

MANUEVER, APLIKASI MONITORING SUMBER DAYA VIRTUAL SERVER DI LINGKUNGAN IOT

Iin Karmila Yusri^{1,3}, Misnawati¹, Kasim³, dan Andi Rifqial Nur¹)

¹Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang, Makassar, 90245

² Center of Applied IT Research (CAIR), Politeknik Negeri Ujung Pandang, Makassar,
90245

E-mail: iin.yusri@poliupg.ac.id

Abstract

The utilization of virtual servers greatly supports the implementation of IoT in integrating many different IoT devices. However, virtual servers often experience interference in terms of their resources. Monitoring the use of virtual server resource capacity must be done to prevent the decline of server services. However, resource monitoring is not enough. There needs to be action during critical conditions of server resources. Virtual server resource management is needed to prevent critical conditions from lasting long and can cause the entire system in the network to go off. This research aims to create a mobile application that can monitor the condition of virtual server resources and also increase the capacity of these resources to prevent servers from going off automatically and manually. This application is built using Proxmox and Nginx. To access the Maneuver application, an internet connection is required. Based on the test results of the Virtual Server Monitoring Application, it is concluded that the Maneuver Application has been successfully made in accordance with the design that has been made. The Maneuver application can monitor the condition of virtual server resources in real-time and the information is displayed in graphical form. The Maneuver application can also add resource capacity manually and automatically.

Keywords: *resources monitoring; virtual server ; IoT; mobile application*

Abstrak

Pemanfaatan virtual server sangat mendukung implementasi IoT dalam mengintegrasikan banyak perangkat IoT yang berbeda. Namun virtual server sering kali mengalami gangguan dari sisi sumber dayanya. Monitoring penggunaan kapasitas sumber daya *virtual server* harus dilakukan untuk mencegah menurunnya layanan server. Namun, monitoring sumber daya belum cukup. Perlu ada tindakan saat kondisi kritikal sumber daya server. Pengaturan sumber daya virtual server sangat dibutuhkan untuk mencegah kondisi kritis berlangsung lama dan bisa menyebabkan seluruh sistem dalam jaringan menjadi *off*. Penelitian ini bertujuan untuk membuat aplikasi mobile yang dapat memonitor kondisi sumber daya virtual server dan juga menambah kapasitas sumber daya tersebut untuk mencegah server *off* secara otomatis dan manual. Aplikasi ini dibangun dengan menggunakan Proxmox dan Nginx. Untuk mengakses aplikasi Manuver dibutuhkan koneksi internet. Berdasarkan hasil pengujian dari Aplikasi Monitoring Virtual Server ini disimpulkan bahwa Aplikasi Manuver berhasil dibuat sesuai dengan perancangan yang telah dibuat. Aplikasi Manuver dapat memonitoring kondisi sumber daya virtual server secara real-time dan informasi ditampilkan dalam bentuk grafis. Aplikasi Manuver juga dapat melakukan penambahan kapasitas sumber daya secara manual dan otomatis.

Kata Kunci: *monitoring sumber daya; virtual server; IoT; aplikasi mobile;*

PENDAHULUAN

Seiring berkembangnya teknologi, komputasi lebih banyak dilakukan secara virtual. Pemanfaatan teknologi virtualisasi untuk server dilakukan dengan membagi sumber daya server fisik menjadi beberapa server virtual (Suryayusra et al., 2022). Begitu pula dalam lingkungan Internet of Things (IoT). Pemanfaatan *virtual server* sangat mendukung implementasi IoT yang mengintegrasikan banyak perangkat IoT (Dewi et al., 2021).

Dengan adanya virtual server pada sistem IoT, administrator IoT dapat melakukan beberapa operasi seperti halnya pada server yang memiliki wujud fisik (Dwiyatno et al., 2020). Hanya saja, virtual server sering kali mengalami gangguan dari sisi sumber dayanya yaitu *Central Processing Unit (CPU)*, *Hard Disk Drive (HDD)* dan *Random Access Memory (RAM)* (Soni et al., 2019). Virtual server bisa terganggu bahkan *off* secara tiba-tiba karena tidak mampu lagi melayani permintaan dengan sumber daya yang terbatas (Suryayusra et al., 2022). Mengatasi kondisi tersebut diperlukan monitoring sumber daya *virtual server* oleh administrator guna menghindari terjadinya masalah operasional server (Soni et al., 2019). Pemantauan penggunaan kapasitas sumber daya *virtual server* harus dilakukan untuk mencegah menurunnya layanan server.

