



## Seminar Nasional Ilmu Teknik dan Aplikasi Industri (SINTA)

Homepage: [sinta.eng.unila.ac.id](http://sinta.eng.unila.ac.id)



### Pengembangan REST API dengan Express js, ASP.NET CORE dan DJANGO: Studi Kasus Perbandingan Performa Dengan Beban Yang Beragam

Mardiana <sup>a,\*</sup>, Y. Mulyani <sup>b</sup>, D. Despa <sup>c</sup>, W. E. Sulistiono <sup>d</sup>, R. Yulianto <sup>e</sup>

<sup>a,b,c,d,e</sup> Teknik Informatika, Program Profesi Insinyur, Universitas Lampung, Jalan Prof. Dr Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Kota Bandar Lampung, Lampung 35141

#### INFORMASI ARTIKEL

#### ABSTRAK

##### Riwayat artikel:

Diterima 11/11/2024

Direvisi 13/01/2025

Dipublish 22/05/2025

##### Kata kunci:

Web Service

Express JS

ASP .NET Core

Django

Performa

Studi Kasus

Skenario Penggunaan

Dalam pengembangan aplikasi web service, pemilihan framework yang tepat dapat berdampak signifikan pada performa dan kualitas aplikasi yang dihasilkan. Express Js, ASP.NET CORE dan Django merupakan framework yang populer digunakan dalam pengembangan aplikasi web service. Namun, belum ada penelitian yang secara komprehensif membandingkan performa ketiga *framework* ini dalam berbagai skenario penggunaan beban dengan metode *Create, Read Update, dan Delete* (CRUD). Penelitian ini bertujuan untuk membandingkan performa ketiga dalam pengembangan aplikasi web service pada skenario penggunaan CRUD. Penelitian ini akan menggunakan pendekatan studi kasus dengan mengimplementasikan aplikasi web service menggunakan Express JS, ASP.NET CORE dan Django. Pengujian performa akan dilakukan dengan mengukur waktu respon, *throughput*, dan *resource utilization* dari ketiga framework. Hasil dari penelitian ini diharapkan dapat memberikan pemahaman yang mendalam tentang performa Express Js, ASP.NET CORE dan Django dalam pengembangan aplikasi *web service*, serta memberikan panduan yang lebih baik bagi pengembang dalam memilih *framework* yang sesuai dengan kebutuhan proyek mereka. Dari pengujian didapatkan bahwa ketiga *framework* menyelesaikan pengujian dengan tingkat *error* 0%. *framework* AS .NET Corecore memiliki keunggulan signifikan dalam metode *GET* dengan rata-rata 73ms, nilai minimal 51ms, nilai maksimal 349ms *throughput* 13,66 request tiap detik dan juga sebagai *framework* yang paling berat dengan 41.6% penggunaan prosesor dan 224,MB RAM

#### 1. Pendahuluan

Web service adalah sebuah aplikasi perangkat lunak yang digunakan untuk para pengguna agar dapat melakukan komunikasi dengan perangkat keras sebagai sarana penyimpanan dan pengolahan informasi yang dibutuhkan. Dewasa ini web service sudah banyak digunakan untuk memfasilitasi berbagai kepentingan user yang diantara lainnya adalah untuk kebutuhan

edukasi, komersial, dan pemerintahan (Dhalla, 2021). Dengan kebutuhan tersebut sebuah web service perlu dibangun dengan cepat dan hal ini dapat dilakukan dengan menggunakan sebuah framework.

Dalam pengembangan aplikasi web service, pemilihan framework yang tepat dapat berdampak signifikan pada performa dan kualitas aplikasi yang dihasilkan dan juga dapat mempercepat dan meringkas biaya pengembangan (Karlsson dan Larsson-Green,

\* Penulis korespondensi.

E-mail: [mardiana@eng.unila.ac.id](mailto:mardiana@eng.unila.ac.id) (Mardiana)

n.d.). Express JS, ASP.NET CORE dan Django merupakan beberapa framework yang populer digunakan dalam pengembangan aplikasi web service. Express JS, sebagai framework untuk bahasa pemrograman Javascript, dikenal dengan kemudahannya dalam pengembangan aplikasi web service yang skalabel dan efisien. ASP.NET CORE adalah framework yang menggunakan Bahasa C# yang didukung oleh perusahaan Microsoft, dan Django, sebagai framework untuk bahasa pemrograman Python, menawarkan fitur yang ringan dengan komponen esensial untuk melakukan pengembangan aplikasi web service.

Namun, meskipun telah banyak digunakan, belum ada penelitian yang secara komprehensif membandingkan performa beberapa framework ini dalam berbagai skenario penggunaan yang beragam. Oleh karena itu, penelitian ini akan fokus pada studi kasus perbandingan performa antara Express Js, ASP.NET CORE dan Django dalam pengembangan aplikasi web service pada skenario penggunaan yaitu dengan operasi Create untuk menyimpan data, Update untuk mengedit data, Read untuk mengambil data dan Delete untuk menghapus data (CRUD).

Dalam penerapan CRUD digunakan arsitektur komunikasi yang bernama Representational State Transfer Application Interface (REST API) adalah sebuah cara bagaimana sebuah program klien dapat melakukan komunikasi dengan web service dimana REST adalah sebuah gaya arsitektur yang digunakan secara umum untuk web service modern dan API yang akan melakukan ekspose dari data dan fungsi yang dibutuhkan oleh program klien.(Ehsan et al., 2022)

Menurut buku REST API Design RuleBook API (Application Programming Interface) adalah sebuah cara dari client untuk melakukan komunikasi dengan sebuah Web Service. API mengekspose kumpulan data data dan fungsi untuk memfasilitasi interaksi antara program komputer dan membolehkan mereka untuk melakukan pertukaran informasi.(Massé dan Massé, 2012). Dalam sebuah API metode dan aturan didefinisikan untuk proses pertukaran data dan pesan yang dimengerti oleh aplikasi lainnya. Hal ini sangat penting digunakan untuk melakukan abstraksi dari fungsi dan metode yang sensitive yang hanya pengembang keahui(Kornienko et al., 2021)

Web service adalah sebuah system perangkat lunak yang digunakan untuk melakukan interaksi antara satu mesin dan yang lainnya dalam sebuah jaringan computer. Sistem ini biasanya akan terintegrasi dengan sebuah basis data untuk proses penyimpanan data dalam proses pengoprasiannya.(Yanuardi et al., 2024)

Framework adalah sebuah aplikasi semi komplit yang bisa diberikan sebuah spesialisasi untuk membuat sebuah aplikasi kustom dan dapat di gunakan kembali untuk mengembangkan aplikasi yang lain.

