

**DEPARTEMEN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2023**

**AYSUKA ANSARI**

**ANALISIS PERBANDINGAN RESPONSE TIME RESTFUL API PADA FRAMEWORK ROCKET DAN EXPRESS.JS MENGGUNAKAN LOAD TESTING**

**ABSTRAK**

Pada era digital yang semakin berkembang, penggunaan teknologi Internet semakin meluas dan menjadi salah satu kebutuhan utama dalam menunjang aktivitas sehari-hari. Salah satu teknologi Internet yang hampir selalu digunakan adalah website. Melalui website pengguna dapat mengakses segala jenis informasi yang diinginkan dengan mudah dan cepat. Penggunaan teknologi website yang semakin luas memunculkan kebutuhan akan aplikasi web dengan performa yang baik dan responsif. Salah satu komponen yang berpengaruh pada performa suatu website adalah API dalam hal ini RESTful API. RESTful API telah menjadi pilihan yang populer dalam menyediakan layanan berupa komunikasi antara sistem yang berbeda. Namun, performa dari RESTful API itu sendiri bergantung kepada bahasa pemrograman khususnya framework yang berperan dalam mengimplementasikan suatu arsitektur API. Oleh karena itu, diperlukan pemilihan *framework* web yang tepat agar diperoleh aplikasi web dengan performa tinggi dan sesuai dengan kebutuhan pengembangan. Penelitian ini bertujuan untuk melakukan analisis perbandingan *response time* pada RESTful API yang dikembangkan menggunakan *framework* web Rocket dan Express.JS melalui metode *load testing* untuk membantu para pengembang web dalam memilih *framework* web yang sesuai dengan kebutuhan mereka. Penelitian dilakukan dengan pendekatan komparatif. Metode *load testing* dilakukan untuk menguji performa API dalam menangani jumlah *request* yang tinggi secara bersamaan (*concurrent*). Load testing dilakukan dengan menyimulasikan kondisi beban kerja yang tinggi pada API dan menganalisis *response time* yang dihasilkan. Penelitian ini menggunakan Apache JMeter sebagai alat untuk melakukan *load testing*. Setelah proses simulasi dilakukan, data akan disimpan dan kemudian divisualisasikan dalam bentuk plot sederhana untuk kemudian dianalisis. Analisis dilakukan untuk melihat *framework* mana yang memberikan performa yang lebih baik dalam menangani *request* dengan jumlah yang bervariasi dengan *response time* tertentu. Hasil penelitian ini diharapkan dapat memberikan pemahaman yang lebih mendalam terkait *response time* dari RESTful API yang dikembangkan menggunakan framework Rocket dan Express.JS. Selain itu, informasi yang dihasilkan dari penelitian ini diharapkan dapat digunakan sebagai acuan dalam memilih *web framework* yang sesuai dengan kebutuhan dan preferensi pengembang web dalam mengembangkan aplikasi web.

**ABSTRACT**

In the rapidly evolving digital era, the use of Internet technology is becoming increasingly widespread and has become one of the primary needs to support daily activities. One of the Internet technologies that is almost always used is a website. Through a website, users can easily and quickly access various types of desired information. The growing use of website technology has created a need for web applications with good performance and responsiveness. One component that influences the performance of a website is the API, specifically the RESTful API. RESTful API has become a popular choice for providing services through communication between different systems. However, the performance of the RESTful API itself depends on the programming language, particularly the framework used to implement an API architecture. Therefore, selecting the right web framework is necessary to obtain a high-performance web application that meets development needs. This research aims to conduct an analysis and comparison of response times on RESTful APIs developed using the Rocket and Express.js web frameworks through load testing methods to assist web developers in choosing the appropriate web framework for their needs. The research is conducted using a comparative approach. Load testing is performed to test the performance of the API in handling a high number of concurrent requests. Load testing is done by simulating high workload conditions on the API and analyzing the resulting response times. Apache JMeter is used as the tool for load testing. After the simulation process is completed, the data is stored and then visualized in the form of simple plots for analysis. The analysis is conducted to determine which framework performs better in handling requests with varying numbers and specific response times. The results of this research are expected to provide a deeper understanding of the response times of RESTful APIs developed using the Rocket and Express.js frameworks. Additionally, the information generated from this research is expected to be used as a reference in selecting the appropriate web framework based on the needs and preferences of web developers in developing web applications.

