

# Implementasi Teknik Clustering untuk Meningkatkan Performa Aplikasi Node JS

Bahrul Rozak<sup>1</sup>, Erizal<sup>\*2</sup>, Firman Noor Hasan<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Universitas Muhammadiyah Prof. Dr. Hamka

E-mail: <sup>1</sup>[bahrulrozak@uhamka.ac.id](mailto:bahrulrozak@uhamka.ac.id), <sup>\*2</sup>[erizal@uhamka.ac.id](mailto:erizal@uhamka.ac.id),

<sup>3</sup>[firman.noorhasan@uhamka.ac.id](mailto:firman.noorhasan@uhamka.ac.id)

## Abstrak

Permasalahan yang kerap kali muncul pada aplikasi berbasis server side, ialah permintaan request dan response dalam jumlah yang besar. Proses request serta response akan terus berlanjut selama pengguna berinteraksi dengan aplikasi, Jika hal ini terus berlanjut maka penggunaan sumber daya (resource) yang berlebih pada CPU dapat menjadikan performa aplikasi tidak optimal bahkan dapat menimbulkan crash, yang berdampak pada layanan dan kualitas aplikasi. Oleh karena itu, diperlukan teknik untuk menangani permasalahan tersebut. Metode yang diimplementasikan pada penelitian ini ialah membuat aplikasi berbasis sever side dengan dua spesifikasi yaitu dengan module cluster dan tanpa module cluster, selanjutnya kedua aplikasi dengan spesifikasi yang berbeda, masing-masing akan dilakukan tahap pengujian performa serta monitoring, kemudian dilakukan proses analisa hasil perbandingan untuk mendapatkan kesimpulan. Module cluster akan membantu untuk melakukan teknik clustering, teknik clustering ialah mengelompokkan proses yang sama serta sering dieksekusi. Dengan teknik ini beban kerja pada CPU akan terdistribusi, sehingga memberikan peningkatan performa aplikasi.

**Kata Kunci**—Clustering, Performa, Aplikasi, Node js

## 1. PENDAHULUAN

Awalnya JavaScript adalah bahasa pemrograman *scripting* yang dibuat dan dikembangkan untuk membuat aplikasi berbasis *client side* [1][2]. Namun sejak hadirnya teknologi *runtime environment* yaitu Node JS [3], JavaScript dapat digunakan untuk membuat aplikasi berbasis *server side* [4]. Node JS bekerja dengan cara mengeksekusi kode JavaScript menggunakan *engine* yang diadopsi oleh browser yaitu V8 [5]. *Engine* V8 yang diadopsi oleh teknologi Node JS memberikan manfaat, salah satunya yaitu meningkatkan kecepatan untuk melayani dan menangani proses *request* serta *response* dari pengguna lebih cepat dan *responsive* [6]. Karena hal tersebut, maka JavaScript memungkinkan untuk dikembangkan diluar *environment client side* untuk membuat aplikasi *server side* yang dapat melayani dan menangani proses *request* dan *response* dari pengguna [7]. Dengan fitur *non-blocking input output* pada Node JS sehingga memungkinkan proses *request* dilakukan secara *parallel* dan memberikan luaran berupa *response* secara cepat dan efisien [8]. Proses *request* dan *response* ini akan terus berlangsung terjadi selama pengguna berinteraksi dengan aplikasi [9]. Masalah yang kerap kali muncul pada proses tersebut ialah permintaan *request* dan *response* dalam jumlah yang besar [10]. Sehingga penggunaan sumber daya yang berlebih pada CPU, memori dan *bandwith* yang berperan untuk melayani dan menangani proses tersebut [11]. Akibatnya aplikasi dapat mengalami *crash* yang dapat mempengaruhi performa dan *latensi* menjadi lambat [12]. Menyebabkan penurunan kualitas, produktivas dan layanan terhadap pengguna aplikasi [13]. Oleh karena itu, diperlukan teknik untuk mengoptimalkan kinerja aplikasi agar mampu bekerja secara efisien dan stabil dari segi performa. Ada banyak teknik yang dapat diimplementasikan untuk mengoptimalkan kinerja aplikasi diantaranya pemanfaatan *cache (caching)*[14], menerapkan *load balancing*[15] dan

*clustering* [16]. Namun pada penelitian ini, peneliti hanya akan berfokus pada implementasi teknik *clustering*. Teknik *Clustering* merupakan cara pengelompokan proses berdasarkan karakteristik dan kesamaan fitur [17]. Sehingga beban kerja pada CPU akan terdistribusi ketika menangani proses komputasi dan meningkatkan performa aplikasi.