Framework dibuat untuk mengatasi masalah yang terdapat dalam pengembangan aplikasi web yaitu masalah pengulangan. Framework memberikan para pengembang kumpulan dari komponen kerangka yang memiliki beberapa karakteristik yaitu dapat bekerja dengan baik dengan aplikasi lain, framework siap untuk digunakan dalam proyek selanjutnya, dan framework dapat dimengerti dan digunakan oleh tim lain dalam organisasi.

Javascript adalah sebuah Bahasa pemrograman yang menggunakan konsep client side programming language dimana permrosesannya akan dilakukan oleh klien (Ananda dan Nama, 2024). Javascript ini memiliki kapabilitas untuk melakukan runtime object construction, variable parameter lists, function variables, dynamic script creation yang membuatnya menjadi Bahasa yang sangat powerfull untuk melakukan pengembangan.(*JavaScript / MDN*, 2024).

Express JS adalah sebuah framework atau kerangka kerja yang menggunakan bahasa pemrograman Javascript. Express JS adalah sebuah framework Javascript yang dikembangkan dari Node JS yang memberikan kumpulan fitur aplikasi web dan mobile yang kokoh. Dengan berbagai utilitas metode dan middleware HTTP pengembang dapat membuat API yang kokoh dengan cepat dan mudah.(*Express - Node.js Web Application Framework*, n.d.).

C# adalah sebuah Bahasa pemrograman yang gratis, cross platform, dan opensource. Sebuah aplikasi yang dibangun dengan menggunakan Bahasa ini dapat dijalankan dalam perangkat yang beragam. C# juga memiliki komunitas yang sangat ramai dengan jutaan pengembang sehingga memiliki ekosistem yang sangat terjaga (BillWagner, 2024).

ASP.NET CORE adalah sebuah framework yang diluncurkan pada tahun 2016(*What Is ASP.NET Core?*, n.d.) web aplikasi berbasis open source yang dibangun menggunakan Bahasa pemrograman C#. framework ini didesain untuk membangun aplikasi terkoneksi internet dan berbasis komputasi awan, seperti aplikasi web, aplikasi Internet Of Things, dan backend mobile. Framework ini berisikan komponen modular sehingga para pengembang dapat memiliki fleksibilitas dalam membuat sebuah solusi dari aplikasi yang ingin dikembangkan(*Dotnet/Aspnetcore*, 2014/2024).

Python adalah sebuah Bahasa yang powerful tetapi mudah dipelajari. Bahasa ini memiliki struktur data level tinggi yang efisien dan memiliki pendekatan yang simpel tetapi efektif dalam pemrograman berbasis objek. Dengan sintaks yang elegan dan pengetikan dinamis menjadikan python sebagai Bahasa yang sesuai untuk membuat script dan membangun aplikasi dengan cepat dalam berbagai area dan platform.(*The Python Tutorial*, n.d.)

Django adalah sebuah framework tingkat tinggi menggunakan Bahasa Python yang menganjurkan

pengembangan aplikasi web yang cepat dan bersih dengan desain pragmatic. Django dibangun oleh pengembang yang professional yang memudahkan proses pengembangan aplikasi tanpa harus bekerja dua kali. Beberapa kelebihan dari Django adalah proses pengembangan aplikasi yang super cepat, sudah berisikan dengan pendukung untuk proses pengembangan aplikasi web, keamanan yang diutamakan, dan sangat bisa untuk di kembangkan menjadi lebih besar (*Django Overview / Django*, n.d.).

Mysql adalah sebuah system manajemen basis data relasional paling populer yang dikembangkan, disebar, dan didukung oleh perusahaan Oracle. System manajemen basis data adalah sebuah aplikasi yang digunakan untuk menambahkan akses, dan memproses data yang tersimpan dalam basis data sebuah computer. Sql dari Mysql adalah sebuah kependekan dari Structured Query Language yang merupakan bahasa umum yang terstandarisasi untuk melakukan akses pada sebuah basis data (*MySQL :: MySQL 8.0 Reference Manual :: 1.2.1 What Is MySQL?*, n.d.).

Jmeter adalah sebuah aplikasi sumber terbuka yang merupakan 100% aplikasi yang menggunakan bahasa java yang di desain untuk melakukan pengujian beban dan melakukan pengukuran performa. Awalnya aplikasi Jmeter di desain hanya untuk aplikasi Web tetapi seiring waktu berjalan aplikasi ini sudah meluas dan dapat digunakan untuk menguji hal lain. Jmeter dapat digunakan untuk melakukan pengujian performa aplikasi web yang statis maupun dinamis. Jmeter juga dapat digunakan untuk melakukan simulasi beban berat kepada server atau jaringan sebagai pengujian kekuatannya agar dapat dianalisa performanya dalam pemberian beban yang berbeda. (*Apache JMeter - Apache JMeter™*, n.d.).

Visual studio code adalah sebuah editor kode sumber yang ringan tetapi kuat. Aplikasi editor ini dapat berjalan dalam system operasi windows, linux, dan mac dengan dukungan javascript, typescript dan node js yang sudah tertanam. Selain itu visual studio code juga memiliki ekosistem yang kaya akan Bahasa lain seperti Java, C++, Python, .Net dan lain lain. (*Documentation for Visual Studio Code*, n.d.).