**DAFTAR ISI**

[DAFTAR TABEL iii](#_30j0zll)

[DAFTAR GAMBAR iii](#_1fob9te)

[DAFTAR LAMPIRAN iii](#_3znysh7)

[I](#_2et92p0) PENDAHULUAN 1

[I.1](#_3dy6vkm) Latar Belakang 1

[I.2](#_1t3h5sf) Rumusan Masalah 1

[I.3](#_4d34og8) Tujuan 2

[I.4](#_17dp8vu) Manfaat 2

[I.5](#_26in1rg) Ruang Lingkup 2

[II](#_44sinio) TINJAUAN PUSTAKA 4

[II.1](#_z337ya) Internet 4

[II.2](#_3j2qqm3) Website 4

II.3 HTTP 5

II.4 API 6

II.5 Protokol REST 6

II.6 RESTful API 7

II.7 JSON 7

II.8 Web Framework 8

II.9 Framework Rocket 8

II.10 Framework Express.JS 8

II.11 Load Testing 9

II.12 Apache JMeter 9

[III](#_1y810tw) METODE 3

[3.1](#_4i7ojhp) Waktu dan Tempat 3

[3.2](#_2xcytpi) Alat dan Bahan 3

[3.3](#_1ci93xb) Prosedur Kerja 3

[3.4](#_3whwml4) Analisis Data 3

[IV](#_qsh70q) HASIL DAN PEMBAHASAN (terpisah atau gabung) 3

[4.1](#_3as4poj) Hasil 3

[4.2](#_1pxezwc) Pembahasan 3

[V](#_2p2csry) SIMPULAN DAN SARAN 3

[5.1](#_147n2zr) Simpulan 3

[5.2](#_3o7alnk) Saran 3

[DAFTAR PUSTAKA 12](#_ihv636)

[LAMPIRAN 3](#_32hioqz)

[RIWAYAT HIDUP 3](#_2grqrue)

**DAFTAR TABEL**(jika tabel hanya satu tidak perlu dibuat daftar)

1. [Tingkat kekerasan dan kandungan gula buah pisang ambon pada suhu simpan yang berbeda dan pemberian putresina 3](#_3rdcrjn)
2. [Tingkat kekerasan buah pisang raja pada suhu simpan yang berbeda dan pemberian putresina](about:blank) 3

**DAFTAR GAMBAR**(jika gambar hanya satu tidak perlu dibuat daftar)

1. [Contoh gambar 3](#_2s8eyo1)
2. [Contoh judul gambar lebih dari satu baris maka baris kedua dimulai tepat di bawah huruf pertama judul gambar 3](#_lnxbz9)

**DAFTAR LAMPIRAN**(jika lampiran hanya satu tidak perlu dibuat daftar)

1. [Lampiran 1 Rata-rata dan simpangan baku beberapa sifat físik dan kimia tanah dari 78 contoh tanah di Kebun Percobaan Ciheuleut 3](#_1hmsyys)
2. [Lampiran 2 Umur, indeks luas daun, dan hasil biji kering jagung yang ditanam pada lima ketinggian tempat 3](#_41mghml)

# PENDAHULUAN

* 1. **Latar Belakang**

Pada era digital yang semakin berkembang, penggunaan teknologi Internet semakin meluas dan menjadi salah satu kebutuhan utama dalam menunjang aktivitas sehari-hari. Salah satu teknologi Internet yang hampir selalu digunakan adalah website. Melalui website pengguna dapat mengakses segala jenis informasi yang diinginkan dengan mudah dan cepat. Penggunaan website yang semakin meningkat kelak mendorong para pengembang aplikasi web untuk terus mengembangkan dan meningkatkan performa aplikasi web yang dibuat agar dapat memberikan pengalaman pengguna yang lebih baik. Salah satu cara untuk meningkatkan performa aplikasi web adalah dengan menyediakan API untuk menangani request yang diberikan oleh pengguna.

RESTful API merupakan salah satu teknologi *web service* yang paling populer saat ini. Dengan menggunakan RESTful API, pengembang dapat membuat aplikasi web yang dapat berkomunikasi dengan berbagai sistem, bahasa pemrograman, atau platform yang berbeda. Hal ini memungkinkan aplikasi web dapat diakses oleh berbagai perangkat seperti komputer, *smartphone*, tablet, dan lain-lain.

Untuk memudahkan pengembangan aplikasi web yang menggunakan RESTful API maka terdapat berbagai macam Web Framework yang dapat digunakan. Web Framework merupakan kerangka kerja atau serangkaian kode yang berisi fitur dan komponen inti dari aplikasi web yang dapat digunakan untuk mengembangkan suatu aplikasi web tanpa harus memulai proses nya dari nol. Terdapat berbagai jenis web framework yang dapat digunakan oleh seorang *developer* untuk mengembangkan aplikasi web dengan basis bahasa pemrograman yang juga bervariasi. Pada penelitian kali ini, terdapat dua buah web framework yang akan digunakan, yaitu Rocket dan Express.JS.

Rocket adalah web framework yang dibangun menggunakan bahasa pemrograman Rust. Rocket menyediakan banyak fitur yang dapat membantu *developer* untuk membangun aplikasi web dengan performa yang cepat dan tingkat keamanan yang baik. Hal ini dikarenakan bahasa pemrograman Rust berfokus pada performa *runtime*, *memory safety*, dan *type safety*. Sedangkan Express.JS merupakan web framework yang dibangun menggunakan bahasa pemrograman JavaScript. Express.JS dipilih pada penelitian kali ini dikarenakan popularitas nya di kalangan *developer* untuk mengembangkan aplikasi web. Selain itu, ExpressJS juga memiliki komunitas yang besar dan dokumentasi yang lengkap sehingga memudahkan *developer* untuk menggunakan dan beradaptasi dengan teknologi ini.

