



Pengujian Performansi Lima Back-End JavaScript Framework Menggunakan Metode GET dan POST

I Putu Agus Eka Pratama¹

¹Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana

¹eka.pratama@unud.ac.id

Abstract

Developers need to choosing the best Back-End JavaScript framework among the five available options (Koa, Plumier, Express, Nest, Loopback), to making development process runs well. This study conducted a performance comparison test of the five framework options, along with the recommendation levels. Testing on the HTTP protocol using the GET and POST methods which based on the ability of them to support data processing in terms of parsing, validation, routing, and requests. Two assessment indicators are used to measure performance in this research, namely: 1.)The ability of the framework to handle requests per second (req/s), where the greater the value, the better the performance, and 2.)Decrease in data processing speed (%), where the smaller the percentage, the better the performance. Tests were carried out ten times then recorded and calculated the average value. The test results showed that Koa obtained the highest average value for handling requests of 6446,278 req / s on the GET method and 4256,151 req / s on the POST method, and a percentage reduction of 0% on the GET method and the POST method. It can be concluded that Koa has the best performance and is highly recommended to be chosen by the developer.

Keywords: Back-End JavaScript, framework, GET, performance, POST.

Abstrak

Pengembang sangat perlu memperoleh masukan di dalam memilih Back-End JavaScript framework yang tepat di antara kelima pilihan yang tersedia (Koa, Plumier, Express, Nest, Loopback), agar proses pengembangan berjalan baik. Oleh karena itu, pada penelitian ini dilakukan pengujian perbandingan performansi terhadap kelima pilihan Back-End JavaScript framework, beserta dengan pemberian tingkatan rekomendasi. Pengujian dilakukan pada protokol HTTP menggunakan metode GET dan POST. Pemilihan kedua metode ini didasari kemampuan metode GET dan POST di dalam mendukung pemrosesan data pada Back-End JavaScript framework dalam hal parsing, validasi, routing, dan request. Dua buah indikator penilaian yang digunakan untuk pengukuran performansi pada penelitian ini, yaitu:1.)Kemampuan framework di dalam menangani request per detik (req/s), di mana semakin besar nilainya maka semakin baik performansinya, dan 2.)Penurunan kecepatan pemrosesan data (%), di mana semakin kecil persentasenya maka semakin baik performansinya. Pengujian dilakukan sebanyak sepuluh kali pada kelima framework, untuk kemudian dicatat dan dihitung nilai rata-rata. Hasil pengujian menunjukkan bahwa Koa memperoleh nilai rata-rata tertinggi untuk penanganan request sebesar 6446,278 req/s pada metode GET dan 4256,151 req/s pada metode POST, serta persentase penurunan sebesar 0% pada metode GET dan metode POST. Dari hasil pengujian pada penelitian ini, dapat disimpulkan bahwa Koa memiliki performansi terbaik dan sangat direkomendasikan untuk dipilih oleh pengembang.

Kata kunci: Back-End JavaScript, framework, GET, performansi, POST.

1. Pendahuluan

Teknologi Informasi (TI) berkembang semakin pesat dari waktu ke waktu. Perkembangan TI diikuti dengan perkembangan teknologi pada aplikasi dan layanan berbasis web. Salah satu teknologi tersebut adalah JavaScript. JavaScript merupakan bahasa pemrograman tingkat tinggi yang diperkenalkan pertama kali tahun 1995 untuk lingkungan pengembangan aplikasi berbasis web dinamis di sisi client, yang memungkinkan

pengembang untuk mengembangkan website dengan tampilan menarik [1]. JavaScript mengalami peningkatan kemampuan yang pesat, antara lain berupa: dukungan terhadap pemrograman berorientasi objek, pemrograman fungsional, struktural, prosedural, event-driven, prototyping, serta kemudahan pengembangan aplikasi di sisi server [2]. NodeJS menjadi contoh kemudahan pengembangan website berbasis JavaScript di sisi server berupa runtime, yang menyediakan built-in

HTTP server di dalamnya, sehingga pengembang dapat langsung menjalankan website tanpa perlu menambahkan aplikasi web server secara terpisah [3].

Peningkatan popularitas JavaScript di kalangan pengembang website, diikuti dengan tersedianya sejumlah pilihan JavaScript framework. JavaScript framework merupakan sebuah kerangka kerja siap pakai dan terstruktur pada JavaScript dalam bentuk sekumpulan library, untuk memudahkan pengembang di dalam mengembangkan website berbasis JavaScript, menuliskan fungsi, prosedur, dan fitur-fitur layanan, melalui penambahan library yang tersedia tanpa perlu menuliskan ulang sumber kode [4].

JavaScript framework terdiri atas dua jenis, yaitu: Front-End JavaScript framework dan Back-End JavaScript framework. Front-End JavaScript framework menyediakan dukungan pengembangan tampilan dan tatap muka website yang interaktif dan menarik untuk memudahkan pengguna di dalam menggunakan layanan berbasis web sekaligus meningkatkan user experience, sedangkan Back-End JavaScript framework berkaitan dengan internal sistem, dalam bentuk dukungan terhadap pemrosesan data, parsing data, suplai data, penanganan request, routing, dan validasi data [5].

Di dalam pengembangannya, Front-End JavaScript menggunakan JavaScript dan JQuery untuk website responsif [6], menggunakan Bootstrap untuk website sistem informasi [7], atau menggunakan pilihan Front-End JavaScript Framework seperti React, Angular, Vue, untuk dikombinasikan dengan HTML5 di dalam pengembangan aplikasi dan layanan E-Business berbasis website dan mobile [8]. Tidak diperlukan adanya pemilihan teknologi dan framework secara lebih seksama untuk Front-End JavaScript. Sedangkan pada Back-End JavaScript, sangat diperlukan adanya pemilihan framework secara lebih seksama. Hal ini disebabkan oleh karena Back-End JavaScript framework yang dipilih, akan mempengaruhi performansi dan kinerja dari website yang dikembangkan, dalam hal pemrosesan data hingga isu keamanan.

Terdapat lima buah Back-End JavaScript framework yang umum digunakan saat ini oleh sebagian besar pengembang, yaitu: Koa, Express, Plumier, Loopback, dan Nest, di mana kelima buah Back-End JavaScript framework ini berbasiskan NodeJS, yang menggunakan metode GET dan POST untuk pengiriman dan penerimaan data di dalam jaringan komputer [9][10][11][12][13]. Metode GET dan POST memiliki perbedaan mendasar, di mana metode GET melakukan parsing data pada URL, sedangkan metode POST melakukan parsing query string pada background.

