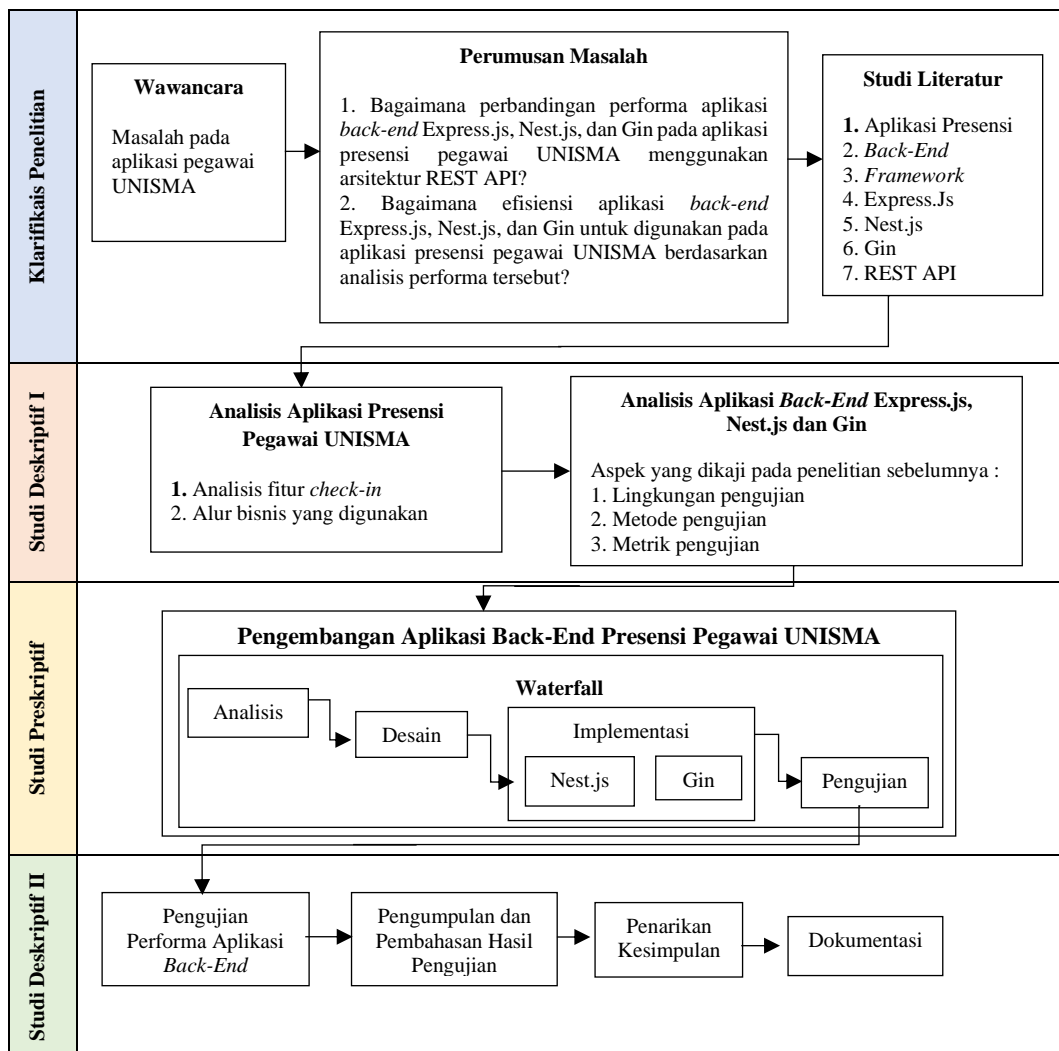


BAB III METODOLOGI PENELITIAN

3.1 Desain Penelitian

Penelitian ini mengadopsi pendekatan yang mengikuti *Design Research Methodology* (DRM). DRM adalah serangkaian langkah sistematis yang diterapkan dalam penelitian desain. Menurut Bertoni & Bertoni, penelitian desain fokus pada pengembangan aplikasi sebagai dukungan atau pendukung, serta melibatkan validasi pengetahuan secara terstruktur (Bertoni & Bertoni, 2019).



