*.

3.7 Metode Pengujian

Subbab ini akan menjelaskan tentang metode pengujian performa pada *website e-commerce* yang dibangun menggunakan *framework* Laravel dan Codeigniter. Parameter-parameter dan alat yang digunakan dalam pengujian, dapat dilihat pada tabel dibawah ini.

Tabel 3.1 Parameter dan Alat Pengujian.

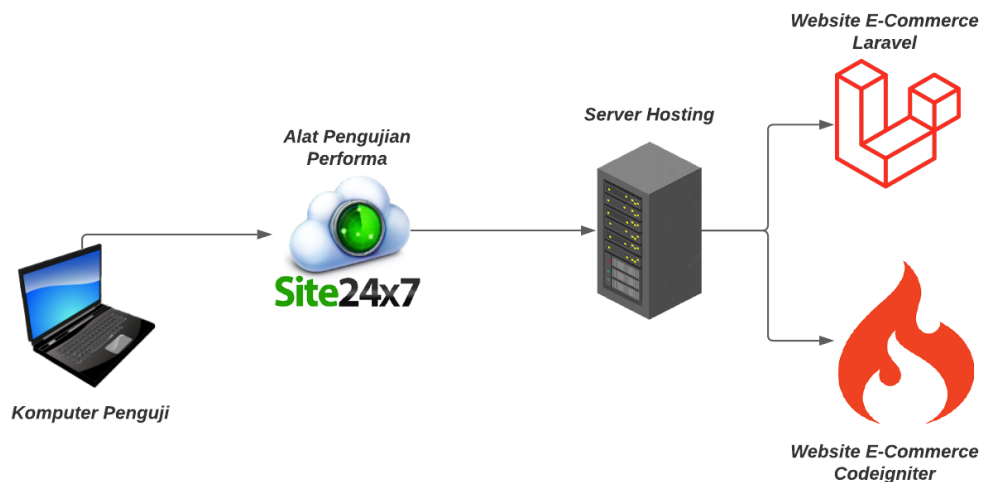
No	Parameter	Satuan	Alat
1	<i>Response Time</i>	<i>Milisecond (Ms)</i>	Site24x7.com
2	<i>Throughput</i>	<i>Kilobyte/second (KB/s)</i>	
3	<i>Page Load Time</i>	<i>Second (s)</i>	
4	<i>Request Per Second</i>	<i>Request/second</i>	Apache Benchmark
5	Waktu Eksekusi Transaksi Data REST	<i>Second (s)</i>	-

Dari tabel 3.1 dapat dilihat bahwa pengujian performa dilakukan dengan alat yang berbeda, pada parameter *response time*, *throughput*, dan *page load time* menggunakan alat *monitoring website* yaitu *site24x7.com*. Sedangkan pada parameter *request per second* menggunakan alat *Apache Benchmark*. Dan parameter waktu eksekusi transaksi data REST tidak menggunakan alat, melainkan

perhitungan yang telah dijelaskan pada bab sebelumnya. Oleh karena itu pengujian performa ini dibagi menjadi 3 bagian, sebagai berikut:

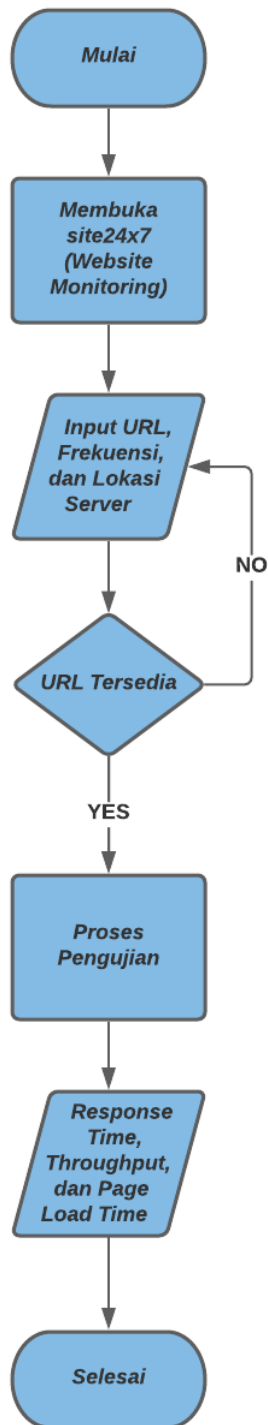
3.7.1 Pengujian *Response Time*, *Throughput*, dan *Page Load Time*

Untuk memberi gambaran mengenai pengujian, penulis membuat skenario pengujian yang dapat dilihat pada gambar dibawah ini:



Gambar 3.10 Skenario Pengujian *Response Time*, *Throughput*, dan *Page Load Time*.