Terkait dengan kelima Back-End JavaScript framework yang diujikan di dalam penelitian ini, sebelumnya telah terdapat penelitian yang melakukan pengujian perbandingan pada kelima framework ini dengan menggunakan empat indikator penilaian, yaitu:

1.)GitHub Stars, 2.)Issues, 3.)Updated, 4.)Created, di mana hasil penelitian tersebut menunjukkan Back-End JavaScript framework mana yang memiliki penilaian terbaik dari sisi pilihan pengguna sesuai dengan kebutuhan [14]. Namun penelitian tersebut belum menampilkan penilaian performansi dari masing-masing framework yang diujikan.

Selanjutnya, telah terdapat penelitian oleh [15] untuk menguji perbandingan performansi JavaScript dengan PHP, Python, dan NodeJS ke dalam aplikasi REST, di mana hasil penelitian tersebut menunjukkan bahwa NodeJS memiliki respon tercepat serta PHP memiliki performa terbaik. Namun penelitian tersebut belum menitikberatkan kepada performansi dari Back-End JavaScript framework berbasis NodeJS.

Penelitian lainnya juga dilakukan oleh [16] untuk menguji perbandingan performansi dari tiga Front-End JavaScript Framework (AngularJS, Aurelia, Ember) beserta dampaknya terhadap resource CPU, koneksi internet, dan bandwidth internet yang digunakan, dengan menggunakan dua belas metrik pengukuran, yaitu: size, version, data-binding, dependencies, controller, scope, service, directives, template, routing, structure, third-party addons. Hasil penelitian menunjukkan Aurelia memiliki performansi terbaik, Ember memiliki struktur terbaik, dan AngularJS memiliki update third-party terbaik. Sedangkan penelitian yang dilakukan oleh [17] juga dilakukan di sisi Front-End terhadap tiga Front-End JavaScript framework (Angular, React, Vue), namun ditinjau dari sisi performansinya terkait dengan penggunaan Application Programming Interface (API) Web. Kedua penelitian ini memberikan gambaran perbandingan untuk Front-End JavaScript framework, yang dapat menjadi acuan di dalam melakukan pengujian perbandingan performansi pada kelima Back-End JavaScript framework pada penelitian ini.

Pengujian terhadap empat JavaScript Framework (AngularJS, Ember, Knockout, Backbone) menggunakan arsitektur Model View Controller (MVC), juga telah dilakukan oleh [18] untuk mengamati kelebihan dan kekurangan dari masing-masing framework yang diujikan, beserta dengan performansi dan keamanan kode di dalamnya. Dari penelitian ini diperoleh gambaran mengenai perbandingan keempat jenis JavaScript framework modern, namun belum secara spesifik mencakup ke arah Back-End JavaScript Framework.

Pengujian perbandingan performa sejumlah JavaScript Framework juga telah dilakukan oleh [19] terhadap enam pilihan framework (Dojo, ExtJS, JQuery, MooTools, Prototype, dan YUI2), menggunakan Quality Test dengan tiga matriks pengukuran (Size Metrics, Complexity Metrics, Maintainability Metrics) untuk memperoleh penilaian kualitas, validitas, dan performa. Hasil yang diperoleh adalah MooTools memperoleh hasil terbaik untuk ketiga penilaian tersebut

dibandingkan framework lainnya. Hasil penelitian ini mengkhususkan kepada JavaScript framework berorientasi objek, namun belum mencakup typescript.

Penelitian lainnya dilakukan oleh [20] terhadap tujuh versi dari JQuery (jQuery 1.4.1, 1.5.1., 1.6.1., 1.6.2, 1.7.0, 1.8.0, 1.9.0) menggunakan pengujian performansi dan pengujian validitas. Hasil pengujian pada penelitian ini menunjukkan bahwa versi terbaru JQuery (1.9.0) memiliki nilai terbaik dari sisi validasi, performansi, dan dukungan komunitas. Hasil pengujian ini menjadi bahan pertimbangan di dalam pemilihan versi NodeJS dan framework yang diujikan pada penelitian ini.

Terakhir, telah dilakukan penelitian mengenai performa JavaScript Framework dari sisi web browser oleh [21] menggunakan SAPUI5 dan JQuery berbasis HTML5, CSS3, dan JavaScript Libraries. Meskipun penelitian ini mengkhususkan kepada Single Page Application (SPA) JavaScript Framework serta belum mencakup perbandingan performansi untuk Back-End JavaScript Framework, namun memberikan kontribusi berupa gambaran performansi dari JavaScript framework di sisi client melalui web browser, terutama terkait dengan metode GET dan POST yang berelasi dengan parsing URL dan parsing query string.

Berangkat dari sejumlah tinjauan pustaka dan penelitian-penelitian sebelumnya beserta dengan studi kasus permasalahan pemilihan Back-End JavaScript framework yang tepat bagi pengembang di antara kelima pilihan yang tersedia (Koa, Express, Plumier, Loopback, Nest), menjadikan perlunya dilakukan penelitian dan pengujian mengenai perbandingan performansi kelima Back-End JavaScript framework tersebut. Metode pengujian yang dipilih untuk digunakan adalah metode GET (di URL) dan POST (di background) pada protokol HTTP di dalam jaringan komputer (internet), yang di dalamnya meliputi empat proses, yaitu: routing, parsing, validasi, dan request. Kedua metode ini umum digunakan di sisi server, terutamanya pada framework berbasis NodeJS, yang dalam hal ini meliputi kelima framework yang diujikan.

Selain itu, kelima framework yang diujikan, juga merupakan typescript framework yang menggunakan NodeJS Package Manager (NPM) untuk kebutuhan semua library dan paket yang dibutuhkan secara online. Kelima framework ini, juga dilengkapi dengan Router untuk merutekan URL dan request, Body Parser untuk melakukan parsing nilai dari form HTML, serta Validator untuk memvalidasi nilai. Kelima framework yang diujikan, memiliki basis framework masing-masing atau menjadi basis bagi framework lainnya, sebagaimana disajikan pada Tabel 1:

Tabel 1. Basis Framework, Router, Body Parser, dan Validator dari Masing-Masing Framework

Back-End JavaScript Framework	Basis Framework	Router	Body Parser	Validator
Koa	-	Koa-router	Koa-parser	joi
Plumier	Koa	built-in	built-in	built-in
Express	-	built-in	body-parser	joi
Nest	Express	built-in	built-in	built-in
Loopback	Express	built-in	built-in	built-in

Back-End JavaScript Framework	Basis Framework	Router	Body Parser	Validator
Koa	-	Koa-router	Koa-parser	joi
Plumier	Koa	built-in	built-in	built-in
Express	-	built-in	body-parser	joi
Nest	Express	built-in	built-in	built-in
Loopback	Express	built-in	built-in	built-in

Tabel 1 menunjukkan bahwa Koa dan Express menjadi basis bagi ketiga framework lainnya (Plumier, Nest, Loopback). Koa menjadi basis bagi Plumier, sedangkan Express menjadi basis bagi Nest dan Loopback. Umumnya, beberapa karakteristik dari framework yang menjadi basis akan dimiliki juga oleh framework turunannya.