Hyper Text Transfer Protocol(HTTP) adalah sebuah kumpulan dari protokol permintaan dan respon yang *stateless* yang memiliki kesamaan interface, semantic dan pesan yang memudahkan interaksi fleksibel antara komunikasi antar system informasi daring. (Fielding et al., 2022).

## 2. Metodologi

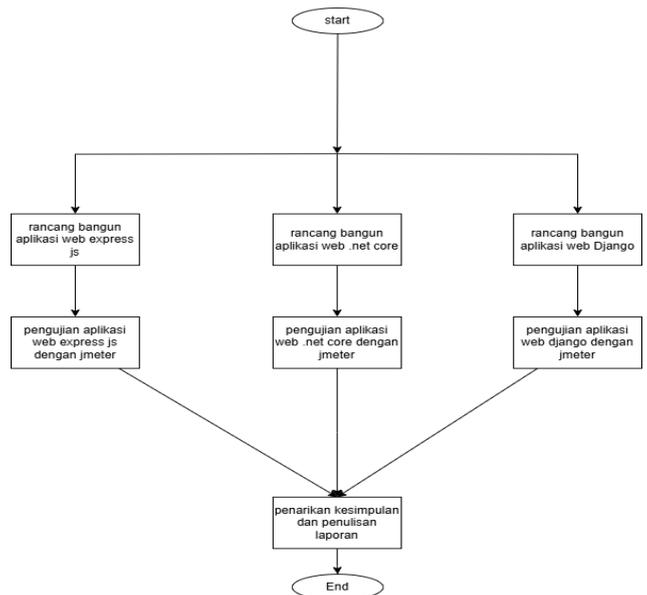
### 2.1. Alat dan bahan

Adapun alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut

Tabel 1. Alat dan bahan.

No	Nama	Fungsi
1	Laptop asus x441ub intel@ coretm i3-6006u dengan ram 8gb dan sistem operasi windows 10	Perangkat keras yang digunakan untuk melakukan proses pengembangan dan pengujian aplikasi.
2	Visual studio code versi 1.81.1	Perangkat lunak untuk melakukan pengembangan aplikasi web dengan framework express js, dan django
3	Node versi 20.1.0	Perangkat lunak untuk menjalankan kode javascript
4	.net versi 8.0	Basis framework yang digunakan .net core
5	Django versi 5	Framework aplikasi web berbasis python
6	Python versi 3.11	Bahasa pemrograman yang digunakan framework django
7	Apache jmeter versi 5.6.1	Aplikasi untuk melakukan pengujian web service
8	Mariadb versi 10.4.28	Database untuk menyimpan data yang dikirimkan.

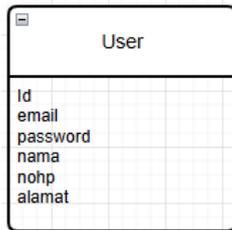
### 2.2. Prosedur percobaan



Gambar 1. Alur Pengujian

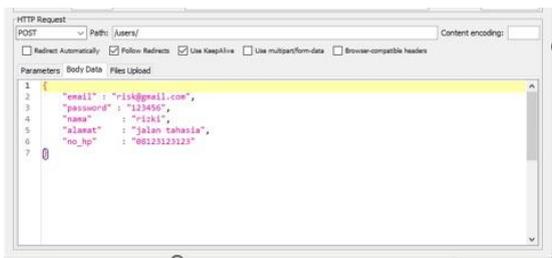
Dalam pengujian dibuat aplikasi web dengan framework Express Js, Django, dan .Net Core masing masing

sebuah REST API yang akan melakukan operasi CRUD kepada sebuah basis data yang sama. Kemudian dengan aplikasi Jmeter akan di simulasikan beban penggunaan yang bertambah dari 1000, 3000, 5000, dan 10000. Ketiga aplikasi web oleh yang dibuat dari masing masing framework akan di ukur performanya dan hasil akhirnya akan dibandingkan untuk menilai framework mana yang lebih unggul.



Gambar 2 Entity User

Dari gambar 2 dapat dilihat entity user yang berisikan data yang diterima oleh masing masing web service dan akan disimpan dalam database mysql. Data ini akan diterima melalui CRUD dengan menggunakan bentuk Javascript Object Notation (JSON).



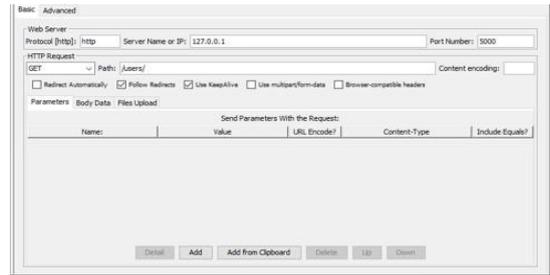
Gambar 3 Value Data JSON metode POST,PUT

Dari gambar 3 dapat dilihat data yang dikirimkan berbentuk pasangan *key and value* dimana *key* nya adalah kolom yang terdapat pada entity dan *value* nya adalah data sembarang. Data ini akan dikirimkan menggunakan metode *post* dan akan diterima oleh framework sebelum disimpan



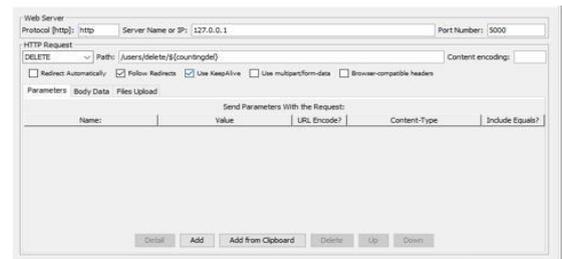
Gambar 4 Value Data JSON metode PUT

Dari gambar 4 dapat dilihat data untuk melakukan metode PUT hamper sama dengan data yang dikirimkan dengan metode POST. Namun terdapat perbedaan yaitu penambahan ID dengan parameter counting yang akan di ulang sebanyak data yang ada dalam *database*.