Gambar 3.1 Desain Penelitian

3.1.1 Klarifikasi Penelitian

Penelitian ini mengawali langkahnya dengan melakukan klarifikasi yang penting dalam menentukan fokus topik, yang pada kasus ini membahas perbandingan performa aplikasi *back-end* Express.js, Nest.js, dan Gin pada aplikasi presensi pegawai di UNISMA menggunakan arsitektur REST API. Proses awal melibatkan pengumpulan data yang dilakukan melalui wawancara bersama dengan *developer* untuk menemukan masalah pada kasus nyata yang terlampir pada Lampiran 1.

Kemudian studi literatur untuk merangkum teori-teori yang relevan sebagai landasan penelitian. Beberapa teori yang dianalisis dalam studi literatur mencakup aplikasi presensi pegawai, aplikasi *back-end*, *framework*, aplikasi presensi, *framework* Express.js, *framework* Nest.js, *framework* Gin, dan REST API. Penjelasan mendalam mengenai teori-teori ini tersaji pada Bab II dari penelitian ini. Terakhir, setelah melakukan analisis mendalam terhadap teori-teori yang telah dikaji, penelitian ini akan merumuskan masalah dan tujuan penelitian yang akan menjadi fokus utama dalam penelitian mengenai perbandingan performa aplikasi *back-end* Express.js, Nest.js, dan Gin pada aplikasi presensi pegawai di lingkungan UNISMA.

3.1.2 Studi Deskriptif I

Dalam tahap studi deskriptif mengenai perbandingan performa aplikasi *back-end* Express.js, Nest.js dan Gin pada Aplikasi Presensi Pegawai UNISMA menggunakan arsitektur REST API, analisis Aplikasi Presensi Pegawai UNISMA melibatkan beberapa poin utama yang menjadi fokus penelitian. Pertama, basis aplikasi yang dikembangkan yang mencakup infrastruktur inti dan kebutuhan fungsional aplikasi, seperti manajemen kehadiran, data mahasiswa, dan integrasi sistem. Kedua, fitur yang tersedia dalam aplikasi, termasuk fungsi-fungsi khusus yang memfasilitasi proses presensi, pelaporan, dan manajemen data mahasiswa. Selanjutnya, penelitian juga mempertimbangkan *endpoint* yang digunakan dalam Aplikasi Presensi Pegawai UNISMA, yang menjadi titik akses untuk berinteraksi dengan *back-end*. Analisis terperinci mengenai alur penggunaan presensi pegawai juga dilakukan untuk memahami secara menyeluruh bagaimana mahasiswa berinteraksi dengan aplikasi untuk melakukan proses presensi.

Fathoni Zikri Nugroho, 2024

ANALISIS PERBANDINGAN PERFORMA BACK-END EXPRESS.JS, NEST.JS, DAN GIN PADA APLIKASI PRESENSI PEGAWAI UNISMA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Kemudian, fokus analisis beralih ke aplikasi *back-end* Express.js, Nest.js, dan Gin. Terdapat beberapa aspek yang dijadikan pertimbangan berdasarkan penelitian sebelumnya. Pertama, lingkungan pengujian yang digunakan untuk membandingkan performa kedua aplikasi *back-end*, termasuk konfigurasi *server*, sistem operasi, dan spesifikasi teknis lainnya. Kedua, metodologi yang diterapkan dalam pengujian, yang mencakup proses pengujian, skenario penggunaan, dan pendekatan yang digunakan untuk membandingkan performa keduanya. Terakhir, metrik pengujian yang digunakan sebagai acuan untuk mengukur kinerja aplikasi, seperti *response time*, *throughput*, *CPU load*, dan *memory usage*.

3.1.3 Studi Preskriptif

Setelah melakukan pemahaman mengenai *back-end* aplikasi yang sudah ada dan melakukan perbandingan dengan penelitian-penelitian terdahulu yang relevan pada tahap studi deskriptif 1, selanjutnya dilakukan proses studi preskriptif. Studi preskriptif yang dilakukan yaitu dengan melakukan pengembangan aplikasi *back-end* menggunakan metode *waterfall* terhadap *framework* Nest.js dan Gin.