Pemantauan sumber daya dapat dilakukan dengan mengakses *virtual server* dari jarak jauh (Armipajasa, 2022). Namun pemantauan masih dilakukan per server menyebabkan *maintenance virtual server* memakan waktu yang lama.

Sistem monitoring sumber daya server ada yang berbasis website dan android. Pada sistem monitoring sumber daya berbasis website, administrator tidak perlu lagi melakukan *remote* server satu-per-satu lagi. Administrator cukup memasukkan username, password dan IP address dari server fisik maka isi dari seluruh *virtual server*-nya dapat langsung ditampilkan (Christanto & Suprayogi, 2017; Prasetyo et al., 2022). Namun, pemantauan menggunakan website kurang efektif dan tidak aman. Administrator harus masuk ke browser dahulu untuk membuka link atau alamat website untuk mengakses layanan yang berpotensi mengganggu keamanan server fisik karena IP Address-nya terpublikasi (Azizi et al., 2022). Sistem pemantauan server berbasis android telah di implementasikan di Kantor Dinas Kominfo Kabupaten Rembang (Azizi et al., 2022). Namun pemantauan yang dilakukan hanya kondisi on dan off server saja.

Monitoring sumber daya belum cukup. Perlu ada tindakan saat kondisi kritikal sumber daya server, seperti kapasitas hard drive dan/atau RAM yang minim, pemakaian

CPU berlebih (Christanto & Suprayogi, 2017). Pengaturan sumber daya virtual server sangat dibutuhkan untuk mencegah kondisi kritis berlangsung lama dan bisa menyebabkan seluruh sistem dalam jaringan menjadi *off*. Kondisi kritis tersebut dapat diatasi dengan penambahan kapasitas sumber daya virtual server (Afrizal & Prihanto, 2022).

Penelitian ini bertujuan untuk membuat aplikasi mobile yang dapat memonitor kondisi sumber daya virtual server dan juga mengatur kapasitas sumber daya tersebut untuk mencegah server *off* secara otomatis dan manual.

METODE PENELITIAN

Metode yang diterapkan pada penelitian ini ditunjukkan pada Gambar 1. Penelitian ini terdiri atas lima tahapan terstruktur untuk mencapai tujuan penelitian. Penelitian diawali dengan studi literatur tentang sistem monitoring server. Kemudian dilanjutkan dengan analisis kebutuhan. Tahap Analisis kebutuhan dilakukan untuk mengetahui kebutuhan sistem, meliputi perangkat lunak, perangkat keras, dan fungsional sistem.

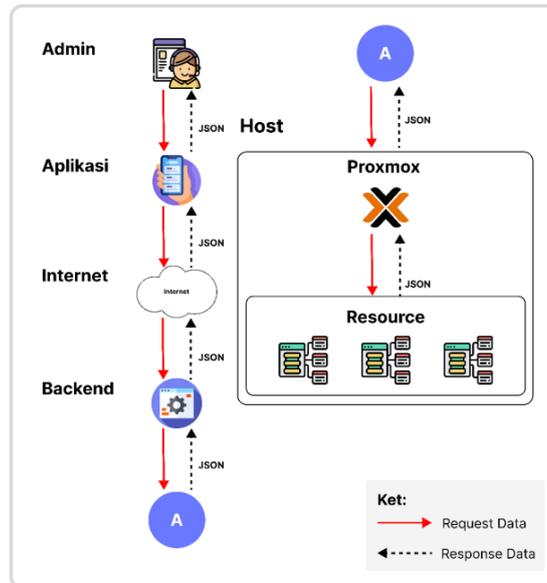


Gambar 1. Prosedur Penelitian

Berikutnya adalah tahap desain sistem yang merupakan tahapan pendefinisian dari analisis sistem dan analisis kebutuhan yang menggambarkan secara detail sistem yang dirancang.