Dalam mengembangkan aplikasi web yang menggunakan RESTful API dengan Web Framework, penting untuk memperhatikan performa aplikasi. Performa suatu aplikasi dapat diukur menggunakan suatu metode tertentu. Terdapat beberapa jenis metode yang umum digunakan untuk mengevaluasi performa aplikasi seperti *stress testing, soak testing, spike testing, load testing, endurance testing, dan scalability testing*. Penelitian ini hanya akan berforkus pada satu jenis metode saja, yaitu *load testing*. Hal ini dikarenakan metode *load testing* mudah untuk diimplementasikan dan dapat memberikan gambaran umum dari performa suatu aplikasi dengan cukup baik.

*Load testing* merupakan salah satu metode pengujian performa aplikasi yang berfokus pada pengukuran kemampuan sistem dalam menangani beban kerja yang tinggi. Metode ini bertujuan untuk memastikan bahwa sistem tetap berfungsi dengan baik dan memberikan respons yang memadai dalam kondisi beban maksimal. Terdapat beberapa variabel yang dapat dianalisis setelah *load testing* dijalankan seperti *response time*, *throughput,* dan *error rate.* Penelitian ini hanya akan berfokus pada satu variabel saja, yaitu *response* time.

Pada konteks REST API, metode *load testing* digunakan untuk mengukur kinerja API dalam menangani jumlah permintaan yang tinggi. Hal ini dilakukan dengan menyimulasikan kondisi beban yang tinggi kepada API untuk kemudian dianalisis *response time*-nya. Proses simulasi ini dapat dilakukan secara manual maupun menggunakan sebuah aplikasi. Terdapat berbagai jenis aplikasi yang dapat digunakan untuk menyimulasikan *load testing* seperti Gatling, LoadRunner, Locust, Tsung, JMeter, dan lain-lain. Namun, penelitian ini hanya akan berfokus pada penggunaan satu aplikasi saja, yaitu Apache JMeter. Aplikasi ini dipilih dikarenakan kehandalannya dalam menguji kinerja serta skalabilitas dari aplikasi web.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang permasalahan yang telah dijelaskan maka rumusan masalah yang didapatkan yaitu:

* + - 1. Bagaimana performa RESTful API pada framework Rocket dan ExpressJS dalam menangani *user request* jika dilihat dari sisi *response time* ?
      2. Apa perbedaan performa antara framework Rocket dan Express.JS dalam menangani *user request* pada RESTful API jika dilihat dari sisi *response time* ?
      3. Bagaimana dampak penggunaan Apache JMeter terhadap performa RESTful API pada framework Rocket dan ExpressJS ?
      4. Apakah ada perbedaan signifikan antara performa RESTful API pada framework Rocket dan Express.JS dalam kondisi penggunaan Apache JMeter ?
      5. Bagaimana cara untuk meningkatkan performa RESTful API pada framework Rocket dan Express.JS berdasarkan hasil analisis perbandingan *response time* menggunakan Apache JMeter ?
  1. **Tujuan**

Tujuan dari penelitan ini adalah untuk membandingkan performa RESTful API pada dua framework yaitu Rocket dan Express.JS dengan menggunakan Apache JMeter. Performa dari masing-masing framework kemudian dianalisis dan hasilnya digunakan untuk menentukan framework mana yang lebih baik digunakan jika ingin memprioritaskan *response time* yang cepat. Selain itu, hasil penelitian ini diharapkan dapat membantu para pengembang web untuk memahami lebih dalam terkait perbedaan serta kelebihan dan kekurangan dari framework yang diteliti.

* 1. **Manfaat**

Berikut adalah beberapa manfaat penelitan yang diharapkan dapat dihasilkan dari penelitian ini:

Memberikan pemahaman yang lebih baik tentang performa RESTful API pada framework Rocket dan Express.JS

Memberikan pemahaman yang lebih baik tentang penggunaan *tool* Apache JMeter dalam menguji performa RESTful API dari framework Rocket dan Express.JS

Memberikan rekomendasi terkait framework mana yang lebih baik digunakan dalam mengembangkan aplikasi web yang memerlukan RESTful API

Memberikan masukan bagi *developer* *web* yang menggunakan framework Rocket dan Express.JS untuk meningkatkan performa framework mereka dalam mengelola RESTful API

Meningkatkan pemahaman tentang pemilihan teknologi yang tepat dalam mengembangkan aplikasi web dengan fokus pada performa dan penggunaan RESTful API.

* 1. **Ruang Lingkup**

Berikut adalah ruang lingkup dari penelitan yang akan dilakukan:

* + - 1. Mengimplementasikan RESTful API secara sederhana menggunakan framework Rocket dan Express.JS
      2. Membuat suatu fungsi sederhana yang dijadikan sebagai parameter pengujian performa dari masing-masing framework. Dalam hal ini akan digunakan 3 buah fungsi, yaitu fungsi fibonacci, fungsi faktorial, serta fungsi penjumlahan n buah bilangan bulat pertama secara iteratif.
      3. Menampilkan hasil *response time* dari masing-masing framework menggunakan Apache JMeter
      4. Menganalisis *response time* dari kedua framework berdasarkan hasil yang diperoleh dari Apache JMeter
      5. Evaluasi framework Rocket dan Express.JS dalam pengembangan aplikasi web dengan RESTful API.