Metode *waterfall* merupakan metode yang sering digunakan dalam pengembangan perangkat lunak yang menggunakan pendekatan sistematis dan berurutan (Wahid, 2020). Tahapan proses dalam metode ini dimulai dari analisis atau perencanaan hingga tahap pemeliharaan (*maintenance*). Namun dalam penelitian ini tidak sampai tahap pemeliharaan, dikarenakan penelitian ini bertujuan untuk pembuktiaan konsep dan berfokus pada evaluasi kinerja.

3.1.3.1 Analisis

Pada tahap analisis untuk sistem yang akan dikembangkan dengan *framework* Nest.js dan Gin, penulis perlu menentukan kebutuhan-kebutuhan dengan meninjau sistem yang sudah ada (Express.js). Dengan mengidentifikasi kebutuhan fungsional dan non-fungsional serta batasan-batasan yang ada, pengembangan *back-end* dengan *framework* Nest.js dan Gin dapat direncanakan dan dieksekusi dengan lebih baik, memastikan bahwa sistem baru dapat memenuhi atau melebihi performa dan keandalan dari sistem yang sudah ada.

3.1.3.2 Desain

Proses perancangan akan melibatkan pembuatan desain arsitektur yang jelas untuk *back-end* yang akan dikembangkan dengan *framework* Nest.js dan Gin. Ini mencakup struktur logika bisnis dan arsitektur yang akan digunakan.

- a. Desain arsitektur, menjelaskan secara terperinci tentang integrasi antara sisi pengguna, API, dan basis data pada Aplikasi Presensi Pegawai UNISMA dengan pendekatan yang sama pada setiap *framework*.
- b. Diagram *flowchart*, menggambarkan alur kerja atau proses dalam sistem, memberikan gambaran bagaimana pergerakan data berdasarkan aturan dan fungsional sistem.

3.1.3.3 Implementasi

Tahap ini merupakan proses implementasi berdasarkan *back-end* aplikasi yang sudah ada yaitu *back-end* dengan Express.js. Dua aplikasi *back-end* akan dibuat, masing-masing menggunakan Nest.js dan Gin dengan arsitektur REST API sehingga perlu dilakukan sebanyak dua kali. Langkah-langkah implementasi meliputi pengembangan (pengkodean) aplikasi *back-end* dengan arsitektur REST API dengan *framework* Nest.js dan Gin.

3.1.3.4 Pengujian

Setelah pembuatan aplikasi pada tahap implementasi, dilakukan pengujian dengan metode *black box* oleh penulis. Metode *black box* adalah metode pengujian sistem yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal atau implementasi kode. Tujuan utamanya adalah memastikan bahwa aplikasi *back-end* berjalan sesuai kebutuhan sebelum melakukan pengujian performa pada kedua arsitektur yang digunakan, baik Express.js, Nest.js, maupun Gin dengan arsitektur REST API. Salah satu teknik yang digunakan dalam metode *black box* adalah *equivalence partitioning*. *Equivalence partitioning* adalah teknik pengujian perangkat lunak yang membagi domain masukan menjadi partisi nilai valid dan invalid (Irawan dkk., 2018). Teknik ini memerlukan data sampel yang bersifat representatif untuk setiap partisi nilai guna melakukan uji coba terhadap masukan.

3.1.4 Studi Deskriptif II

Dalam rangka menilai keberhasilan penelitian yang fokus pada perbandingan performa aplikasi *back-end* Express.js, Nest.js, dan Gin pada Aplikasi Presensi Pegawai UNISMA dengan arsitektur REST API, pengujian performa menjadi langkah krusial. Pengujian ini akan menggunakan aplikasi JMeter untuk mengevaluasi performa aplikasi yang telah dibangun. Setelah hasil pengujian diperoleh, analisis mendalam akan dilakukan guna mengevaluasi kualitas aplikasi dari aspek performa yang diuji. Hasil analisis akan menjadi bahan pengetahuan yang mendasari pembahasan tentang performa aplikasi dari berbagai sudut pandang. Analisis dilakukan dengan melakukan pengujian hipotesis dan menggunakan rumus persamaan efisiensi pada tiap-tiap hasil dari metrik yang diuji.