Di dalam penelitian ini, terdapat dua poin yang menjadi rumusan masalah dalam bentuk pertanyaan penelitian, yaitu: 1.) Bagaimana cara melakukan pengujian perbandingan performansi dari kelima Back-End JavaScript framework yang diujikan? 2.) Framework mana yang memiliki nilai performansi terbaik untuk dapat direkomendasikan kepada para pengembang?

Tujuan penelitian ini adalah memperoleh perbandingan hasil penilaian performansi dari kelima pilihan Back-End JavaScript framework dari sisi kemampuan menangani request per detik serta penurunan kecepatan terkait dengan parsing, untuk dapat menjadi masukan, acuan, dan pertimbangan bagi para pengembang di dalam memilih framework yang tepat.

2. Metode Penelitian

Untuk mendukung jalannya implementasi dan pengujian pada penelitian ini, digunakan sejumlah hardware dan software pendukung. Hardware yang digunakan adalah notebook Dell Latitude E6440 (Intel core i7, 16GB RAM, harddisk 500GB), modem internet, mouse. Software yang digunakan adalah sistem operasi Linux Ubuntu 20.04 64 bit, NodeJS 10.19.0, dan NodeJS Package Manager (NPM). Penelitian dilakukan pada kurun waktu awal bulan September 2020 hingga pertengahan bulan November 2020, bertempat di rumah pribadi peneliti selama masa pandemi Covid19.

Mengingat bahwa Back-End JavaScript framework bekerja di sisi server pada protokol HTTP, maka penelitian ini menggunakan metode pengujian berupa metode GET dan metode POST. Metode GET memiliki konsep parsing data pada URL di browser, sedangkan metode POST memiliki konsep parsing query string pada background.

Pengujian dilakukan sebanyak sepuluh kali untuk kelima framework yang diujikan, dengan menggunakan metode

GET dan POST. Adapun pertimbangan pengujian dilakukan sebanyak sepuluh kali adalah untuk memperoleh akurasi nilai rata-rata yang lebih baik, serta memudahkan di dalam analisa hasil pengujian.

Terdapat dua buah indikator penilaian performansi framework yang diujikan, yaitu: 1.)Kemampuan masing-masing framework di dalam menangani request per detik atau request/second (req/s), di mana semakin besar nilainya maka semakin baik performansi dari framework yang diujikan, dan 2.)Persentase penurunan kecepatan pemrosesan data (validasi, parsing, routing, request) dari masing-masing framework (%), di mana semakin kecil persentasenya maka semakin baik performansi dari framework yang diujikan.

Data-data hasil pengujian dicatatkan ke dalam tabel, untuk kemudian dihitung nilai rata-rata. Nilai rata-rata menjadi tolok ukur penilaian performansi dari setiap framework.

Untuk mendukung jalannya penelitian, digunakan repositori online NodeJS Package Manager (NPM). NPM membantu proses instalasi kelima framework berbasis NodeJS tersebut, melalui library yang tersedia.

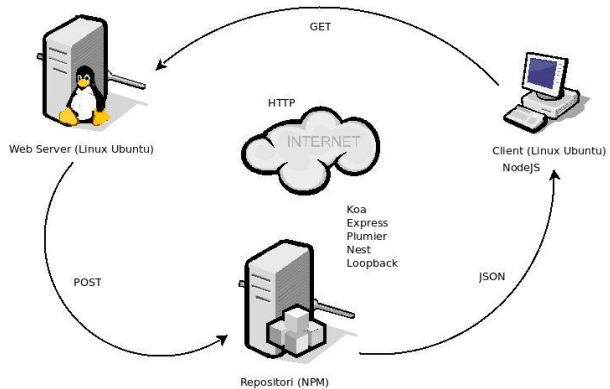
Untuk konfigurasi dan struktur format pertukaran data, penyimpanan data, versi, dependencies, framework, dan router dari masing-masing framework, disimpan pada file JavaScript Object Notation (JSON) bernama package.json. Isi dari file package.json sebagai berikut:

```

package.json
{
  "name": "framework-cost-benchmark",
  "version": "1.0.0",
  "scripts": {
    "start": "ts-node .",
    "start:server": "ts-node"
  },
  "dependencies": {
    "@loopback/boot": "^1.4.4",
    "@loopback/context": "^1.20.2",
    "@loopback/core": "^1.8.5",
    "@loopback/rest": "^1.16.3",
    "@nestjs/common": "^6.0.0",
    "@nestjs/core": "^6.0.0",
    "@nestjs/platform-express": "^6.0.0",
    "@types/axios": "^0.14.0",
    "@types/joi": "^14.3.3",
    "@types/koa-router": "^7.0.42",
    "@types/numeral": "^0.0.25",
    "axios": "^0.19.0",
    "class-transformer": "^0.2.3",
    "class-validator": "^0.9.1",
    "joi": "^14.3.1",
    "koa-router": "^7.4.0",
    "numeral": "^2.0.6",
    "plumier": "1.0.0-beta.9",
    "reflect-metadata": "^0.1.13",
    "rxjs": "^6.5.2"
  },
  "devDependencies": {
    "autocannon": "^4.0.0",
    "ts-node": "8.3.0",
    "typescript": "3.5.3"
  }
}
    
```

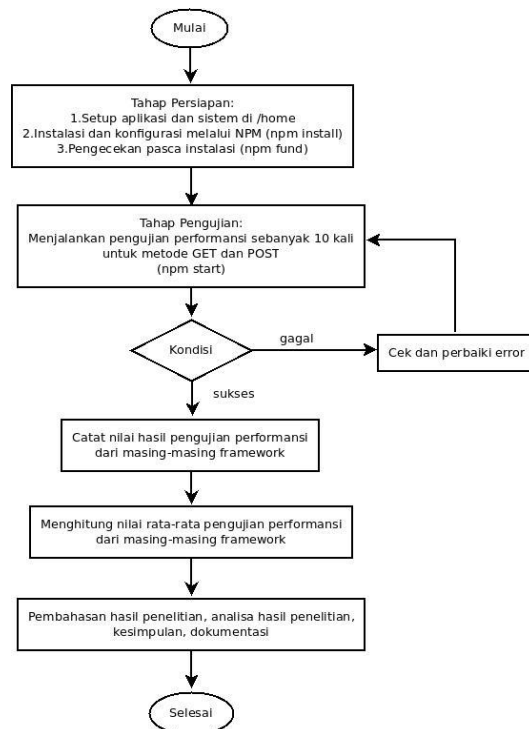
Langkah-langkah pengujian dilakukan berdasarkan kepada desain skenario pengujian dan flowchart

penelitian. Desain skenario pengujian berdasarkan penjelasan sebelumnya ditunjukkan pada Gambar 1.