# TINJAUAN PUSTAKA

* 1. **Internet**

Internet merupakan kependekan dari “Interconnected Network”, yaitu istilah yang merujuk kepada suatu jaringan komputer global yang terdiri dari jutaan perangkat komputer yang saling terhubung satu sama lain. Jaringan berskala masif ini memungkinkan pengguna nya untuk saling bertukar informasi dan berkomunikasi dengan berbagai cara atau yang disebut dengan istilah protokol seperti TCP/IP, HTTP, dan lain-lain. Internet menyediakan berbagai jenis layanan yang mendukung proses transfer informasi dan komunikasi seperti layanan email, transfer file, dan navigasi web. Pengertian ini sejalan dengan yang dijelaskan oleh Prakoso (2007) bahwa Internet merupakan sekumpulan komputer lokal yang saling terhubung menggunakan perangkat lunak internet dan protokol TCP/IP atau HTTP. Jaringan internet bersifat publik artinya dapat diakses oleh siapa pun dan kapan pun. Internet berkembang pesat dari jutaan bisnis, pendidikan, dan jaringan pemerintahan yang saling berhubungan dengan pengguna yang terdistribusi di lebih dari 200 negara (O’brian 2003). Frost (2003) juga menjelaskan bahwa pada dasarnya setiap komputer yang terhubung ke koneksi Internet menyimpan file pada penyimpanan lokal dan file ini kemudian dapat diakses dengan bebas oleh semua komputer lain yang juga terhubung ke Internet.

* 1. **Website**

Website merupakan sebuah aplikasi yang terdiri dari sekumpulan halaman web atau dokumen elektronik berupa teks, gambar, suara, animasi, dan video yang dapat diakses melalui internet menggunakan URL (Uniform Resource Locator) dan sebuah web browser (Hasugian dan Sudarto, 2018). Sebuah website biasanya berisi informasi atau konten tertentu yang dibuat oleh satu atau lebih pengguna atau pengembang, seperti situs web berita, situs web perusahaan, situs web toko online, dan lain-lain. Halaman website biasanya berbentuk dokumen dalam format Hyper Text Markup Language (HTML) dan dapat diakses melalui protokol HTTP atau HTTPS, yang digunakan untuk berbagi informasi dari server kepada pengguna melalui web browser (Sibero, 2013). Setiap website dapat diakses melalui IP atau domain yang bersifat unik, artinya satu IP atau domain hanya dapat digunakan oleh satu website saja. Selain itu, menurut Agung (2000), setiap halaman pada website saling terhubung satu sama lain beserta file-file yang terdapat di dalamnya, sehingga terdapat dependensi antara halaman dan file yang terdapat pada suatu website.

* 1. **HTTP**

“Untuk mengambil data dari web, maka kita harus menjadikan perangkat lunak yang kita miliki mampu berkomunikasi dengan *server* dan *web services*. Lingua franca dari cara berkomunikasi di web disebut sebagai HTTP, yaitu Hyper Text Transfer Protocol” (Munzert *et al.* 2014:101). Berdasarkan pemaparan di atas maka HTTP dapat diartikan sebagai sebuah protokol yang digunakan untuk melakukan proses pertukaran data antara client dan server pada Internet. HTTP dapat digunakan untuk berbagi data berupa teks, gambar, suara, video, animasi, dan lain-lain. Versi HTTP yang paling umum digunakan adalah HTTP/1.1 dan HTTP/2.

Setiap komunikasi antara client dan server pada HTTP terdiri dari sebuah request dan sebuah response. Struktur dari sebuah request adalah sebagai berikut:

* + - * 1. Request Line: berisi informasi mengenai jenis request yang dilakukan, URI (Uniform Resource Identifier) dari sumber daya yang diminta, dan versi HTTP yang digunakan.
        2. Headers: berisi informasi tambahan mengenai request seperti informasi tentang client, jenis dokumen yang diminta, dan lain-lain.
        3. Body (opsional): berisi data yang dikirimkan dari client ke server seperti konten formulir atau data file upload.

Sedangkan struktur dari sebuah response pada HTTP terdiri dari tiga bagian utama, yaitu:

Status line: berisi informasi status dari request yang dilakukan berupa kode status (misals 200 OK atau 404 Not Found), teks penjelasan mengenai status, dan versi HTTP yang digunakan.

Headers: berisi informasi tambahan mengenai response seperti informasi tentang server, jenis konten yang dikirimkan, dan lain-lain.

Body (opsional) berisi data yang dikirimkan dari server ke client seperti halaman HTML atau file gambar.

Kemudian pada setiap request yang dilakukan oleh client menuju server diperlukan sebuah *method* atau suatu aksi yang ingin dilakukan oleh client terhadap server. Pada HTTP terdapat 4 jenis method yang digunakan, yaitu:

GET: digunakan untuk mengambil informasi dari suatu *resource* pada server.