Dari gambar 3.10 kita dapat melihat skenario pengujian, dimana melibatkan komputer pengujian yang hanya digunakan untuk membuka site24x7 dan melakukan pengaturan awal pengujian, lalu ada site24x7 yang bertugas melakukan pengujian dengan mengirimkan *request* ke *website e-commerce* Laravel dan Codeigniter. Selanjutnya ada *server hosting* yang akan merespon *request* dari site24x7, sehingga site24x7 dapat mengakses *website e-commerce* Laravel dan Codeigniter, lalu menghitung *response time*, *throughput*, dan *page load time* pada masing-masing *website* tersebut. Setelah mengetahui skenario pada pengujian ini, selanjutnya penulis membuat *flowchart* pengujian yang ada pada gambar 3.11.



Gambar 3.11 *Flowchart* Pengujian *Response Time*, *Throughput*, dan *Page Load Time*.

Dari gambar 3.11 kita dapat melihat bahwa untuk dapat memulai pengujian diharuskan memasukan parameter terlebih dahulu. Parameter yang dimaksud adalah URL, frekuensi, dan lokasi *server*. Dan berikut ini adalah parameter yang digunakan penulis dalam pengujian ini.

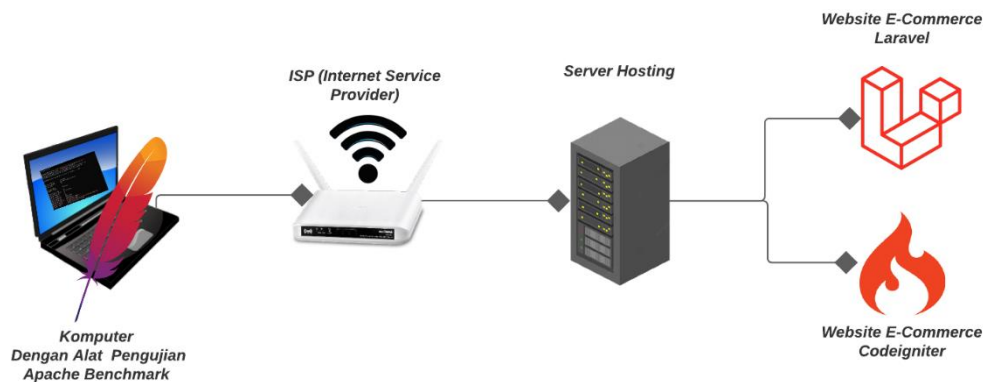
Tabel 3.2 Parameter Pengujian Performa Pada Site24x7.

URL	Frekuensi	Lokasi <i>Server</i>
https://laravel.sasparts-ci.my.id	10 Menit	Jakarta
https://codeigniter.sasparts-ci.my.id		

Parameter URL digunakan untuk memberitahukan alamat *website* yang akan diuji pada *platform* site24x7. Lalu parameter frekuensi digunakan untuk mengatur periode waktu pada setiap kali pengujian. Selanjutnya parameter lokasi *server* digunakan untuk menentukan lokasi *server* site24x7 saat melakukan pengujian. Jadi dari parameter-parameter yang ditunjukkan pada tabel 3.2, site24x7 akan melakukan pengujian performa *website* selama 10 menit sekali dan pengujian dilakukan pada *server* Jakarta. Sehingga untuk mendapatkan data yang cukup dalam pengujian performa ini, pengujian akan dilakukan selama 6 jam.