Gambar. 5 metode request metode GET

Dari gambar 5 dapat dilihat bahwa untuk metode GET tidak dikirimkan data berbentuk JSON. Dalam metode ini akan dilakukan hit pada alamat end point pada aplikasi untuk dan aplikasi akan mengirimkan data kepada klien.



Gambar. 6 metode request metode GET

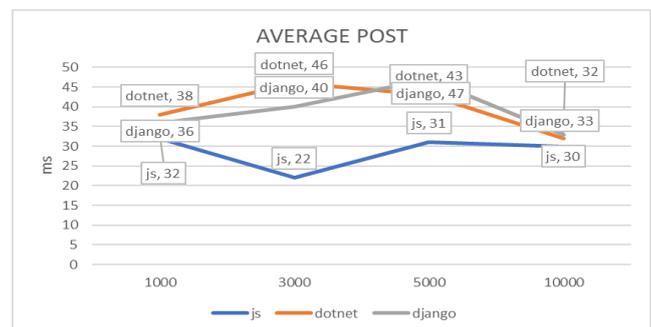
Dari gambar 6 dapat dilihat metode Delete sama dengan metode get hanya akan mengirimkan request. Tetapi dalam metode ini dibutuhkan id yang akan mengakses database untuk mencari data yang akan dihapus dengan parameter countingdel.

Semua data akan dikirimkan melalui aplikasi Jmeter. Aplikasi Jmeter ini juga akan melakukan pemantauan atas performa masing masing framework dengan mencatat waktu respon terendah, tertinggi, rata rata, tingkat galat, throughput, dan berapa banyak data yang diterima dan dikirimkan tiap detik.

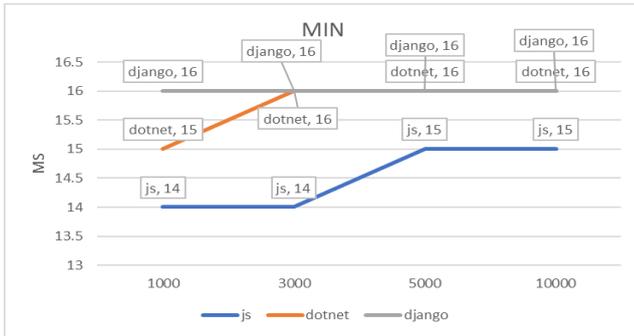
### 3. Hasil dan pembahasan

#### 3.1 Pengujian Post

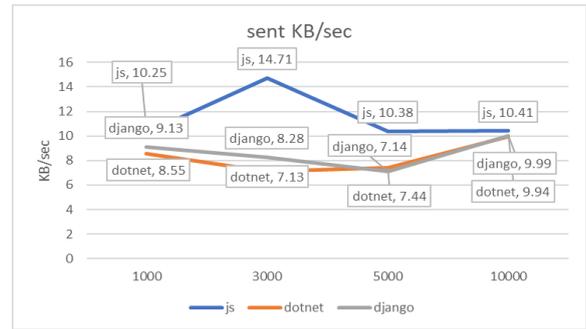
Dari pengujian Post didapatkan hasil sebagai berikut :



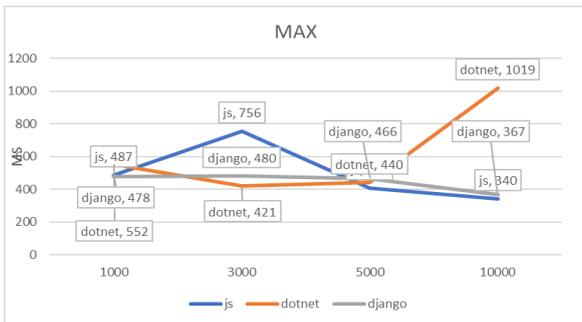
Gambar. 7 grafik Data average waktu respon pengujian metode POST



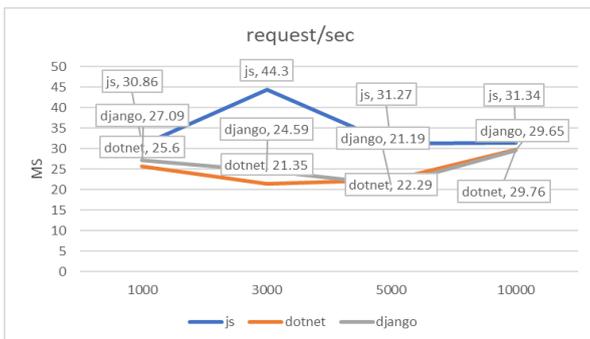
**Gambar. 8** grafik Data minimum waktu respon pengujian metode POST



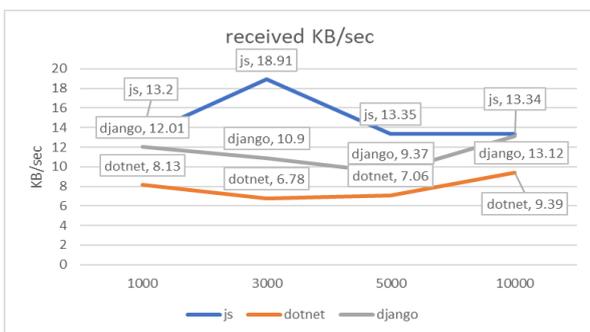
**Gambar. 12** grafik Data dikirim tiap detik pengujian metode POST