POST: digunakan untuk mengirimkan informasi dari client menuju server, seperti data formulir maupun *file upload*.

PUT: digunakan untuk memperbarui suatu *resource* yang terdapat pada server.

DELETE: digunakan untuk menghapus suatu *resource* yang terdapat pada server.

Komponen terakhir pada HTTP adalah Header. Header merupakan informasi tambahan yang dikirmkan bersamaan dengan suatu request atau response pada HTTP. Header berisi informasi mengenai jenis konten yang dikirimkan, informasi tentang client dan server, dan lain-lain. Beberapa contoh header HTTP yang umum digunakan adalah Content-type yang digunakan untuk menentukan jenis konten yang dikirimkan dan User-agent untuk mengidentifikasi jenis dan versi web browser yang digunakan oleh client.

* 1. **API**

API (Application Programming Interface) adalah representasi fungsi-fungsi pada logic basis data yang diakses oleh logic pada antarmuka secara programmatis (Rama *et al.* 2013). API (Application Programming Interface) atau Antarmuka Pemrograman Aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu (Ichwan dan Hakiky 2011). Application Programming Interface (API) memungkinkan dua aplikasi berbeda untuk saling berinteraksi dan saling bertukar data atau informasi. Dalam konteks pengembangan perangkat lunak, API mengacu pada serangkaian protokol, aturan, dan alat yang digunakan oleh para pengembang untuk membangun perangkat lunak. API juga memungkinkan para pengembang untuk mengakses dan menggunakan layanan atau fungsionalitas yang disediakan oleh aplikasi atau platform lain. Dengan demikian, API sangat penting dalam pengembangan perangkat lunak modern, karena memungkinkan integrasi dan interaksi antara berbagai sistem dan aplikasi yang berbeda. Contoh API yang umum digunakan adalah Google Maps API, Facebook API, Twitter API, dan masih banyak lagi.

* 1. **Protokol REST**

*REST (Representational State Transfer)* adalah arsitektur untuk mengembangkan sebuah *web services* atau API yang dapat diakses melalui jaringan. Menurut Fielding (2000) *REST* merupakan arsitektur utama dari *World Wide Web*. REST mengusulkan pendekatan arsitektur web yang didasarkan pada penggunaan protokol HTTP dengan memanfaatkan metode HTTP seperti GET, POST, PUT, DELETE, dan lain-lain untuk melakukan operasi pada sumber daya (resource) yang tersedia di web. Pada protokol REST, sumber daya diidentifikasi dengan sebuah URI (Uniform Resource Identifier), dan setiap operasi yang dilakukan pada sumber daya menggunakan metode HTTP yang sesuai. Selain itu, REST juga menyarankan penggunaan representasi data dalam format tertentu seperti JSON (JavaScript Object Notation) atau XML (eXtensible Markup Language) untuk mempermudah pertukaran data antara aplikasi.

* 1. **RESTful API**

RESTful API (Representational State Transfer API) adalah sebuah arsitektur web yang sering digunakan dalam pengembangan layanan web. Arsitektur ini didasarkan pada prinsip-prinsip REST yang mengutamakan arsitektur berbasis resource dan standar HTTP. RESTful API memungkinkan klien (client) dan server berkomunikasi secara terpisah tanpa mengenal status keduanya. Sistem RESTful menggunakan HTTP Request seperti GET, POST, PUT, DELETE, dll., untuk berkomunikasi dengan antarmuka. Data yang dikirimkan biasanya berupa format JSON atau XML, yang dapat digunakan oleh client untuk mengakses resource pada server. API yang menggunakan arsitektur RESTful dirancang untuk berjalan dengan mudah dan efisien, serta mudah diimplementasikan pada berbagai platform, seperti desktop, mobile, dan web. API RESTful juga memungkinkan pengembang untuk memisahkan antarmuka (frontend) dari backend, sehingga memudahkan dalam pengembangan aplikasi yang kompleks.

* 1. **JSON**

JSON (JavaScript Object Notation) merupakan suatu sintaks untuk menyimpan dan menukar data yang memiliki bobot yang ringan dan mudah dipahami. JSON dapat berisi berbagai nilai seperti bilangan, string, Boolean, array, object atau null, dan mudah diterjemahkan dan dihasilkan oleh komputer. JSON didasarkan pada bagian dari bahasa pemrograman javascript dan standar ECMA-262 Edisi ke-3 - Desember 1999. JSON berbentuk teks dan tidak bergantung pada bahasa pemrograman tertentu karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C seperti C, C++, C#, Java, JavaScript, Perl, Python, dan lainnya. Karena sifat-sifat tersebut, JSON dianggap sebagai format ideal untuk pertukaran data antar sistem (Juansyah 2015).