Gambar 1. Desain Skenario Pengujian Perbandingan Performansi Back-End JavaScript Framework

Gambar 1 menunjukkan desain skenario pengujian, di mana terdapat komputer dengan sistem operasi Linux Ubuntu, yang bertindak sebagai web server dan client. Pada komputer server diinstall kelima framework dengan menggunakan repositori online dari NPM, sedangkan struktur dan konfigurasi disimpan ke dalam bentuk file JSON. Pengujian dilakukan menggunakan metode GET dan POST melalui web browser di komputer client pada protokol HTTP sebanyak sepuluh kali terhadap kelima framework untuk melakukan pemrosesan data (request, validasi, parsing, routing), menggunakan perintah npm start pada Terminal Linux. Hasil pengujian tersebut kemudian dicatat, dibahas, dan dianalisa.



Gambar 2. Flowchart Penelitian

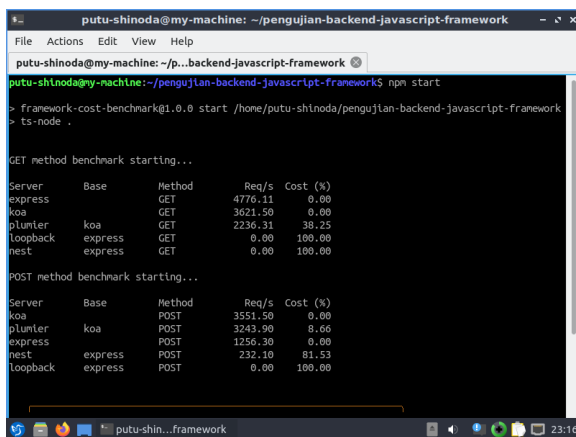
Untuk urutan langkah penelitian, disesuaikan dengan desain skenario, yang ditampilkan ke dalam bentuk flowchart penelitian. Gambar 2 menampilkan flowchart penelitian.

Gambar 2 menampilkan tahap persiapan dan tahap pengujian. Pada tahap persiapan, dilakukan instalasi NodeJS dan NPM, instalasi kelima framework yang diujikan beserta library terkait menggunakan perintah npm install, serta pengecekan melalui npm fund.

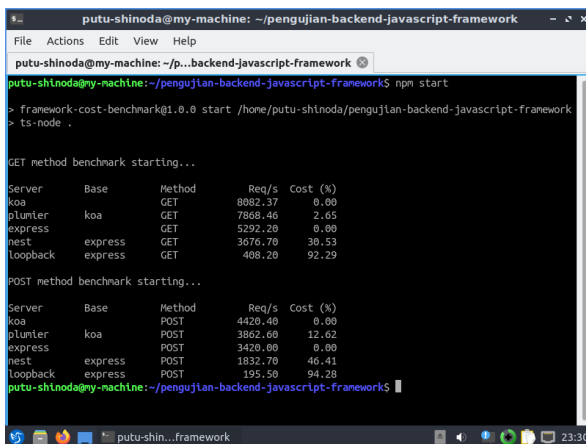
Pada tahap pengujian, dilakukan sepuluh kali pengujian, dengan menggunakan metode GET dan POST melalui perintah npm start. Apabila terdapat kegagalan, dilakukan pengecekan dan perbaikan error, kemudian mengulangi perintah npm start. Apabila pengujian sukses, maka hasil pengujian dicatat dan dihitung nilai rata-rata. Setelah tahap pengujian, dilakukan pembahasan hasil penelitian, analisa hasil penelitian, kesimpulan, dan dokumentasi dalam bentuk publikasi paper.

3. Hasil dan Pembahasan

Pengujian dilakukan sebanyak sepuluh kali pada metode GET dan metode POST. Tampilan kesepuluh buah pengujian yang telah dilakukan, ditunjukkan pada Gambar 3 hingga Gambar 12.



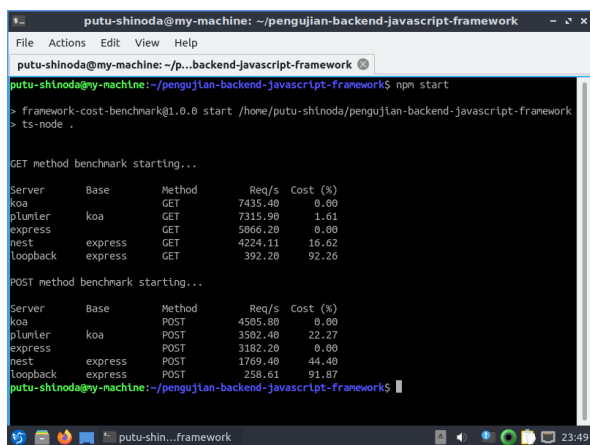
Gambar 3. Pengujian Pertama



Gambar 4. Pengujian Kedua

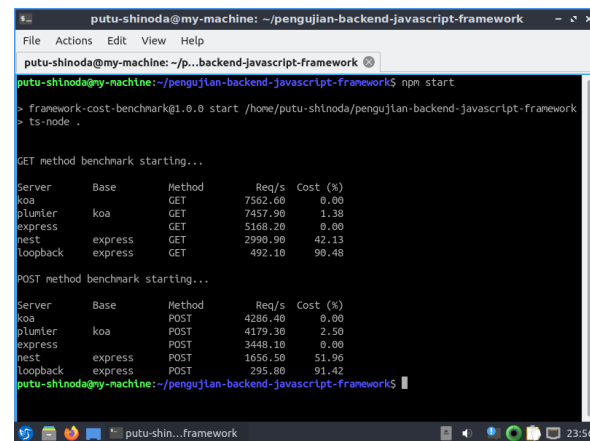
Gambar 3 menampilkan pengujian pertama. Pada metode GET, Express memiliki kemampuan penanganan request tertinggi (4776,11 req/s) serta Express dan Koa tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (3551,50 req/s) serta Koa, Plumier, dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).

Gambar 4 menampilkan pengujian kedua. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (8082,37 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4420,40 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).