**Gambar. 9** grafik data maksimum waktu respon pengujian metode POST



**Gambar. 10** grafik Data waktu respon tiap detik pengujian metode POST

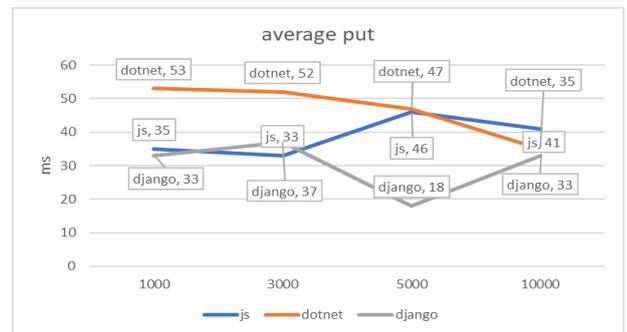


**Gambar. 11** grafik Data diterima tiap detik pengujian metode POST

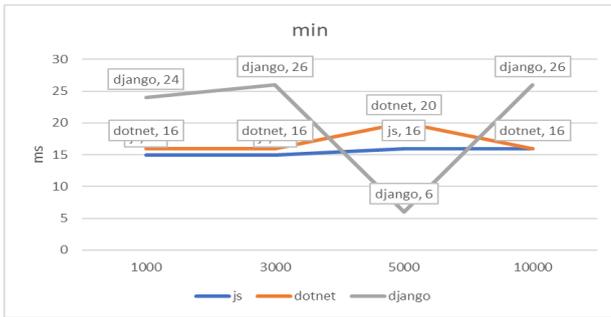
Gambar 7 hingga gambar 12 dapat dilihat hasil dari pengujian metode post didapatkan hasil yaitu Untuk nilai rata rata request paling rendah didapatkan pada framework expressJs dengan nilai 22ms pada pengujian 3000 request. Request paling cepat didapatkan pada expressJS yaitu 14ms pada pengujian dengan 3000 dan 5000 request. Nilai request paling lama yang terendah didapatkan pada framework django pada pengujian dengan 10000 request dengan waktu 367ms. untuk nilai throughput terbesar dimiliki oleh framework express js pada pengujian dengan 3000 request dengan nilai 44.30 request tiap detik. Data yang diterima tiap detik paling besar dimiliki oleh framework expressjs dengan nilai 18,91 KB/Secon. Nilai data yang dikirim tiap detik paling banyak juga didapatkan oleh expressjs dengan nilai 14,71 KB/sekond. Selanjutnya untuk ketiga framework didapatkan bahwa nilai dari error yang dialami adalah 0%.

### 3.2 Pengujian Put

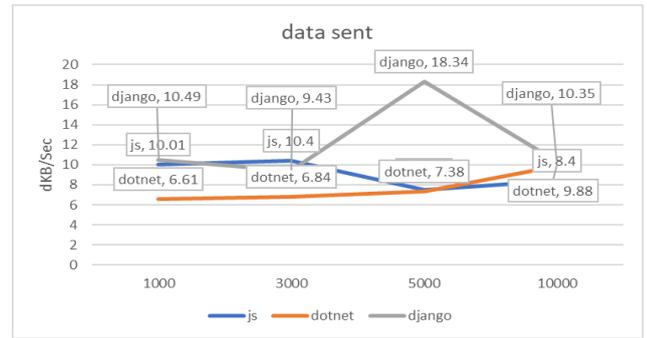
Dari pengujian metode PUT didapatkan data berikut :



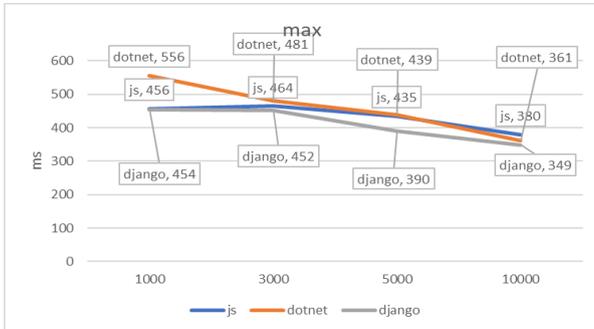
**Gambar. 12** grafik Data waktu rata rata pengujian metode PUT



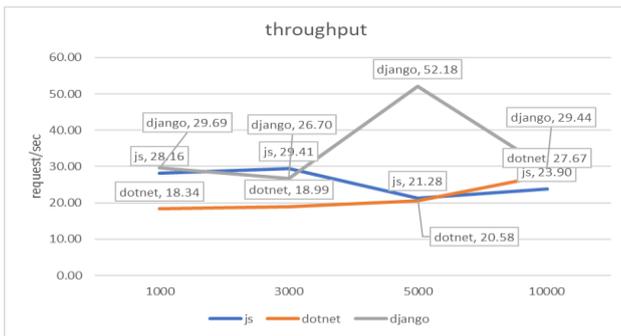
**Gambar. 13** grafik Data waktu respon minimal pengujian metode PUT



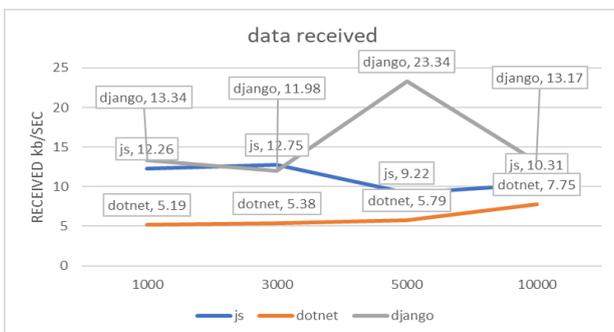
**Gambar. 17** grafik Data dikirim tiap detik pengujian metode PUT