* 1. **Web Framework**

Menurut Budi Raharjo (2015:2) “*Web Application Framework* (WAF), atau sering disingkat *web framework* adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi web”. Proses pengembangan web dapat dilakukan dengan berbagai macam bahasa pemrograman. Artinya setiap bahasa pemrograman bisa membuat *framework* nya sendiri yang digunakan untuk mengembangkan web. Salah satu bahasa pemrograman dengan *framework web* terbanyak adalah PHP. Beberapa jenis dari *framework* yang menggunakan bahasa PHP adalah *CodeIgniter, Laravel, Symfony,* dan lain-lain. Selain itu, juga terdapat bahasa pemrograman lain dengan *web framework* nya sendiri seperti di JavaScript menggunakan *Express.JS,* bahasa Rust menggunakan *Rocket*, dan Python menggunakan *Django*. Setiap *web framework* memiliki pola desain atau arsitektur yang menjadi landasan dari pengembangan web. Sebagai contoh pada *Laravel* dan *Django* menggunakan arsitektur MVC (Model-View-Controller) untuk mengembangkan web berdasarkan *framework* tersebut.

* 1. **Framework Rocket**

Rocket merupakan sebuah *web framework* yang dapat digunakan untuk mengembangkan aplikasi web. Jenis *framework* ini dikembangkan dari bahasa pemrograman Rust yang tergolong masih baru namun popularitasnya terus meningkat dari waktu ke waktu. Rocket menyediakan fitur-fitur yang memudahkan pengembangan aplikasi web, seperti routing, middleware, dan integrasi dengan database. Rocket juga memperkenalkan konsep "type-level programming" yang memungkinkan programmer untuk menentukan kontrak antara client dan server secara tipe pada saat kompilasi, sehingga dapat mengurangi kesalahan pada saat runtime. Rocket juga memiliki sistem pengujian yang kuat untuk memastikan kualitas kode yang dibuat. Rocket merupakan salah satu pilihan populer untuk pengembangan aplikasi web dengan bahasa Rust.

* 1. **Framework Express.JS**

ExpressJS adalah sebuah framework web untuk bahasa pemrograman JavaScript yang dibangun di atas Node.js. Framework ini memberikan fitur dan alat yang membantu pengembangan aplikasi web secara cepat dan mudah. ExpressJS juga merupakan salah satu framework web yang paling populer dan banyak digunakan dalam pengembangan aplikasi web berbasis Node.js.

ExpressJS menyediakan berbagai fitur, seperti routing, handling permintaan HTTP, integrasi dengan template engine, dan manajemen middleware, yang mempermudah pengembangan aplikasi web. Selain itu, ExpressJS juga mendukung penggunaan berbagai macam plugin atau modul tambahan untuk memperluas fitur dan kemampuannya dalam mengembangkan aplikasi web.

* 1. **Load Testing**

Load Testing merupakan sebuah metode pengujian performa suatu sistem dalam menangani beban (load) atau *request* yang diberikan oleh beberapa *client* dalam waktu yang bersamaan (*concurrent*). Metode ini perlu dilakukan pada suatu layanan sistem yang berpotensi digunakan secara masif oleh *client*, misalnya layanan pemesanan barang pada suatu sistem berupa aplikasi *e-commerce*. Variabel yang umumnya dianalisis pada metode ini adalah *runtime* atau waktu yang diperlukan oleh server untuk mengolah *request* yang diberikan oleh *client*. Pada *load testing* terdapat komponen *load generator* yang berfungsi untuk membuat satu atau lebih *request*. Kemudian *request* yang terbentuk dikirimkan ke website yang diuji dan menunggu website untuk mengembalikan *response* dengan *response time* tertentu. Setiap *request* yang dibuat oleh *load generator* disebut dengan *virtual user*. Sebuah *load test* dikatakan valid hanya jika virtual user dapat merepresentasikan perilaku dari *real user* (Menasce 2002)*.*

* 1. **Apache JMeter**

Apache JMeter adalah sebuah perangkat lunak open source yang digunakan untuk melakukan pengujian kinerja pada aplikasi web. Apache JMeter menyediakan fitur untuk merekam, menguji, dan menganalisis kinerja aplikasi web atau layanan web dalam berbagai kondisi beban dan skenario penggunaan yang berbeda-beda. Dalam pengujian kinerja dan beban, Apache JMeter dapat mensimulasikan aksi pengguna dengan mengirimkan permintaan HTTP kepada aplikasi web atau layanan web secara bersamaan dalam jumlah besar, untuk mengetahui seberapa besar kapasitas, kecepatan, dan daya tahan aplikasi atau layanan tersebut dalam menangani beban. Dengan hasil pengujian ini, pengembang atau administrator sistem dapat menentukan apakah aplikasi atau layanan tersebut dapat menangani jumlah pengguna atau trafik yang diinginkan, atau perlu dilakukan perbaikan dan optimasi lebih lanjut.

# METODE

Metode penelitian yang digunakan pada penelitian ini adalah metode komparatif dengan cara membandingkan *response time* RESTful API pada framework web Rocket dan Express.JS. Pengujian dilakukan via *load testing* dengan mengirimkan sejumlah permintaan dengan jumlah tertentu untuk kemudian diukur *response time­*-nya pada masing-masing framework. Jumlah *request* akan divariasikan sebanyak 5 nilai yang berbeda. Hasil perbandingan akan dianalisis untuk menentukan framework mana yang memberikan kinerja lebih baik dari sisi *response time*.