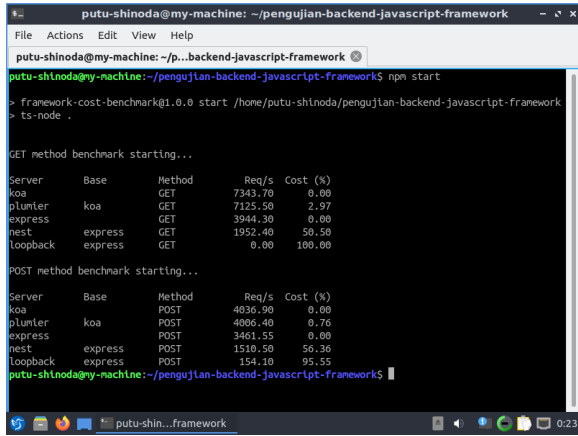
Gambar 5. Pengujian Ketiga

Gambar 5 menampilkan pengujian ketiga. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (7435,40 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4505,80 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).



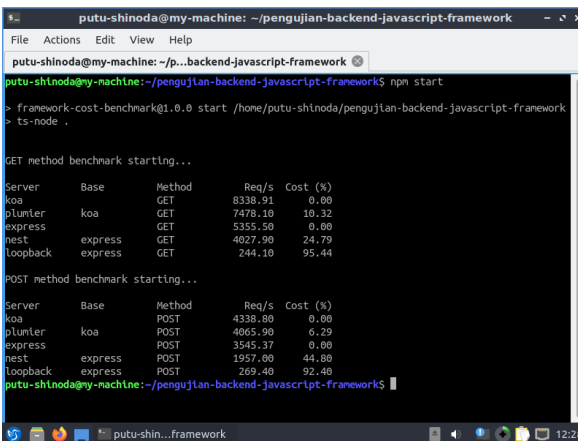
Gambar 6. Pengujian Keempat

Gambar 6 menampilkan pengujian keempat. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (7562,60 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4286,40 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).



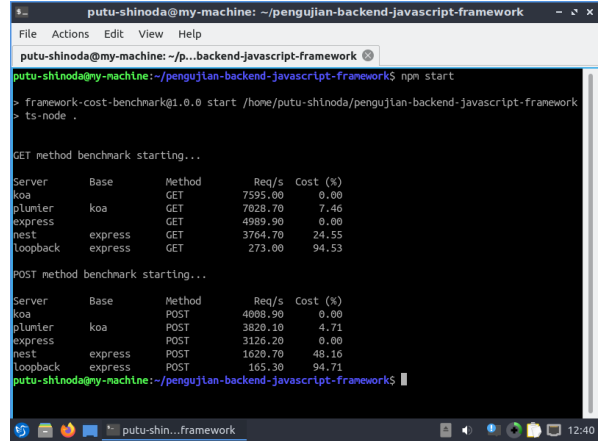
Gambar 7. Pengujian Kelima

Gambar 7 menampilkan pengujian kelima. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (7343,70 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4036,90 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).



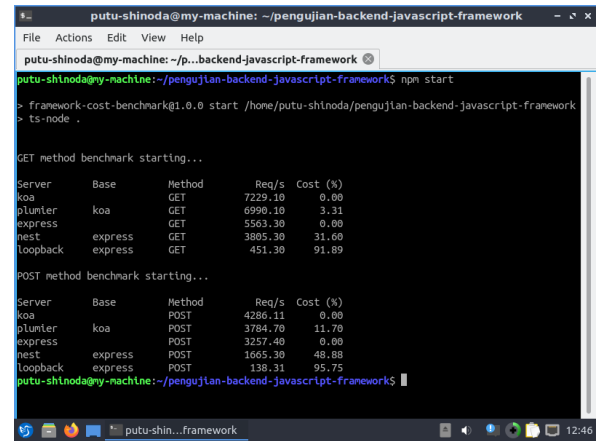
Gambar 8. Pengujian Keenam

Gambar 8 menampilkan pengujian keenam. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (8338,91 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4338,80 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).

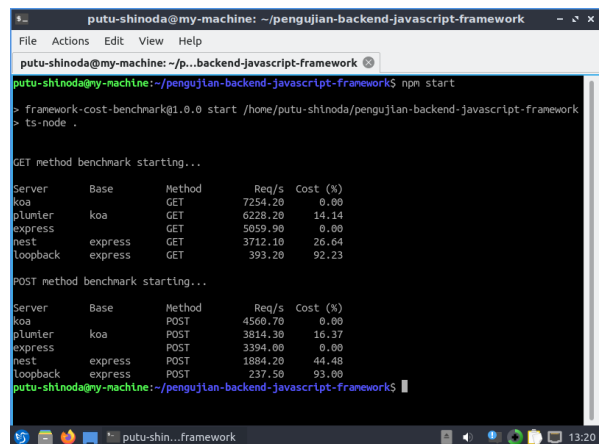


Gambar 9. Pengujian Ketujuh

Gambar 9 menampilkan pengujian ketujuh. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (7595,00 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4008,90 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).



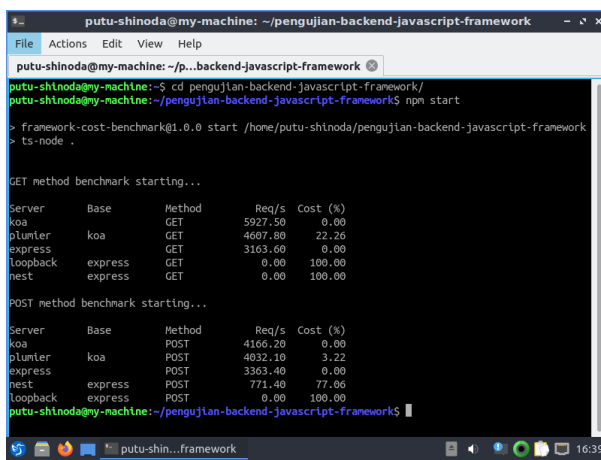
Gambar 10. Pengujian Kedelapan



Gambar 11. Pengujian Kesembilan

Gambar 10 menampilkan pengujian kedelapan. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (7229,10 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4786,11 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).

Gambar 11 menampilkan pengujian kesembilan. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (7254,20 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4560,70 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).



Gambar 12. Pengujian Kesepuluh

Gambar 12 menampilkan pengujian kesepuluh. Pada metode GET, Koa memiliki kemampuan penanganan request tertinggi (5927,50 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %). Pada metode POST, Koa memiliki kemampuan penanganan request tertinggi (4166,20 req/s) serta Koa dan Express tidak mengalami penurunan kecepatan pemrosesan data (persentase 0,00 %).