**Gambar. 14** grafik Data waktu respon maksimal pengujian metode PUT



**Gambar. 15** grafik Data throughput pengujian metode PUT

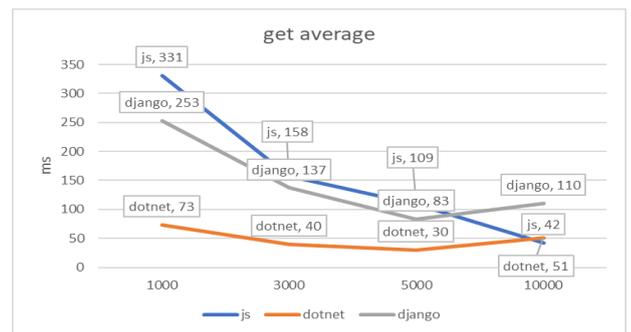


**Gambar. 16** grafik Data diterima tiap detik pengujian metode PUT

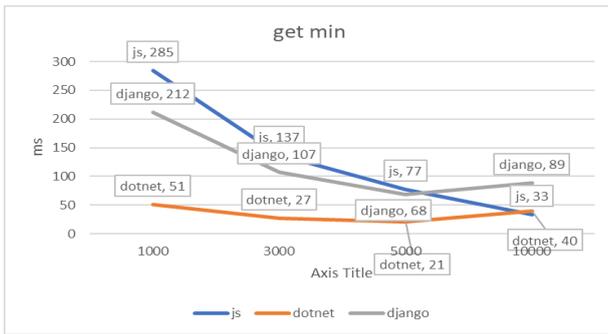
Gambar 12 hingga gambar 17 dapat dilihat hasil pengujian menggunakan metode put didapatkan hasil yaitu Untuk nilai rata rata request paling rendah didapatkan pada framework Django yaitu 18ms pada pengujian 5000 request. Request paling cepat didapatkan pada Django yaitu 6ms pada pengujian 5000 request. Dan request paling lama terendah didapatkan pada framework django dengan nilai 349ms pada pengujian 10000 request. Throughput paling besar didapatkan oleh framework Django dengan nilai 52,18 request/sekon pada pengujian 5000 request. Untuk data dikirim dan diterima paling banyak didapatkan oleh Django juga pada pengujian 5000 request/sekon. Kemudian untuk nilai error semua framework tidak mengalami error pada pengujian metode Put.

### 3.3 Pengujian metode Get

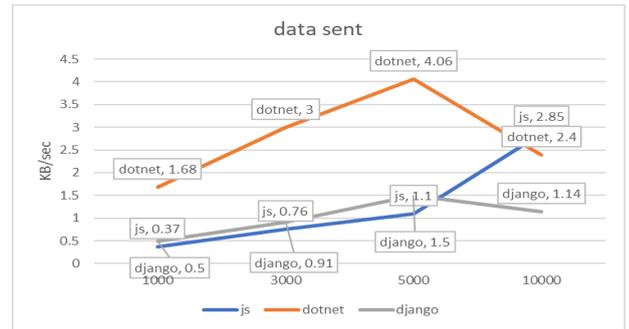
dari pengujian metode get didapatkan hasil berikut :



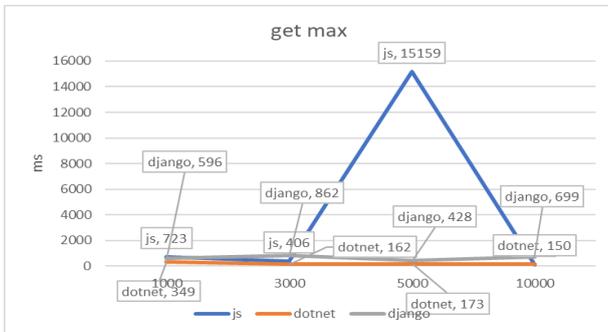
**Gambar. 18** grafik Data waktu rata rata request pengujian metode GET



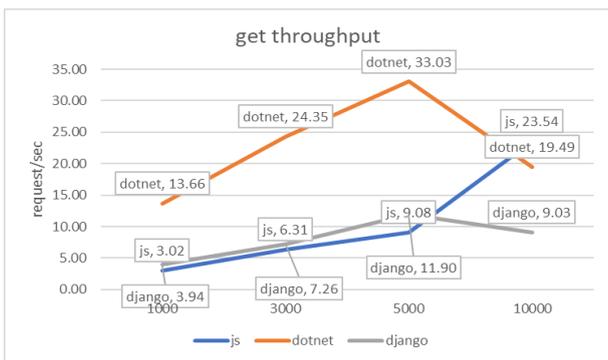
**Gambar. 19** grafik Data waktu minimal request pengujian metode GET



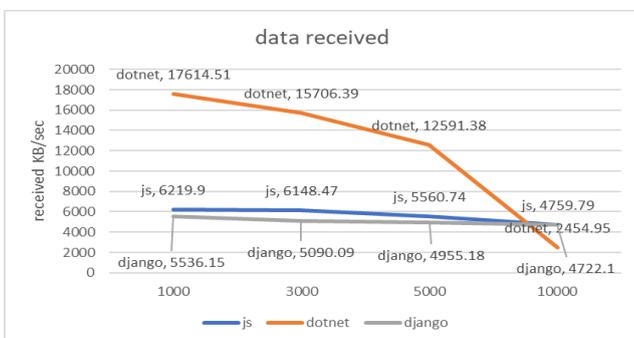
**Gambar 23** grafik Data dikirim tiap detik pengujian metode GET