Selanjutnya, akan disusun kesimpulan atas hasil pengujian yang diperoleh. Di sini, kelebihan dan kekurangan dari kedua pendekatan, baik Express.js, Nest.js, maupun Gin pada arsitektur REST API, akan dibahas secara rinci. Kesimpulan ini akan menjawab rumusan masalah yang terbentuk dari hasil pengujian yang dilakukan. Selain kesimpulan, penelitian ini akan memberikan rekomendasi dan saran kepada pihak UNISMA untuk memilih aplikasi *back-end* yang terbaik untuk *website*. Dan juga saran kepada peneliti selanjutnya agar dapat melanjutkan serta meningkatkan penelitian ini. Hal ini bertujuan untuk memperbaiki atau memperluas aspek-aspek tertentu yang mungkin ditemukan dalam evaluasi performa aplikasi.

Terakhir, hasil seluruh rangkaian penelitian ini akan didokumentasikan dalam sebuah skripsi yang menggambarkan secara komprehensif langkah-langkah, analisis, kesimpulan, dan rekomendasi yang dihasilkan dari penelitian perbandingan performa aplikasi *back-end* Express.js, Nest.js, dan Gin pada Aplikasi Presensi Pegawai UNISMA menggunakan arsitektur REST API.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Adapun spesifikasi perangkat keras (*hardware*) yang digunakan untuk melakukan pengujian dan spesifikasi perangkat lunak (*software*) serta bahasa pemrograman yang dibutuhkan untuk sistem yang akan dibangun.

1. Kebutuhan Perangkat Keras

- a. Virtual Machine Google Cloud Platform
 - Seri E2
 - Tipe mesin e2-micro (2 vCPU)
 - RAM 4 GB
 - Memori 10 GB
 - b. Laptop
 - Prosesor Intel i7-10750H
 - RAM 16 GB
 - SSD 500 GB
2. Kebutuhan Perangkat Lunak
 - a. Microsoft Windows 11
 - b. Microsoft Word
 - c. Microsoft Excel
 - d. Microsoft Edge
 - e. Visual Studio Code
 - f. JMeter
 - g. SPSS
 - h. Postman API
 - i. MySQL
 3. Bahasa Pemrograman
 - a. JavaScript (dengan *runtime* Node.js versi 18.17.0)
 - b. Golang versi 1.20

3.2.2 Bahan Penelitian

Bahan yang akan digunakan selama penelitian meliputi adalah artikel ilmiah, jurnal, dokumentasi, dan lain sebagainya yang digunakan untuk membantu penulis untuk melakukan penelitian.

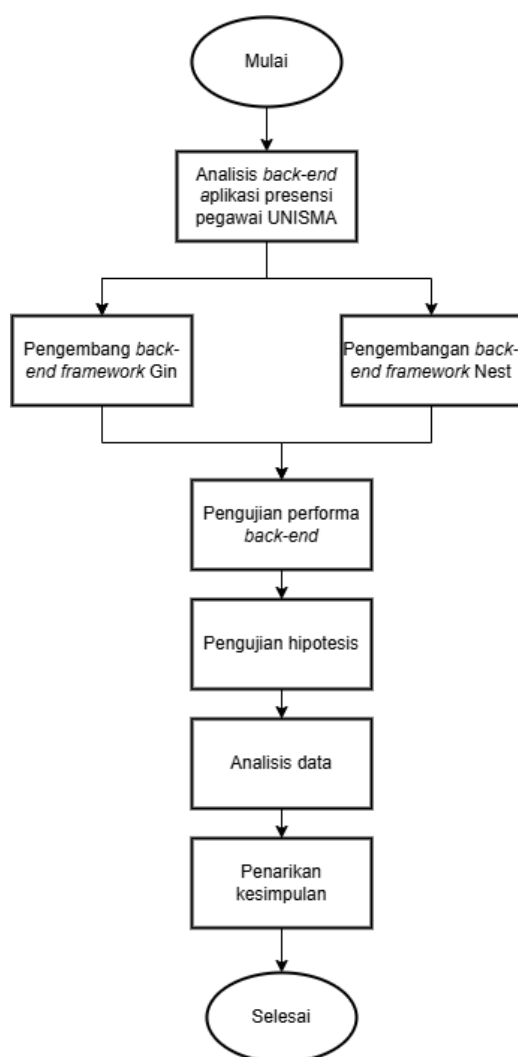
3.3 Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini adalah aplikasi JMeter. Aplikasi JMeter digunakan untuk pengujian kinerja yang fleksibel yang dapat mensimulasikan banyak pengguna aplikasi secara bersamaan sehingga dapat dilakukan evaluasi terhadap kinerja aplikasi (*Apache JMeter*, 2014). Pengujian

dengan jumlah pengguna yang dapat disesuaikan ini akan sangat berguna bagi penelitian ini karena dapat menggambarkan kondisi yang sebenarnya khususnya dalam studi kasus penelitian ini.