3.7.2 Pengujian *Request Per Second*

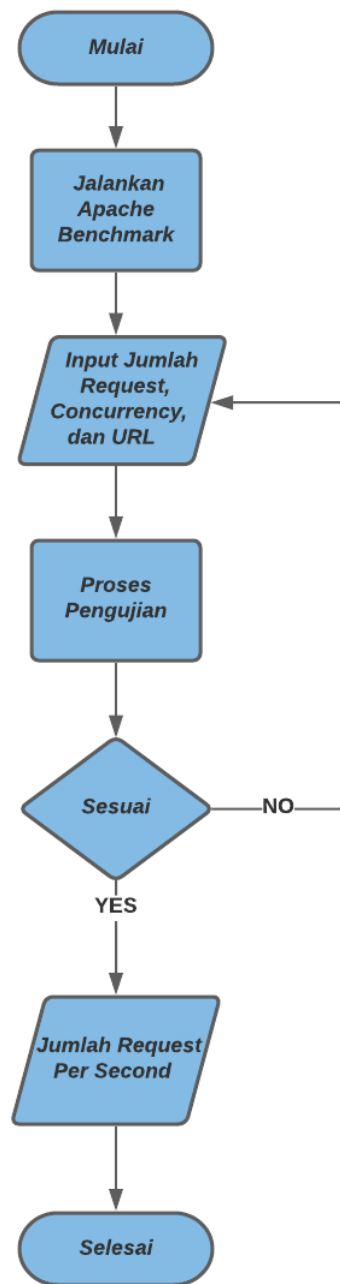
Sama seperti sebelumnya penulis akan membuat skenario pada pengujian ini untuk menggambarkan proses pengujian yang terjadi, sebagai berikut:



Gambar 3.12 Skenario Pengujian *Request Per Second*.

Dari gambar 3.12 kita dapat melihat skenario pengujian yang sedikit berbeda dari skenario pengujian sebelumnya yang ada pada gambar 3.10. Dimana pada pengujian ini komputer penguji digunakan untuk menjalankan alat pengujian *Apache Benchmark*, sedangkan pada pengujian sebelumnya alat pengujian *site24x7* dijalankan pada *server* nya sendiri dan tidak melibatkan komputer penguji selain melakukan pengaturan awal.

Lalu karena pengujian ini dilakukan pada komputer penguji, maka dari itu dibutuhkan penyedia layanan internet (ISP), agar *Apache Benchmark* yang berjalan pada komputer penguji dapat mengirimkan *request* ke *server hosting*. Selanjutnya terdapat *server hosting* yang bertugas merespon *request* dari *Apache Benchmark*, sehingga *Apache Benchmark* dapat mengakses *website e-commerce*, dan mulai menghitung *request per* pada masing-masing *website*. Dan penulis juga telah membuat *flowchart* pengujian *request per second* yang dapat dilihat pada gambar 3.13.



Gambar 3.13 *Flowchart Pengujian Request Per Second.*

Pada pengujian ini ISP yang digunakan memiliki kecepatan internet sampai dengan 15 Mbps. Dan untuk melakukan pengujian menggunakan alat Apache *Benchmark* dibutuhkan beberapa parameter, seperti banyaknya *request* yang ingin dikirimkan, berapa jumlah *request* yang dikirimkan dalam satu waktu (*concurrency*), dan URL *website* yang ingin diuji. Berikut ini adalah parameter-parameter yang digunakan dalam pengujian ini.

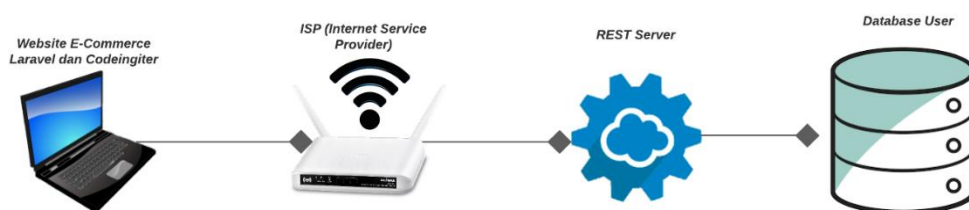
Tabel 3.3 Parameter Pengujian Performa Pada Apache *Benchmark*.

URL	Jumlah <i>Request</i> (n)	<i>Concurrency</i> (c)
www.laravel.sasparts-ci.my.id	100	100
www.codeigniter.sasparts-ci.my.id		

Jadi pengujian ini akan dilakukan dengan mengirimkan 100 *request* pada waktu yang bersamaan ke masing-masing *website e-commerce*. Dan untuk mendapatkan hasil pengujian yang baik, serta memiliki konsistensi pada nilai yang dihasilkan, pengujian akan dilakukan sebanyak 10 kali dengan jumlah *request* dan *concurrency* yang sama.