**Gambar. 20** grafik Data waktu maksimal request pengujian metode GET



**Gambar. 21** grafik Data request tiap detik pengujian metode GET

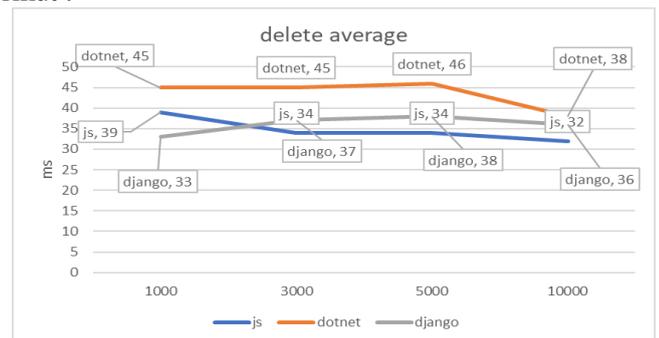


**Gambar. 22** grafik Data diterima tiap detik pengujian metode GET

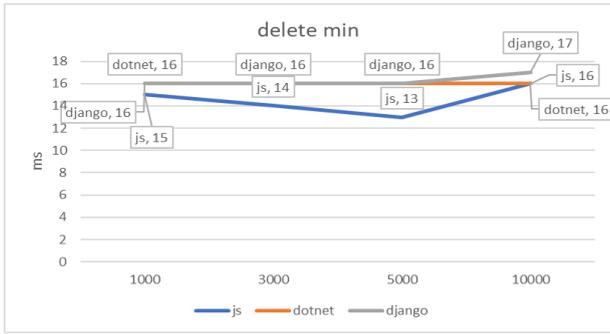
Gambar 18 hingga gambar 23 dapat dilihat hasil pengujian menggunakan metode get didapatkan hasil nilai request rata rata paling rendah didapatkan framework AS .NET Core core yaitu 30ms pada pengujian 5000 request. Nilai request paling cepat didapatkan oleh framework AS .NET Core core dengan nilai 21ms pada pengujian 5000ms. Nilai request paling lama terendah didapatkan oleh framework js yaitu 99ms pada pengujian 10000 request. nilai request paling banyak tiap detik didapatkan oleh framework AS .NET Core core pada pengujian 5000 request dengan nilai 33.03 request/detik. Untuk pengiriman dan penerimaan data paling besar tiap detik didapatkan oleh framework AS .NET Core pada pengujian 1000 request. Pada pengujian metode Get ketiga framework tidak mengalami error.

### 3.4 Pengujian metode Delete

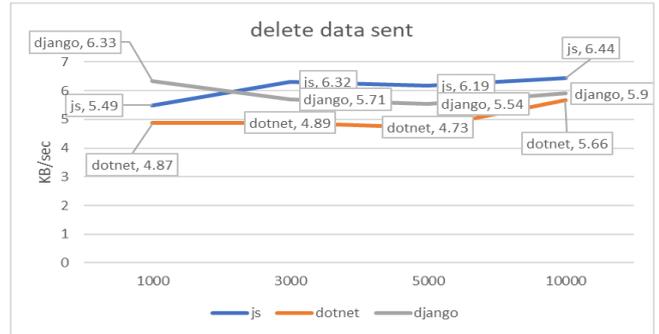
dari pengujian metode Delete didapatkan hasil berikut :



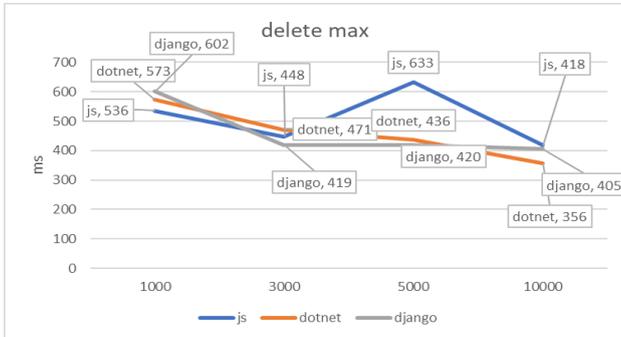
**Gambar. 24** grafik Data waktu rata rata request pengujian metode Delete



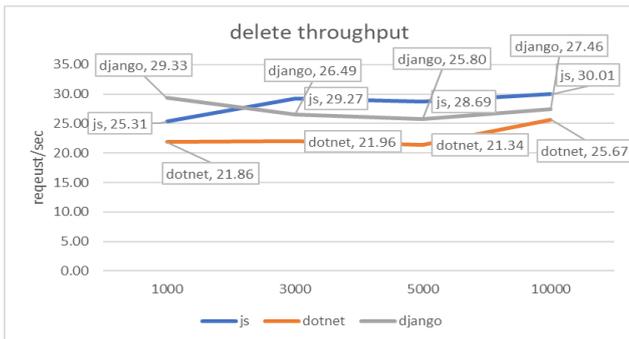
Gambar. 25 grafik Data waktu minimal request pengujian metode Delete



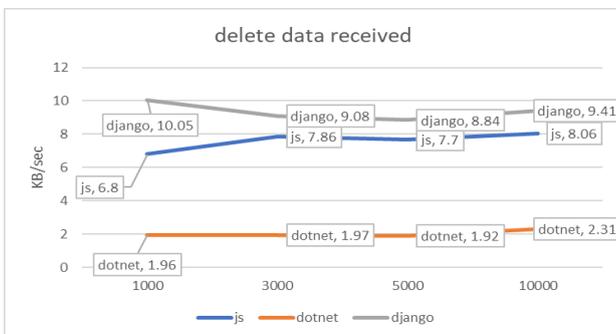
Gambar 29 grafik Data dikirim tiap detik pengujian metode GET