* 1. **Waktu dan Tempat**

Penelitian ini dilaksanakan pada periode bulan Januari hingga Mei 2023. Proses pengumpulan data beserta analisis dilakukan di Wisma D’Campus, Dramaga. Pengujian dilakukan pada server lokal berupa laptop.

* 1. **Lingkungan Pengembangan**
* Perangkat Keras
  1. CPU: AMD Ryzen 3 3250U
  2. GPU: AMD Radeon Vega 3
  3. RAM: 8GB DDR4 dual-slot
  4. Penyimpanan: SSD 256 GB + HDD 1TB
* Perangkat Lunak
  + - * 1. OS Windows 10 Home Edition 64-bit
        2. Rust-Nightly (versi 1.71.0)
        3. Node.JS (versi 18.16.0)
        4. Framework Rocket (versi 0.5.0)
        5. Framework Express.JS (versi 4.17.1)
        6. Apache JMeter (versi 5.5)
        7. NGINX (versi 1.20.2)
        8. Visual Studio Code
        9. Koneksi Internet
        10. Server lokal berupa laptop
  1. **Prosedur Kerja**
* Persiapan Lingkungan Pengembangan

Melakukan instalasi Rust-Nightly dan Node.JS

Melakukan instalasi dan konfigurasi Rocket dan Express.js

Mempersiapkan server lokal dengan NGINX

* Implementasi RESTful API

1. Membangun RESTful API sederhana untuk Rocket dan Express.JS
2. Melakukan konfigurasi endpoint ke setiap aplikasi web

* Implementasi Benchmark Function

1. Membuat tiga buah fungsi untuk *benchmarking*, yaitu fungsi fibonacci, fungsi factorial, dan fungsi exhaustive\_sum
2. Melakukan konfigurasi endpoint ke fungsi yang sesuai
3. Melakukan konfigurasi parameter dari *request* GET agar bisa diolah oleh fungsi yang bersesuaian

* Konfigurasi *Load Testing*

1. Melakukan instalasi Apache JMeter
2. Membuat skenario pengujian berupa *request* GET ke endpoint yang tersedia pada RESTful API yang sudah dibuat sebelumnya
3. Mengatur jumlah simultan (virtual user) dengan nilai tertentu. Virtual User nantinya akan mengakses API secara simultan (*concurrent*). Jumlah simultan akan divariasikan sebanyak 5 buah nilai berbeda.

* Eksekusi *Load Testing*

1. Menjalankan skenario pengujian yang telah disiapkan sebelumnya
2. Mengamati *response time* yang diperoleh dari eksekusi skenario yang dijalankan

* Pengumpulan Data

1. Mengumpulkan data *response time* yang diperoleh dari hasil *load testing* untuk masing-masing framework
2. Menyimpan data *response time* dengan format tertentu, misal csv.
   1. **Analisis Data**

* Komparasi *response time* hasil *load testing*

1. Mengambil data *response time* yang telah disimpan sebelumnya
2. Membuat *plot* sederhana untuk memvisualisasikan data *response time* dari masing-masing framework
3. Menganalisis hasil *response time* melalui *plot* yang telah terbentuk

* Evaluasi Hasil

1. Menentukan framework mana yang lebih baik dan efisien dari sisi *response time*
2. Mempertimbangkan faktor-faktor yang dapat mempengaruhi hasil *response time* seperti algoritma, optimasi framework, dan lain-lain.

# HASIL DAN PEMBAHASAN (terpisah atau gabung)

Dalam penulisan hasil dan pembahasan dapat dipisah sebagai bab Hasil dan bab Pembahasan, atau digabung menjadi bab Hasil dan Pembahasan. Pemisahan atau penggabungan kedua bab ini bergantung pada bidang studi, atau sesuai dengan arahan pembimbing.

* 1. **Hasil**
  2. **Pembahasan**

# SIMPULAN DAN SARAN

* 1. **Simpulan**

Simpulan merupakan jawaban dari tujuan yang sudah ditentukan dan tidak dimaksudkan sebagai ringkasan hasil. Dalam Simpulan, penulis harus dan hanya menjawab masalah dan tujuan penelitian yang telah dirumuskan pada Pendahuluan. Simpulan merupakan generalisasi dari hasil penelitian dan argumentasi penulis, atau pernyataan singkat yang merupakan hakikat dari bab Hasil dan Pembahasan atau hasil pengujian berbagai hipotesis yang berkaitan.

Simpulan merupakan hasil penelitian yang boleh jadi telah dikemukakan dalam perumusan masalah dan telah diberi jawaban sementara berupa hipotesis. Dalam menulis simpulan, penulis harus membedakan dugaan, temuan, dan simpulan hasil studi. Pernyataan simpulan harus dilakukan secara cermat dan hati-hati. Penyampaian simpulan ini dapat dilakukan sebanyak 3 kali, yakni dalam Pembahasan, Simpulan, dan Abstrak sehingga diperlukan kecermatan untuk menyajikannya dengan ungkapan yang berbeda-beda.