Gambar 2 menunjukkan model perancangan diagram sistem yang dibuat berdasarkan dari hasil analisa kebutuhan. Admin merupakan user dari Aplikasi Manuver yang merupakan output dari penelitian ini yang akan mengatur sumber daya virtual server. Data akan di *request* dari aplikasi yang dilakukan oleh admin. Untuk mengakses aplikasi diperlukan koneksi internet. *Backend* merupakan bagian yang menghubungkan Proxmox dan aplikasi yang dibuat yang berisi pengolahan *request* dari aplikasi dan mengatur bagaimana data akan dikembalikan dalam bentuk *response*. Pada *backend* terdapat *endpoint* dari Proxmox yang merupakan tools yang digunakan untuk membuat virtual server dan menyediakan API. Semua data dari Proxmox dapat diakses melalui *endpoint*.

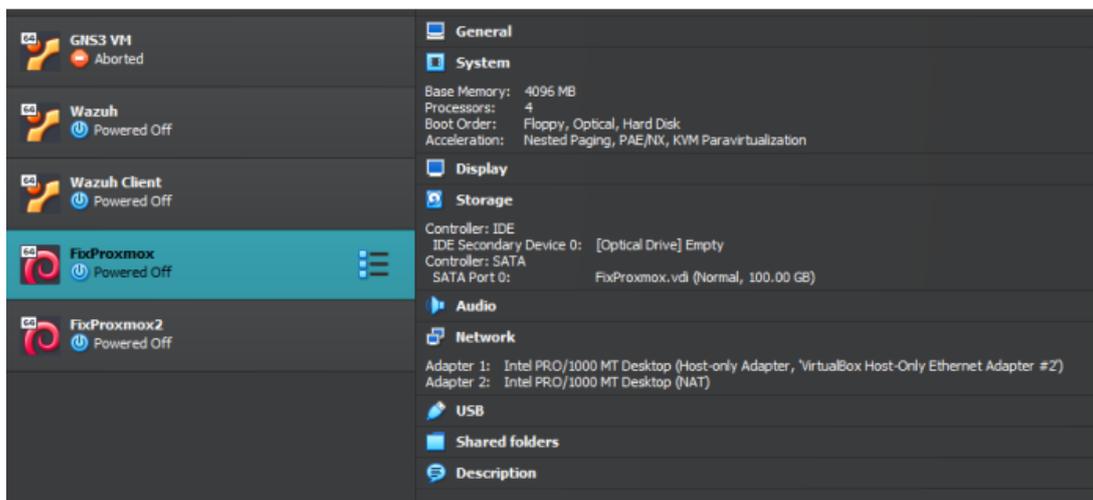
Host merupakan perangkat keras yang didalamnya telah terinstal Proxmox yang akan dimonitoring *resource*-nya yaitu CPU, RAM dan HDD.