3.7.3 Pengujian Waktu Eksekusi Transaksi Data Dengan Arsitektur REST

Pengujian ini dilakukan untuk menguji performa *website* dalam melakukan transaksi data dengan arsitektur REST, yang berarti pengujian akan terlibat dengan sebuah REST *server* yang mengolah sumber data. Untuk menggambarkan pengujian ini penulis membuat skenario pengujian sebagai berikut.



Gambar 3.14 Skenario Pengujian Waktu Eksekusi Transaksi Data Dengan Arsitektur REST.

Dapat dilihat pada gambar 3.14 untuk dapat melakukan pengujian, *website e-commerce* harus terhubung dengan penyedia layanan internet, sehingga *website e-commerce* dapat melakukan *request* transaksi data ke *REST server* yang mengolah data *user*. Untuk mengetahui jenis transaksi data apa saja yang dilakukan dan berapa kali pengujian dilakukan, dapat dilihat pada tabel berikut.

Tabel 3.4 Jenis Transaksi Data Dengan REST Server.

Jenis Transaksi	Method	Keterangan	N
Tambah Data (<i>Create</i>)	POST	Menambah 1 baris data <i>user</i> ke <i>REST server</i>	10
Ambil Data (<i>Read</i>)	GET	Mengambil 200 baris data <i>user</i> dari <i>REST server</i>	10
Ubah Data (<i>Update</i>)	PUT	Mengubah 1 data <i>user</i> pada <i>field</i> akses level di <i>REST server</i>	10
Hapus Data (<i>Delete</i>)	DELETE	Menghapus 1 baris data <i>user</i> dari <i>REST server</i>	10

Dan untuk data pada pengujian ini, *REST server* memiliki *database* yang menyimpan 200 baris data *user* dengan struktur tabel sebagai berikut.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_user	int(9)	UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	nama	varchar(100)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	3	email	varchar(100)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	4	username	varchar(100)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	5	password	varchar(255)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	6	akses_level	varchar(50)	utf8_general_ci	No	None			Change Drop More
<input type="checkbox"/>	7	tanggal_update	timestamp		Yes	current_timestamp()			Change Drop More

Gambar 3.15 Struktur Tabel *User* Pada *Database REST Server*.

Dari tabel 3.4 dapat dilihat bahwa pengujian ini akan dilakukan sebanyak 10 kali pada setiap jenis transaksi data (CRUD). Sama seperti sebelumnya hal ini dilakukan untuk mendapatkan hasil pengujian yang baik, serta memiliki konsistensi pada nilai yang dihasilkan. Dalam pengujian ini penyedia layanan internet yang digunakan memiliki kecepatan sampai dengan 15 Mbps. Lalu karena pengujian dilakukan sebanyak 10 kali, maka dari itu dibutuhkan 10 data baru untuk digunakan pada pengujian transaksi tambah, ubah, dan hapus data. Berikut ini adalah data yang digunakan.

Tabel 3.5 Data Pengujian Transaksi *Create, Update, dan Delete*.

No	Nama	Email	Username	Password	Akses Level
1	Ilham Akbar	ilham@gmail.com	ilham123	12345	Admin
2	Afrian Saputra	afrian@gmail.com	afrian123	12345	Admin
3	Roki Pratama	roki@gmail.com	roki123	12345	Admin
4	Martinus Endar	tinus@gmail.com	tinus123	12345	Admin
5	Arya Gunawan	arya@gmail.com	arya123	12345	Admin
6	Putu Agus	putu@gmail.com	putu123	12345	Admin
7	Yosep Saputra	yosep@gmail.com	yosep123	12345	Admin
8	Robi Iskandar	robi@gmail.com	robi123	12345	Admin
9	Wahyu Saputra	wahyu@gmail.com	wahyu123	12345	Admin
10	Ahmad Adi	adi@gmail.com	adi123	12345	Admin

Tabel diatas adalah data untuk pengujian transaksi tambah, ubah, dan hapus data. Lalu pada pengujian transaksi data ubah, dilakukan dengan mengubah nilai Admin pada kolom Akses Level menjadi *User*. Pada pengujian ini penulis tidak menyertakan *flowchart*, karena pengujian ini tidak menggunakan alat pihak ketiga. Pengujian ini dilakukan pada halaman admin *website e-commerce* dengan melakukan transaksi data pada umumnya, namun pada saat transaksi data selesai, akan tampil sebuah *alert* yang menunjukkan waktu eksekusi transaksi data tersebut.