* 1. **Saran**

Saran seyogianya mengarah ke implikasi atau tindakan lanjutan yang harus dilakukan sehubungan dengan temuan atau simpulan penulis. Saran yang dikemukakan harus berkaitan dengan pelaksanaan atau hasil penelitian. Dengan demikian saran ini mengemukakan hal-hal yang perlu diteliti lebih lanjut terutama untuk memperbaiki kelemahan atau kekurangan dalam penelitian yang dilakukan atau perbaikan asumsi yang diambil sehingga didapatkan hasil yang lebih baik. Jadi, saran tersebut harus diuraikan secara spesifik. Jangan menyarankan hal-hal yang tidak dianalisis dan dibahas dalam penelitian serta terkesan menggurui atau memuaskan keinginan peneliti. Untuk penelitian yang berkaitan dengan permasalahan kebijakan, tidak perlu menyarankan kebijakan yang tidak berkaitan dengan hasil penelitian.

**DAFTAR PUSTAKA**

Oetomo BSD, Widow E, Hartono E, Prakoso S. 2007. *Pengantar Teknologi Informasi Internet: Konsep dan Aplikasi*. Yogyakarta: Penerbit ANDI. hlm 119.

O'Brian, James A. 2003. *Introduction to Information System: Essentials For the E-business Enterprise*. Ed ke-11. New York: McGraw Hill Inc. hlm 10.

Strauss J, El-Ansary A, Frost R. 2003. *E-marketing International*. Ed ke-3. New Jersey: Prentice Hall. hlm 8.

Hasugian, Sudarto P. 2018. Perancangan website sebagai media promosi dan informasi. *JITSINET*. 3(1):82-83.

Sibero AF. 2013. *Web Programming Power Pack*. Yogyakarta: Mediakom.

Gregorius, Agung. 2000. *Desain Web Interaktif Dengan Frontpage 2000 dan Dreamweaver 4*. Jakarta: PT Elex Media Komputindo.

Munzert S, Rubba C, Meißner P, Nyhuis D. 2014. *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. New Jersey: John Wiley & Son. hlm 101.

Raharjo B. 2015. *Belajar Otodidak Framework Codeigniter*. Bandung: Informatika Bandung. hlm 2.

Rama, Maskeri G, Kak A. 2013. *Software – Practice and Experience*. New Jersey: Wiley Online Library.

Fielding RT. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. California: University of California.

Ichwan M, Hakiky F. 2011. *Pengukuran Kinerja Goodreads Application Programming Interface (API) Pada Aplikasi Mobile Android*. Bandung: Institut Teknologi Bandung.

Juansyah A. 2015. Pembangunan Aplikasi Child Tracker Berbasis Assisted-Global Positioning System (A-GPS) Dengan Platform Android. *Jurnal KOMPUTA*. 1(1).

Menasce DA. 2002. *Load testing of Web Sites*. *IEEE Internet Computing*. 6(4).

**LAMPIRAN**

Lampiran 1 Rata-rata dan simpangan baku beberapa sifat físik dan kimia tanah dari 78 contoh tanah di Kebun Percobaan Ciheuleut

|  |  |  |
| --- | --- | --- |
| Sifat | Rata-rata | Simpangan baku |
| Pasir (%) | 47.66 | 23.81 |
| Lempung (%) | 21.80 | 11.94 |
| Liat (%) | 30.72 | 18.09 |
| C-organik (%) | 0.61 | 0.57 |
| Rapatan isi (mg m-3) | 1.43 | 0.16 |
| KTK (mek 100 g-1 tanah)a | 18.08 | 17.09 |
| KAT pada KL (g g-1) | 23.62 | 10.80 |
| KAT pada TLP (g g-1) | 11.11 | 9.05 |

aBanyaknya 70 contoh tanah; KTK: kapasitas tukar kation, KAT: kadar air tanah, KL: kapasitas lapang, TLP: titik layu permanen.

Lampiran 2 Umur, indeks luas daun, dan hasil biji kering jagung yang ditanam pada lima ketinggian tempat

|  |  |  |  |
| --- | --- | --- | --- |
| Ketinggian  (m dpl) | Umur  (hari) | Indeks luas daun | Hasil  (ton ha-1) |
| 856 | 115 | 3.10 | 5.69 |
| 605 | 106 | 3.09 | 5.43 |
| 400 | 100 | 2.47 | 4.80 |
| 210 | 93 | 2.46 | 4.25 |
| 10 | 88 | 2.12 | 4.03 |

**RIWAYAT HIDUP**

Penulis dilahirkan di kota…. pada tanggal bulan tahun sebagai anak ke … dari pasangan bapak … dan ibu … Pendidikan sekolah menengah atas (SMA) ditempuh di sekolah … , dan lulus pada tahun …. Pada tahun ..., penulis diterima sebagai mahasiswa program sarjana (S-1) di Program Studi/Fakultas/Sekolah … di IPB.

Selama mengikuti program S-1, penulis aktif menjadi ... (riwayat dan pengalaman organisasi, asisten akademik, dan sebagainya). Penulis juga pernah mengikuti lomba karya … (riwayat kegiatan ilmiah) memperoleh atau pernah terpilih sebagai … (riwayat prestasi akademik).