Nilai yang ditampilkan dari masing-masing pengujian, diinputkan ke dalam empat tabel berbeda (Tabel 2, Tabel 3, Tabel 4, Tabel 5), untuk kemudian dihitung nilai rata-rata, sehingga memudahkan di dalam melakukan analisa hasil pengujian.

Dari sepuluh kali pengujian kemampuan penanganan request per second dengan metode GET dan POST, nilai dan nilai rata-rata setiap framework, ditampilkan pada Tabel 2 dan Tabel 3 berikut:

Tabel 2. Hasil Pengujian Kemampuan Penanganan Request/second Dengan Metode GET (req/s)

Pengujian ke	Koa	Plumier	Expres	Nest	Loopback
1	3621,5	2236,21	4776,1	0,00	0,00

2	8082,3	7868,46	5292,2	3676,7	408,20
3	7435,4	7315,90	5066,2	4224,1	392,20
4	7562,6	7457,90	5168,2	2990,9	492,10
5	7343,7	7125,50	3944,3	1952,4	0,00
6	8338,9	7478,10	5355,5	4027,9	244,10
7	7595,0	7028,70	4989,9	3764,7	273,00
8	7229,1	6990,10	5563,3	3805,3	451,30
9	7254,2	6228,20	5059,9	3712,1	393,20
10	5927,5	4607,80	3163,6	0,00	0,00
Rata-Rata	6446,2	6433,68	4837,9	2815,4	265,41
	78	7	21	11	

Tabel 3. Hasil Pengujian Kemampuan Penanganan Request/second Dengan Metode POST (req/s)

Pengujian ke	Koa	Plumier	Expres	Nest	Loopback
1	3551,5	3243,9	1256,3	232,10	0,00
2	4420,2	3862,6	3420,0	1832,7	195,50
3	4505,8	3502,4	3182,2	1769,4	258,61
4	4286,4	4179,3	3448,1	1656,5	295,80
5	4036,9	4006,4	3461,5	1510,5	154,10
6	4338,8	4065,9	3545,3	1957,0	269,40
7	4008,9	3820,1	3126,2	1620,7	165,30
8	4286,1	3784,7	3257,4	1665,3	138,31
9	4960,7	3814,3	3394,0	1884,2	237,50
10	4166,2	4032,1	3363,4	771,40	0,00
Rata-Rata	4256,1	3831,1	3145,5	1489,9	171,452
	51	7	2	8	

Tabel 2 dan Tabel 3 memperlihatkan bahwa Koa memiliki nilai rata-rata tertinggi dibandingkan framework lainnya yang diujikan pada pengujian kemampuan penanganan request per second dengan metode GET dan POST.

Dari sepuluh kali pengujian penurunan kecepatan pemrosesan data (parsing, validasi, routing, request) dengan metode GET dan POST, nilai dan nilai rata-rata setiap framework ditampilkan pada Tabel 4 dan Tabel 5 berikut:

Tabel 4. Hasil Pengujian Penurunan Kecepatan Pemrosesan Data Dengan Metode GET (%)

Pengujian ke	Koa	Plumier	Expres	Nest	Loopback
--------------	-----	---------	--------	------	----------

ujian ke	r	s	k		
1	0,00	38,25	0,00	100,00	100,00
2	0,00	2,65	0,00	30,53	92,29
3	0,00	1,61	0,00	16,62	92,26
4	0,00	1,38	0,00	42,13	90,48
5	0,00	2,97	0,00	50,50	100,00
6	0,00	10,32	0,00	24,79	95,44
7	0,00	7,46	0,00	24,55	94,53
8	0,00	3,31	0,00	31,60	91,89
9	0,00	14,14	0,00	26,64	92,23
10	0,00	22,26	0,00	100,00	100,00
Rata - Rata	0,00	10,435	0,00	44,736	94,913

Tabel 5. Hasil Pengujian Penurunan Kecepatan Pemrosesan Data Dengan Metode POST (%)

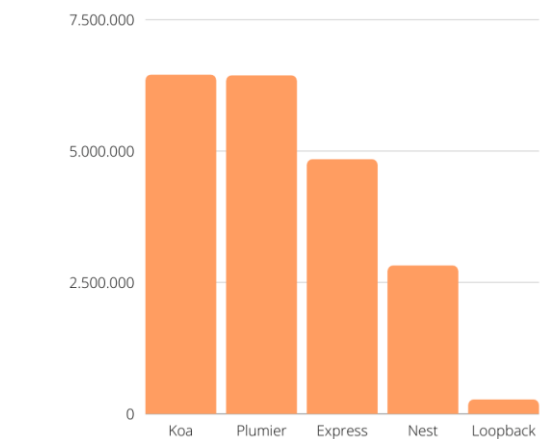
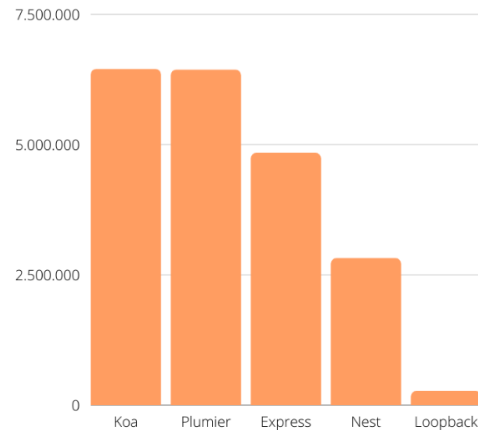
Pengujian ke	Koa	Plumier	Express	Nest	Loopback
1	0,00	8,66	0,00	81,53	100,00
2	0,00	12,62	0,00	46,41	94,28
3	0,00	22,27	0,00	44,40	91,87
4	0,00	2,50	0,00	51,96	91,42
5	0,00	0,76	0,00	56,36	95,55
6	0,00	6,29	0,00	44,80	92,40
7	0,00	4,71	0,00	48,16	94,71
8	0,00	11,70	0,00	48,88	95,75
9	0,00	16,37	0,00	44,48	93,00
10	0,00	3,22	0,00	77,06	100,00
Rata - Rata	0,00	13,305	0,00	54,304	84,898

Tabel 4 dan Tabel 5 memperlihatkan bahwa Koa memiliki persentase nilai penurunan yang konstan sebesar 0,00%, yang menandakan bahwa sama sekali tidak terjadi penurunan kecepatan pemrosesan data (parsing, validasi, routing, request) pada metode GET dan POST.

3.3 Pembahasan Hasil Pengujian

Berdasarkan kepada data-data hasil pengujian, Koa memiliki performansi terbaik untuk penanganan request per second dan pengujian penurunan kecepatan pemrosesan data, pada metode GET dan metode POST. Express dan Plumier memiliki performansi baik, meskipun pernah mengalami penurunan kecepatan pemrosesan data pada salah satu pengujian. Sedangkan Nest memiliki performansi tidak baik, karena beberapa kali mengalami penurunan kecepatan pemrosesan data. Loopback memiliki performansi terburuk karena paling sering mengalami penurunan performansi secara drastis.