Gambar 2. Model perancangan diagram sistem

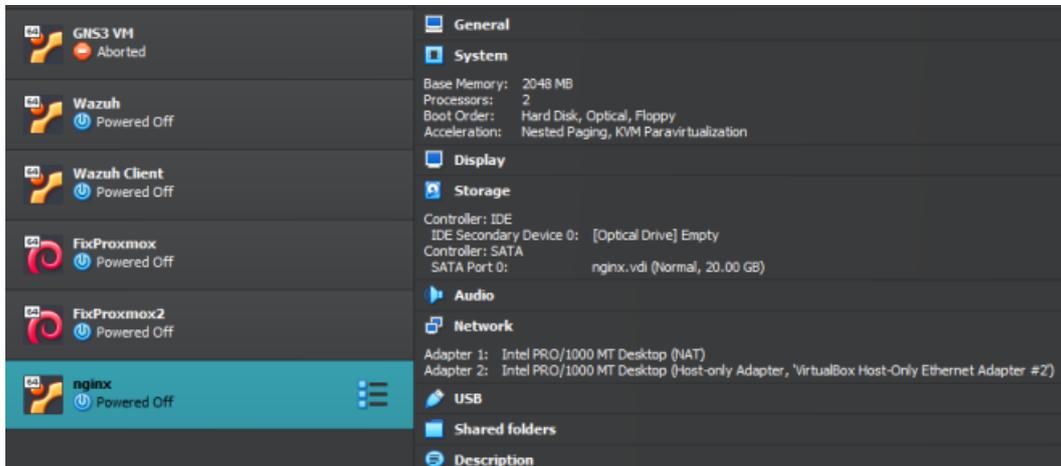
HASIL DAN PEMBAHASAN

Hasil dan luaran yang dicapai merupakan bagian dari tahapan implementasi dan pengujian yang diawali dengan pembuatan *Virtual Server* berbasis lokal dengan melakukan penginstalan aplikasi Proxmox pada Virtual Box. Proxmox ini nantinya yang akan menjadi server-server yang dapat diakses pada aplikasi Manuver. Pada aplikasi Virtual Box, Proxmox diinstal dengan ketentuan seperti pada Gambar 3.



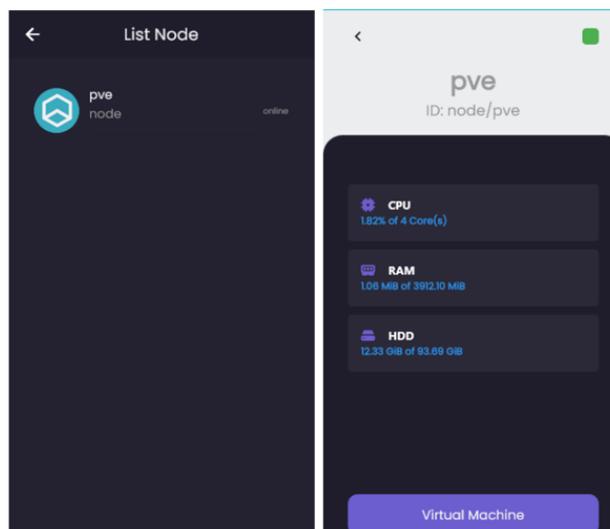
Gambar 3. Proxmox Machine

Langkah selanjutnya adalah pembuatan Server Nginx sebagai *Reverse Proxy* yang didalamnya terdapat *database* dan API dari Aplikasi yang dibuat. Untuk membuat server Nginx diawali dengan membuat satu buah mesin baru di Virtual Box dengan spesifikasi seperti pada Gambar 4 dan melakukan instalasi Nginx pada mesin yang telah dibuat.



Gambar 4. Nginx Machine

Selanjutnya melakukan pemindahan file API ke dalam server. Dalam pemindahan file digunakan perintah `scp` lalu diikuti dengan alokasi file dari API di laptop lalu diikuti dimana file API tersebut akan disimpan pada mesin. Pemindahan *database* lokal ke dalam server Nginx juga dilakukan dengan cara meng-instal Mysql pada server Nginx. Setelah konfigurasi telah selesai maka selanjutnya API dapat dijalankan pada server dan *user* juga dapat melakukan login ke dalam *virtual machine*.

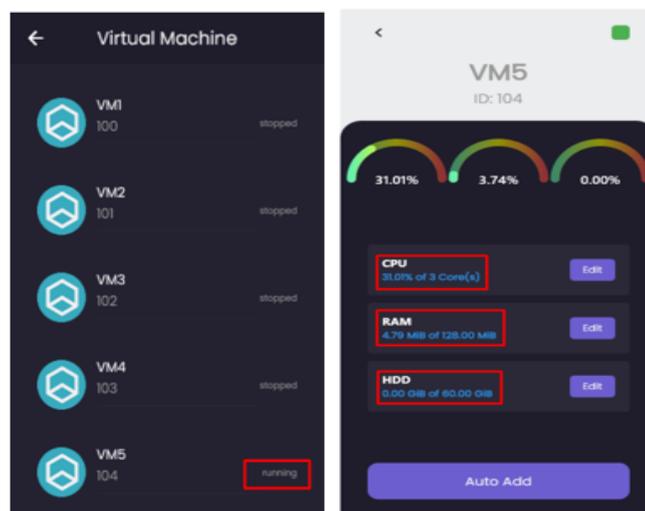


Gambar 5. Halaman Node dan Detil Node pada Aplikasi

Aplikasi mobile berbasis android Manuver dibuat sebagai *interface* yang menampilkan data sumber daya virtual server dalam bentuk grafis. *User* harus melakukan login untuk bisa menggunakan aplikasi ini.