3.4 Prosedur Penelitian

Prosedur penelitian merupakan aspek yang penting dalam proses penelitian, untuk memastikan bahwa penelitian dapat berlangsung sesuai dengan rencana yang telah ditentukan.



Gambar 3.2 Prosedur Penelitian

Prosedur penelitian yang ditunjukkan pada Gambar 3.2 dimulai dengan analisis terhadap *back-end* aplikasi presensi pegawai UNISMA. Selanjutnya, dua jalur pengembangan dilakukan secara paralel, yaitu pengembangan *back-end* menggunakan *framework* Gin dan pengembangan *back-end* menggunakan

framework Nest.js. Setelah kedua *back-end* ini dikembangkan, tahapan berikutnya adalah pengujian performa dari dua *back-end* yang telah dikembangkan (Gin dan Nest.js) dan *back-end* yang sudah berjalan (Express.js). Hasil pengujian tersebut kemudian digunakan untuk melakukan pengujian hipotesis untuk mengetahui apakah ada perbedaan signifikan antar *framework*. Dilanjutkan dengan analisis data menggunakan rumus efisiensi dan visualisasi boxplot. Tahapan terakhir adalah penarikan kesimpulan berdasarkan analisis yang telah dilakukan, dan penelitian pun selesai.

3.5 Metode Pengambilan Data

Dalam penelitian ini terdapat 4 metrik yang diteliti oleh instrumen. Keempat metrik performa tersebut adalah *response time*, *throughput*, *CPU load*, dan *memory usage*. Mereka adalah metrik yang paling umum yang dapat dilacak sebagai titik data dasar yang baik dan paling dibutuhkan dalam mencapai *Quality of Service* (QoS). Untuk variabel *response time*, instrumen akan melakukan pengambilan data dengan cara mengirim permintaan sebanyak 100 pengguna yang dilakukan secara berulang dalam setiap sampel dan tidak ditentukan berapa lama waktunya. Evaluasi dilakukan terhadap *throughput* sebagai jumlah total permintaan dengan kinerja pengambilan data dalam setiap interval waktu 15 detik, dalam periode tertentu yang dapat memberikan gambaran tentang kinerja relatif dari kedua teknologi tersebut dalam konteks pengambilan data yang cepat. Lalu dalam waktu yang sama dilakukan pengukuran metrik *CPU load* dan *memory usage*.

Metode pengambilan data ini mengikuti pendekatan yang digunakan dalam penelitian oleh Lawi dkk., yang menekankan pentingnya pengulangan dan interval waktu tetap untuk memastikan reliabilitas dan validitas data performa yang diperoleh. Permintaan yang dikirimkan untuk seluruh metrik menggunakan *virtual user* atau pengguna sebanyak 100 agar data yang diambil menjadi representasi dari beban yang realistis (Lawi dkk., 2021). Sebanyak 100 pengguna sering kali dipilih sebagai batas aman yang memungkinkan pengujian dilakukan tanpa membebani perangkat keras secara berlebihan, sekaligus memberikan gambaran dasar tentang bagaimana sistem akan berperforma di bawah beban tertentu (Abbas dkk., 2017). Jumlah pengguna 100 juga merupakan beban yang dapat dikelola oleh *server*, sehingga tidak mengganggu tujuan dari penelitian (Schulz dkk., 2020).

Fathoni Zikri Nugroho, 2024

ANALISIS PERBANDINGAN PERFORMA BACK-END EXPRESS.JS, NEST.JS, DAN GIN PADA APLIKASI PRESENSI PEGAWAI UNISMA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Tabel 3.1
Metrik dan Proses Pengambilan Data

Metrik	Proses Pengambilan Data
<i>Average Response Time</i>	Waktu respons rata-rata dari 2000 tes dalam setiap sampel.
<i>Throughput</i>	Jumlah total permintaan yang berhasil diproses pada setiap interval waktu ke-i dengan mengumpulkan jumlah <i>response</i> yang berhasil diproses dalam rentang waktu tertentu.
<i>CPU Load</i>	Beban CPU diukur dan direkam setiap 15 untuk bersamaan dengan pengujian <i>throughput</i>
<i>Memory Usage</i>	Pengukuran penggunaan memori bersamaan dengan pengujian <i>throughput</i> dan beban CPU. Konsumsi memori dari disimpan setiap 15 detik.