## 2. METODE PENELITIAN.

### 2.1 Identifikasi Kebutuhan

#### 2.1.1 Hardware

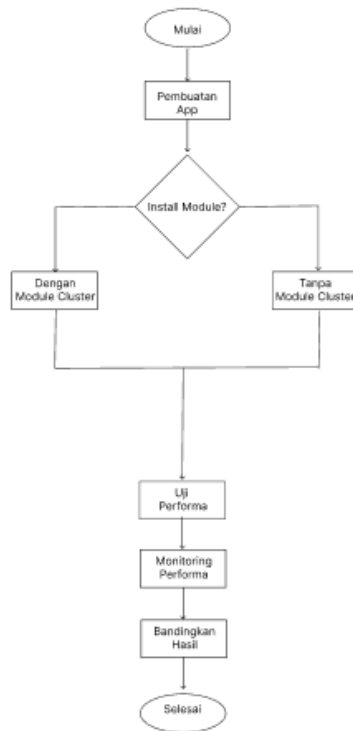
Adapun spesifikasi hardware yang digunakan pada penelitian memiliki CPU AMD Dual Core A4-9125 up to 2.6 GHz, RAM 4GB serta system operasi yaitu Microsoft Windows 10.

#### 2.1.2 Software

Adapun software yang digunakan pada penelitian ini meliputi Visual Studio Code, Node JS v18.9.0, NPM v8.5.5, cluster module dan *performance matrix tools*.

### 2.2 Alur Proses Clustering

Adapun alur dari proses *clustering* pada penelitian disajikan pada gambar 1.



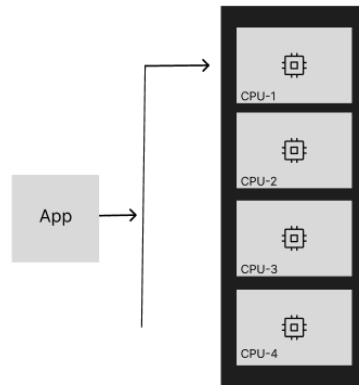
Gambar 1 Alur Proses Clustering

Pada tahapan ini penelitian dimulai dengan proses pembuatan aplikasi server dengan menggunakan express framework, aplikasi dibuat menjadi dua spesifikasi yaitu tanpa module *cluster* dan dengan module *cluster*, selanjutnya kedua aplikasi dengan spesifikasi yang berbeda masing-masing dilakukan proses pengujian performa dan *monitoring* hasil performa. Selanjutnya dilakukan proses analisa hasil perbandingan untuk mendapatkan kesimpulan. .

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Arsitektur tanpa Cluster

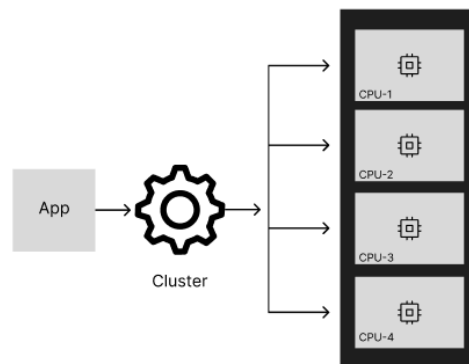
Berikut merupakan arsitektur aplikasi server yang dibuat tanpa menggunakan *cluster*.



Gambar 2 Arsitektur Tanpa Cluster

Pada gambar 2 proses hanya dilakukan pada satu CPU, akibatnya beban kinerja CPU dapat bermasalah jika proses *request* dan *response* terus terjadi, sehingga mengakibatkan performa aplikasi melambat atau bahkan menimbulkan *crash*.

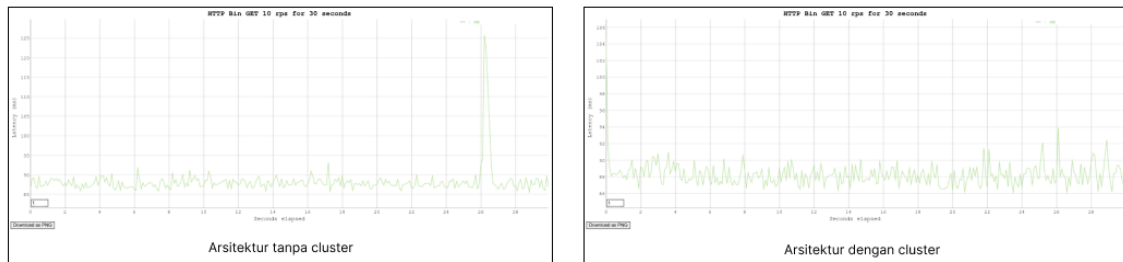
#### 3.2 Arsitektur dengan Cluster



Gambar 3 Arsitektur dengan Cluster

Pada gambar 3 proses dilakukan dengan tambahan module *cluster*. Module *cluster* dapat membantu untuk melakukan teknik *clustering*. Pada teknik *clustering* segala bentuk proses yang sama yang sering dilakukan akan dikelompokkan dan mendistribusikan beban kinerja pada CPU sehingga memberikan peningkatan performa pada aplikasi.

### 3.3 Hasil perbandingan performa



Gambar 4 Hasil Perbandingan Performa

Pada gambar 4 merupakan hasil perbandingan performa, setelah dilakukan tahap pengujian dan *monitoring* menggunakan skenario *load testing* pada masing-masing aplikasi. Aplikasi dengan module *cluster* terlihat jelas mampu menangani sejumlah *request* dan *response* dengan baik, sehingga penggunaan CPU stabil dan performa aplikasi juga meningkat.