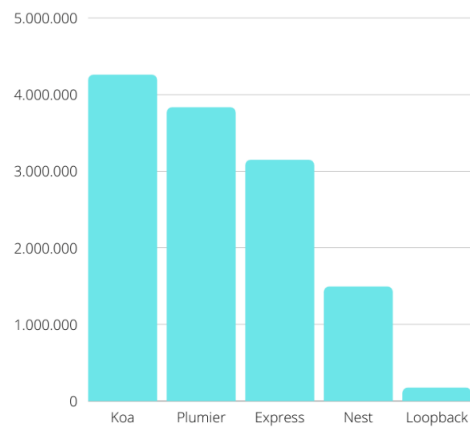
Data-data hasil pengujian, dapat divisualisasikan ke dalam bentuk diagram batang. Diagram batang untuk hasil pengujian pada Tabel 2, disajikan pada Gambar 13.



Gambar 13. Diagram Batang Hasil Pengujian Penanganan Request Per Detik Menggunakan Metode GET

Gambar 13 menampilkan diagram batang untuk hasil pengujian pada Tabel 2, di mana Koa memiliki performansi terbaik, diikuti dengan Plumier dan Express. Loopback terlihat paling rendah yang menunjukkan bahwa framework ini memiliki performansi terburuk.

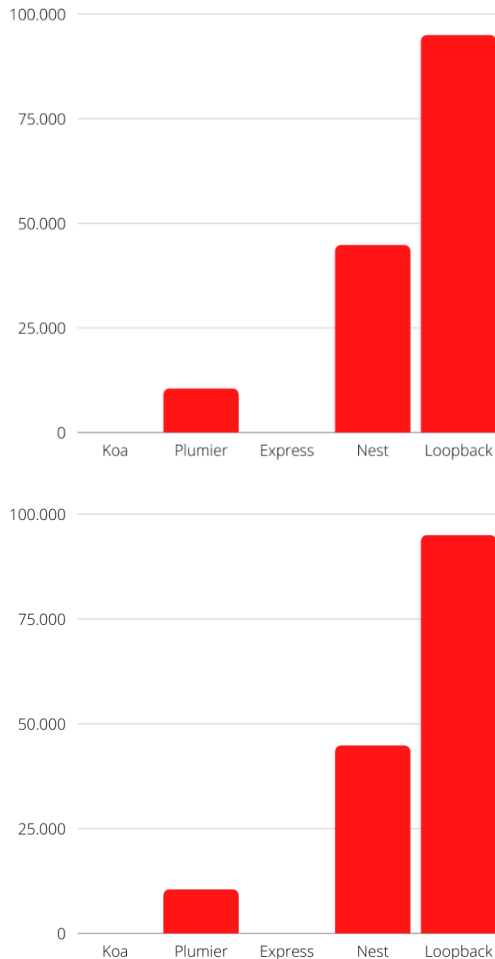
Diagram batang untuk hasil pengujian pada Tabel 3, disajikan pada Gambar 14.



Gambar 14. Diagram Batang Hasil Pengujian Penanganan Request Per Detik Menggunakan Metode POST

Gambar 14 menampilkan visual berupa diagram batang untuk hasil pengujian pada Tabel 3, di mana Koa memiliki performansi terbaik, diikuti dengan Plumier dan Express. Loopback kembali terlihat paling mencolok dengan performansi terburuk.

Diagram batang untuk hasil pengujian pada Tabel 4, disajikan pada Gambar 15.



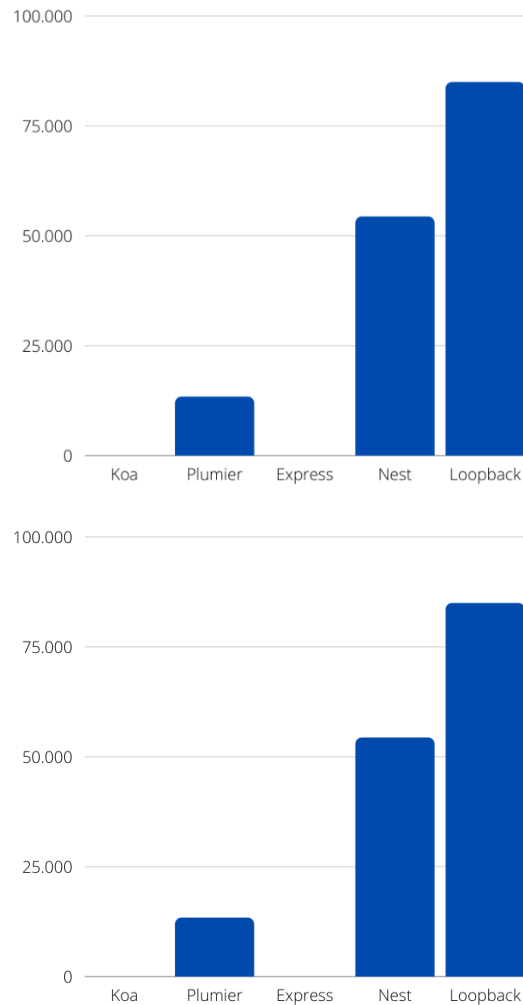
Gambar 15. Diagram Batang Hasil Pengujian Penurunan Kecepatan Pemrosesan Data Menggunakan Metode GET

Gambar 15 menampilkan diagram batang untuk hasil pengujian pada Tabel 4, di mana Koa memiliki performansi terbaik dengan persentase terendah, diikuti dengan Express dan Plumier. Loopback terlihat paling mencolok dengan persentase tertinggi, yang menunjukkan bahwa framework ini memiliki performansi terburuk.

Diagram batang untuk hasil pengujian pada Tabel 5, disajikan dalam bentuk diagram batang pada Gambar 16.

Gambar 16 menampilkan diagram batang untuk hasil pengujian pada Tabel 5, di mana Koa memiliki performansi terbaik dengan persentase penurunan terkecil, diikuti dengan Express dan Plumier. Loopback terlihat paling mencolok dengan persentase penurunan

tertinggi, yang menunjukkan bahwa framework ini memiliki performansi terburuk.