3.6 Analisis Data

Setelah mendapatkan hasil pengukuran untuk keempat metrik tersebut, gunakan rumus untuk menghitung efisiensi kinerja. Penulis memastikan untuk memperoleh nilai dari pengujian tiap-tiap metrik (*response time*, *throughput*, *CPU load*, dan *memory usage*) dari masing-masing REST API untuk digunakan dalam rumus efisiensi. Rumus efisiensi kinerja yang digunakan dalam penelitian ini diadaptasi dari penelitian yang dilakukan oleh Lawi dkk. (2021) pada penelitian performa GraphQL dan REST API.

$$Efficiency = \left(1 - \frac{\min\{\eta_{Express.js}, \eta_{Nest.js}, \eta_{Gin}\}}{\max\{\eta_{Express.js}, \eta_{Nest.js}, \eta_{Gin}\}} \right) \times 100\% \quad (1)$$

Dimana $\eta_{Express.js}$ adalah hasil pengukuran kinerja REST API Express.js, $\eta_{Nest.js}$ merupakan hasil pengukuran kinerja REST API Nest.js, sedangkan η_{Gin} adalah hasil pengukuran kinerja REST API Gin. Masing-masing hasil pengukuran menggunakan variabel yang relevan.

3.7 Hipotesis

Pengujian hipotesis performa *back-end* dilakukan menggunakan uji anova. *Analysis of variance* atau anova merupakan salah satu teknik analisis *multivariate* yang berfungsi untuk membedakan rerata lebih dari dua kelompok data dengan cara membandingkan variansinya (Riadi dkk., 2020). Tujuannya untuk mengukur signifikansi perbedaan antara rata-rata kelompok metrik performa yang diuji. Uji anova yang digunakan yaitu uji anova jenis satu arah karena hanya terdapat satu faktor yang mempengaruhi variabel respon, yaitu *framework*. Untuk menjalankan uji anova, terdapat syarat salah satunya yaitu variansi dari setiap kelompok harus sama. Maka dari itu perlu dilakukan uji homogenitas terlebih dahulu untuk mengetahui apakah memenuhi asumsi uji anova. Metode pengujian hipotesis ini mengikuti pendekatan yang digunakan dalam penelitian oleh Zahir & Akramunnisa. Dimana penelitian tersebut menguji validitas terhadap 10 pernyataan kuesioner terhadap 40 responden awal mengenai evaluasi *usability website* SMA PGRI 2 Palembang menggunakan *System Usability Scale* (SUS) (Zahir & Akramunnisa, 2024).

Dalam uji anova ini, terdapat beberapa elemen penting, yaitu faktor, variabel respon, dan kelompok. Faktor dalam pengujian ini adalah *framework* yang digunakan, yaitu Express.js, Nest.js, dan Gin. Ketiga *framework* ini berfungsi sebagai variabel independen yang mempengaruhi performa *back-end*. Variabel respon adalah metrik yang diukur untuk menilai performa dari setiap *framework* (*response time, throughput, memory usage, dan CPU load*). Variabel respon ini sebagai variabel dependen yang dipengaruhi oleh *framework*. Sedangkan kelompok merupakan hasil dari pengujian metrik dari setiap *framework*. Misalnya, untuk metrik *response time*, terdapat tiga kelompok data yang masing-masing berasal dari pengukuran performa ketiga *framework*. Demikian pula, untuk metrik *throughput, memory usage, dan CPU load*, masing-masing memiliki tiga kelompok data yang terpisah untuk setiap *framework*. Hipotesis pada penelitian ini adalah sebagai berikut:

- H_0 : Tidak ada perbedaan signifikan dalam rata-rata performa antara *framework* Express.js, Nest.js, dan Gin.

- H_1 : Ada perbedaan signifikan dalam rata-rata performa antara *framework* Express.js, Nest.js, dan Gin.