Gambar. 26 grafik Data waktu maksimal request pengujian metode Delete



Gambar. 27 grafik Data request tiap detik pengujian metode Delete

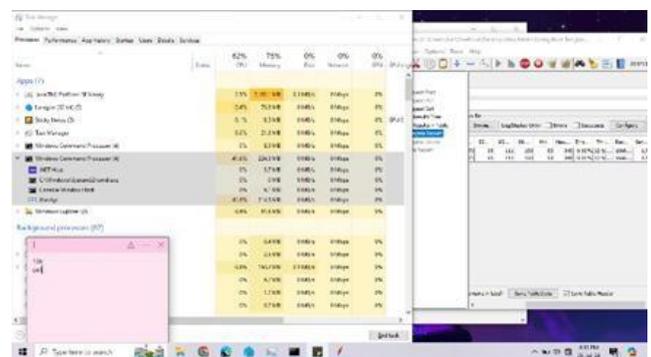


Gambar. 28 grafik Data diterima tiap detik pengujian metode GET

Gambar 24 hingga gambar 29 didapatkan hasil pengujian metode Delete yaitu pada rata rata nilai request didapatkan yang paling kecil adalah 32ms pada framework expressjs pada 10000 request. Untuk request paling cepat didaatkan pada framework expressjs dengan nilai 13ms pada pengujian 5000 request. Untuk nilai request paling lama yang terendah dimiliki oleh framework AS .NET Core core dengan 356ms pada pengujian 10000 request. nilai throughput paling tinggi dimiliki oleh framework express js dengan 30,01 request/detik pada pengujian 10000 request. Untuk nilai data dikirimkan dan diterima paling tinggi dimiliki oleh Django dengan nilai 10.05 KB/sekon untuk received dan 6.33KB/sekon untuk sent. Pada pengujian metode Delete juga tidak didapatkan error pada ketiga framework.

### 3.5 Pemantauan sumber daya

Dari pengujian didapatkan data penggunaan sumberdaya CPU dan penggunaan Random Access Memmory(RAM) yang paling Tinggi adalah framework express AS .NET Core core.



Gambar 30 resource penggunaan framework AS .NET Core core

Gambar 230 menampilkan resource penggunaan paling tinggi dalam pengujian yang didapatkan pada framework AS .NET Core core pada pengujian metode GET dengan 10000 request. Penggunaan resource

menyentuh 41,6% penggunaan CPU dan 224,3 MB RAM.

#### 4. Kesimpulan

Adapun kesimpulan yang dapat diperoleh dari penelitian ini adalah semua framework yang diuji memiliki nilai error yaitu 0% untuk pengujian berupa request 1000, 3000, 5000, 10000. *Framework AS .NET Core core* memiliki keunggulan yang sangat signifikan pada metode GET, yaitu rata-rata 73ms request tiap detik, nilai minimal 51ms, nilai max 349ms, throughput 33,03 request tiap detik, nilai data yang diterima 17614.51 KB tiap detik dan nilai data yang dikirim 1,68 KB tiap detik. *Framework AS .NET Core core* menggunakan sumber daya yang paling tinggi dibandingkan framework lainnya. Hal ini dapat dilihat pada pengujian metoda GET yaitu dengan 41,6% penggunaan prosesor dan 224,3 MB penggunaan RAM. *Framework Express JS* adalah *framework* yang menggunakan sumber daya paling rendah dibandingkan framework lainnya. Hal ini dapat dilihat pada pengujian metoda DELETE yaitu dengan 5,3% penggunaan prosesor dan 77 MB penggunaan RAM.

#### Daftar Pustaka

- Ananda, R. A., & Nama, G. F. (2024). ANALISIS DAN PERANCANGAN LAYANAN STREAMING FILM BERBASIS WEB LANGGANAN MENGGUNAKAN FRAMEWORK NEXTJS. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(1). <https://doi.org/10.23960/jitet.v12i1.3967>
- Apache JMeter—Apache JMeter™*. (n.d.). Retrieved October 30, 2023, from <https://jmeter.apache.org/index.html>
- BillWagner. (2024, May 8). *Overview—A tour of C#*. <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview>
- Dhalla, H. K. (2021). A Performance Comparison of RESTful Applications Implemented in Spring Boot Java and MS.NET Core. *Journal of Physics: Conference Series*, 1933(1), 012041. <https://doi.org/10.1088/1742-6596/1933/1/012041>
- Django overview | Django*. (n.d.). Retrieved April 26, 2024, from <https://www.djangoproject.com/start/overview/>
- Documentation for Visual Studio Code*. (n.d.). Retrieved September 4, 2024, from <https://code.visualstudio.com/docs>
- Dotnet/aspnetcore*. (2024). [C#]. .NET Platform. <https://github.com/dotnet/aspnetcore> (Original work published 2014)
- Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*, 12(9), 4369. <https://doi.org/10.3390/app12094369>
- Express—Node.js web application framework*. (n.d.). Retrieved April 25, 2024, from <https://expressjs.com/>
- Fielding, R. T., Nottingham, M., & Reschke, J. (2022). *HTTP Semantics* (Request for Comments No. RFC 9110). Internet Engineering Task Force. <https://doi.org/10.17487/RFC9110>
- JavaScript | MDN*. (2024, March 5). <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Karlsson, O., & Larsson-Green, P. (n.d.). *A Performance comparison Between ASP.NET Core and Express.js for creating Web APIs*.
- Kornienko, D. V., Mishina, S. V., Shcherbatykh, S. V., & Melnikov, M. O. (2021). Principles of securing RESTful API web services developed with python frameworks. *Journal of Physics: Conference Series*, 2094(3), 032016. <https://doi.org/10.1088/1742-6596/2094/3/032016>
- Massé, M. H., & Massé, M. (2012). *REST API design rulebook: Designing consistent RESTful Web Service Interfaces*. O'Reilly.
- MySQL :: MySQL 8.0 Reference Manual: 1.2.1 What is MySQL?* (n.d.). Retrieved August 1, 2024, from <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- The Python Tutorial*. (n.d.). Python Documentation. Retrieved September 4, 2024, from <https://docs.python.org/3/tutorial/index.html>
- What is ASP.NET Core? | .NET*. (n.d.). Microsoft. Retrieved September 4, 2024, from <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core>
- Yanuardi, M. I., Aminudin, A., & Faiqurahman, M. (2024). Analisis Perbandingan Efektivitas Arsitektur Restful dan Arsitektur Grpc pada Implementasi Web Service. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 10(2), 333. <https://doi.org/10.26418/jp.v10i2.80845>