Setelah login, aplikasi akan menampilkan halaman node yang menampilkan list dari node yang tersedia. Dengan meng-klik node yang tersedia, akan diarahkan ke halaman detail node yang menampilkan penggunaan resource secara keseluruhan dari mesin Proxmox. Gambar 5 menunjukkan halaman node dan detil node pada aplikasi.

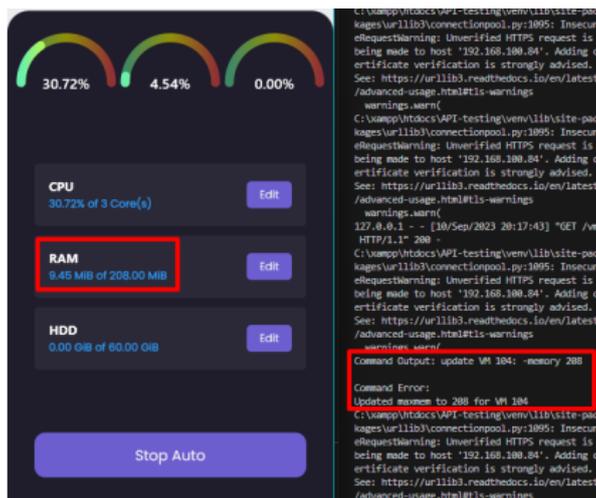
User dapat mengakses virtual machine dengan meng-klik button Virtual Machine pada halaman detail node. Halaman virtual machine berfungsi sebagai halaman yang akan menampilkan list virtual machine dari host seperti terlihat pada Gambar 6. Status virtual machine yaitu *stopped* dan *running*. Saat meng-klik VM5 yang berstatus *running*, dapat dilihat tampilan dari penggunaan sumber daya virtual server yaitu CPU, RAM, dan HDD yang terupdate setiap 10 detik.



Gambar 6. Halaman list dan detail Virtual Machine

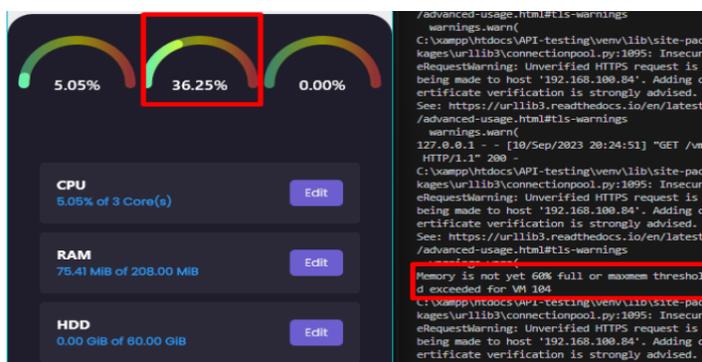
Tombol *Auto Add* pada bagian bawah halaman detail berfungsi untuk melakukan penambahan sumberdaya secara otomatis jika penggunaan sumber daya server melebihi yang sudah ditentukan. Saat tombol *Auto Add* diklik maka *value* yang terkirim pada API pada Nginx berupa nilai *true* yang akan membuat fungsi *start_automation* berjalan.

Selanjutnya pada API akan muncul pesan yang memberitahu bahwa mesin tersebut sedang dipantau setiap 60 detik menit dan proses tersebut berjalan secara terus-menerus. API sudah dikondisikan dengan menentukan batas penggunaannya sampai 60%. Jika penggunaan sumber daya lebih dari 60% maka secara otomatis akan ditambah seperti ditunjukkan pada Gambar 7.