Gambar 16. Diagram Batang Hasil Pengujian Penurunan Kecepatan Pemrosesan Data Menggunakan Metode POST

3.4 Pemberian Tingkatan Rekomendasi

Berdasarkan kepada hasil pengujian, pembahasan, dan analisa, dilakukan pemberian tingkatan rekomendasi terhadap kelima framework yang diujikan. Tingkatan rekomendasi yang diberikan meliputi empat tingkatan, yaitu: 1.)Sangat Direkomendasikan, 2.)Direkomendasikan, 3.)Tidak Direkomendasikan, 4.)Sangat Tidak Direkomendasikan. Pengembang disarankan untuk memilih framework dengan tingkatan rekomendasi Sangat Direkomendasikan atau Direkomendasikan. Tabel 6 menunjukkan tingkatan rekomendasi dari kelima framework yang diujikan:

Tabel 6. Tingkatan Rekomendasi Back-End JavaScript Framework

Framework	Penilaian	Rekomendasi
Koa	Sangat Baik	Sangat Direkomendasikan
Plumier	Baik	Direkomendasikan
Express	Baik	Direkomendasikan
Nest	Buruk	Tidak Direkomendasikan

Loopback	Sangat Buruk	Sangat Tidak Direkomendasikan
----------	--------------	-------------------------------

Berdasarkan Tabel 6, pengembang sangat disarankan untuk memilih Koa. Pengembang juga disarankan untuk memilih Plumier dan Express. Pengembang tidak disarankan untuk memilih Nest dan sangat tidak disarankan untuk memilih Loopback.

4. Kesimpulan

Kesimpulan yang dapat diperoleh dari hasil penelitian ini adalah Koa memiliki performansi terbaik dan sangat direkomendasikan untuk dipilih oleh pengembang. Pemilihan Back-End JavaScript framework yang tepat, akan membantu proses pengembangan software menjadi lebih baik.

Ucapan Terimakasih

Terima kasih sebesar-besarnya kepada komunitas Facebook Developer Cycle Bali, Ketut Sandiarsa (pengembang Plumier framework dan Full-Stack benchmark tool), komunitas Linux Indonesia, NodeJS, Universitas Udayana, serta keluarga penulis.

Daftar Rujukan

- [1] W3Schools, 2020. *JavaScript Introduction*. [Online]. Tersedia di: https://www.w3schools.com/js/js_intro.asp [Accessed 12 November 2020].
- [2] Medium, 2020. *Top JavaScript Frameworks and Topics to Learn in 2020 and the New Decade*. [Online]. Tersedia di: <https://medium.com/JavaScript-scene/top-JavaScript-frameworks-and-topics-to-learn-in-2020-and-the-new-decade-ced6e9d812f9> [Accessed 12 November 2020].
- [3] Bangare, S. L., et al., 2016. Using Node.Js to Build High Speed and Scalable Backend Database Server . *International Journal of Research in Advent Technology, Special Issue*, pp.61-64.
- [4] Pano, A., et al., 2016. Rationale Leading to the Adoption of a JavaScript Framework. *ArXiv Software Engineering Journal*, pp.1-49.
- [5] Putra, P.A., 2017. Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode BackEnd dan FrontEnd. *Jurnal Ilmiah Teknologi Informasi (JUTI)*, Vol.10 No.1, pp.39-48.
- [6] Jon, D., 2014. *JavaScript and JQuery: Interactive Front-End Web Development. 1st edition*. New Jersey: Wiley Publications.
- [7] Faried E., dan Barry N., 2016. Penerapan Framework Bootstrap Dalam Pembangunan Sistem Informasi Pengangkutan dan Penjadwalan Pegawai (Studi Kasus: Rumah Sakit Bersalin Buah Delima Sidoarjo). *Jurnal Informatika Mulawarman*, Vol. 11 No. 1, pp.9-13.
- [8] Yongkang X., Jia P.H., dan Yong, Y.L., 2019. Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development. In: *ICCAE 2019: Proceedings of the 2019 11th International Conference on Computer and Automation Engineering*. Perth, February 2019.
- [9] Koa, 2020. *Koa: Next Generation Web Framework for NodeJS*. [Online]. Tersedia di: <https://koajs.com> [Accessed 14 November 2020].
- [10] OpenJS foundation, 2020. *Express: Fast, Unopinionated, Minimalist Web Framework for NodeJS*. [Online]. Tersedia di: <https://expressjs.com> [Accessed 14 November 2020].
- [11] Plumier, 2020. *Plumier In Five Minutes*. [Online]. Tersedia di: <https://plumierjs.com/docs/overview> [Accessed 14 November 2020].
- [12] Nest, 2020. *Hello, Nest! A Progressive NodeJS Framework for Building Efficient, Reliable and Scalable Server-Side*

Applications. [Online]. Tersedia di: <https://nestjs.com> [Accessed 14 November 2020].

- [13] StrongLoop, 2020. *LoopBack: A Highly Extensible NodeJS and TypeScript Framework for Building APIs and Microservices*. [Online]. Tersedia di: <https://loopback.io> [Accessed 14 November 2020].
- [14] John P., 2020. *NPM Trends: Koa vs Nest vs Loopback vs Express vs Plumier*. [Online]. Tersedia di: <https://www.npmtrends.com/koa-vs-nest-vs-loopback-vs-express-vs-plumier> [Accessed 14 November 2020].
- [15] Anugerah C.R., Rizal F.A., 2018. Performance Comparison of Node.js, PHP, and Python Performance. *Cogito Smart Journal*, 4(1), pp.171-187.
- [16] Jefferson, F., 2018. *A JavaScript Framework Comparison Based on Benchmarking Software Metrics and Environment Configuration Software Metrics and Environment Configuration*. Masters dissertation. Dublin: Technological University (TU) Dublin.
- [17] Samarth M., dan Rashmi R., 2020. Comparison of JavaScript Frontend Frameworks and Web API Services. *International Research Journal of Engineering and Technology (IRJET)*, Vol.07, Issue.06, pp.108-112.
- [18] Sanja D., dan Drazeon D., 2018. Modern JavaScript frameworks: A Survey Study. In: *Zooming Innovation in Consumer Technologies Conference (ZINC) 2018*, Novi Sad, Serbia 30-31 May 2018. Serbia.
- [19] Andreas G., Sotiris C., Theodore P., 2012. Comparative Evaluation of JavaScript Frameworks. In: *WWW '12 Companion: Proceedings of the 21st International Conference on World Wide Web*, Lyon, France April 2012. France.
- [20] Andreas G., Sotiris C., Theodore P., 2014. Performance and Quality of JQuery JavaScript Framework. *Information Engineering (IE) Journal*, Vol.3, pp.1-11.
- [21] Jaime R., et al., 2016. Browser Performance of JavaScript Framework, SAPUI5 and jQuery. In: *IEEE United Nations Department of Economic and Social Affairs, 2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, USA 15-17 December 2012. United Nations: Las Vegas.