## 4. KESIMPULAN

Berdasarkan hasil, pembahasan serta tahapan yang dilakukan maka penggunaan teknik *clustering* dengan module *cluster* dapat meningkatkan performa aplikasi Node JS, hal ini dikarenakan dengan menggunakan teknik *clustering*, beban kerja pada CPU akan terdistribusi dengan baik.

Saran untuk penelitian selanjutnya ialah mengembangkan, meluaskan serta memperbanyak teknik-teknik mutakhir untuk membantu meningkatkan performa aplikasi berbasis Node JS.

## DAFTAR PUSTAKA

- [1] H. Purwanto and A. Rahmawati, "SISTEM INFORMASI PELAPORAN LPG (SIMPEL) BERBASIS WEB DENGAN MODEL MULTI USER," vol. 9, no. 1, 2020.
- [2] N. Višnjevac, R. Mihajlović, M. Šošković, Ž. Cvijetinović, and B. Bajac, "Prototype of the 3D cadastral system based on a NoSQL database and a Javascript visualization application," *ISPRS Int J Geoinf*, vol. 8, no. 5, May 2019, doi: 10.3390/ijgi8050227.
- [3] N. Vasilakis *et al.*, "Preventing Dynamic Library Compromise on Node.js via RWX-Based Privilege Reduction," in *Proceedings of the ACM Conference on Computer and Communications Security*, Nov. 2021, pp. 1821–1838. doi: 10.1145/3460120.3484535.
- [4] N. van Ginkel, W. de Groef, F. Massacci, and F. Piessens, "A Server-Side JavaScript Security Architecture for Secure Integration of Third-Party Libraries," *Security and Communication Networks*, vol. 2019, 2019, doi: 10.1155/2019/9629034.
- [5] J. Jueckstock and A. Kapravelos, "VisibleV8: In-browser monitoring of JavaScript in the wild," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, Oct. 2019, pp. 393–405. doi: 10.1145/3355369.3355599.
- [6] K. Stefanoski, A. Karadimche, and I. Dimitrievski, "PERFORMANCE COMPARISON OF C++ AND JAVASCRIPT (NODE.JS-V8 ENGINE)," 2019. [Online]. Available: <https://www.researchgate.net/publication/336408691>

- [7] S. Kumar, "A REVIEW ON CLIENT-SERVER BASED APPLICATIONS AND RESEARCH OPPORTUNITY," *Article in International Journal of Scientific Research*, 2019, doi: 10.24327/ijrsr.2019.1007.3768.
- [8] K. I. D. Kyriakou and N. D. Tselikas, "Complementing JavaScript in High-Performance Node.js and Web Applications with Rust and WebAssembly," *Electronics (Switzerland)*, vol. 11, no. 19, Oct. 2022, doi: 10.3390/electronics11193217.
- [9] B. Jabiyev, O. Mirzaei, A. Kharraz, and E. Kirda, "Preventing server-side request forgery attacks," in *Proceedings of the ACM Symposium on Applied Computing*, Mar. 2021, pp. 1626–1635. doi: 10.1145/3412841.3442036.
- [10] A. Anwyl-Irvine, E. S. Dalmaijer, N. Hodges, and J. K. Evershed, "Realistic precision and accuracy of online experiment platforms, web browsers, and devices", doi: 10.3758/s13428-020-01501-5/Published.
- [11] S. Vila, N. Moraes, F. Guirado, J. L. L rida, F. Cores, and F. L. Verdi, "Cloud Resources Reassignment Based on CPU and Bandwidth Correlation," in *Advances in Intelligent Systems and Computing*, 2020, vol. 1151 AISC, pp. 1101–1113. doi: 10.1007/978-3-030-44041-1\_95.
- [12] L. Fan *et al.*, "Large-scale analysis of framework-specific exceptions in Android apps," May 2018, pp. 408–419. doi: 10.1145/3180155.3180222.
- [13] J. Bughin, T. Catlin, M. Hirt, and P. Willmott, "Why digital strategies fail."
- [14] D. Lomet, "Cost/performance in modern data stores: How data caching systems succeed," in *14th International Workshop on Data Management on New Hardware, DaMoN 2018*, Jun. 2018. doi: 10.1145/3211922.3211927.
- [15] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature," *IEEE Access*, vol. 6. Institute of Electrical and Electronics Engineers Inc., pp. 14159–14178, Mar. 04, 2018. doi: 10.1109/ACCESS.2018.2805842.
- [16] O. H. Jader, S. R. M. Zeebaree, and R. R. Zebari, "A State Of Art Survey For Web Server Performance Measurement And Load Balancing Mechanisms Image Processing and Security View project Parallel Processing and Distributed Systems View project Omid H Jader Erbil polytechnic university A State Of Art Survey For Web Server Performance Measurement And Load Balancing Mechanisms," *Article in International Journal of Scientific & Technology Research*, vol. 8, 2019, [Online]. Available: www.ijstr.org
- [17] A. Shahidinejad, M. Ghobaei-Arani, and M. Masdari, "Resource provisioning using workload clustering in cloud computing environment: a hybrid approach," *Cluster Comput*, vol. 24, no. 1, pp. 319–342, Mar. 2021, doi: 10.1007/s10586-020-03107-0.