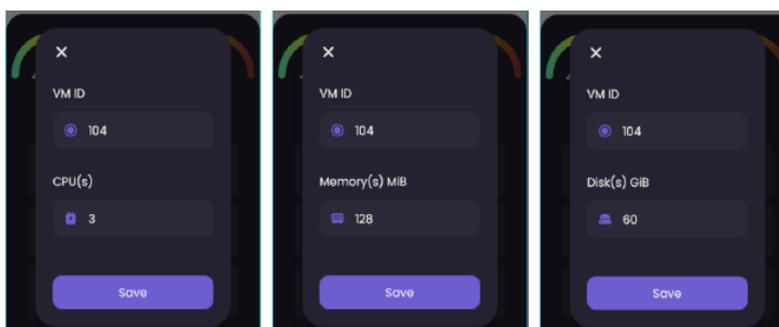
Gambar 7. RAM berhasil di update otomatis

Saat kondisi RAM belum mencapai 60%, maka akan muncul informasi pada terminal seperti yang ditandai pada Gambar 8.



Gambar 8. Saat RAM belum mencapai pemakaian 60%.

Aplikasi Manuver ini juga dapat melakukan penambahan sumber daya secara manual dengan memanfaatkan tombol Edit (Gambar 6) pada jenis sumber daya mana yang ingin ditambahkan. Setelah tombol Edit di klik, maka dapat dilakukan penambahan kapasitas sumber daya yang diinginkan seperti yang ditunjukkan pada Gambar 9.



Gambar 9. Proses penambahan kapasitas sumber daya secara manual

SIMPULAN

Penelitian ini telah menghasilkan aplikasi *mobile* yang diberi nama Manuver yang dapat digunakan untuk mengatur sumber daya virtual server. Dengan aplikasi Manuver ini, administrator dapat memonitoring sumber daya virtual server secara grafis dan *real-time*. Aplikasi Manuver ini juga dapat menambah kapasitas sumber daya virtual server secara otomatis maupun manual. Untuk pengembangannya, Aplikasi Manuver dapat ditambahkan fitur pengurangan kapasitas sumber daya virtual server.

DAFTAR PUSTAKA

- Afrizal, H., & Prihanto, A. (2022). Analisis Kebutuhan Resource Dan Independensi Antara Teknologi Single Server, Virtualisasi Dan Container. *Journal of Informatics and Computer Science (JINACS)*, 4(01), 26–33.
- Armipajasa, G. (2022). Implementasi Monitoring Server dengan Prometheus dan Grafana. *Jurnal Aksara Komputer Terapan*, 11.
- Azizi, M. R., Lutfi, S., & Purwanto, A. (2022). Sistem Monitoring Server berbasis Android di Dinas Komunikasi dan Informatika Kabupaten Rembang. *Journal of Information System and Computer*, 2(1), 53–57.
- Christanto, F. W., & Suprayogi, M. S. (2017). Pemantauan Sumber Daya Virtual Server Pada Cloud Computing Universitas Semarang Menggunakan Network Monitoring System. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 8(2), 629–638.
- Dewi, I., Putri, H., Putra Pratama, H., & Tiara, N. (2021). Framework Design IoT for Smart Agriculture. *Jurnal Sistem Cerdas*, 04(01), 1–8.
- Dwiyatno, S., Rachmat, E., Sari, A. P., & Gustiawan, O. (2020). Implementasi virtualisasi server berbasis docker container. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 7(2), 165–175.
- Prasetyo, E. D., Irawan, J. D., & Ariwibisono, Fx. (2022). Rancang Bangun Sistem Monitoring Server Virtual Berbasis Web menggunakan Script Monitoring pada Proxmox Virtual Environment. *JATI (Jurnal Mahasiswa Teknik Informatika)*.
<https://api.semanticscholar.org/CorpusID:249141512>
- Soni, S., Abdurrahman, A., & Hafid, A. (2019). Optimalisasi Sumber Daya Komputer Dengan Virtualisasi Server Menggunakan Proxmox Ve. *Jurnal Fasilkom*, 9(2), 369–376.
- Suryayusra, M. Izman Herdiansyah, & Dedy Syamsuar. (2022). Perancangan Linux Virtual Server Network Address Translation sebagai Load Balancing untuk Mengurangi Beban Server E-learning pada Universitas Bina Darma. In *Universitas Bina Darma*.