
OPTIMASI SERVER MENGGUNAKAN LOAD BALANCING MICROSERVICE DOCKER PADA BOT TELEGRAM

Oleh

Dini Fakta Sari¹⁾, Deborah Kurniawati²⁾ & Fitria Muriyanto³⁾

^{1,3}Program studi Informatika, STMIK AKAKOM Yogyakarta

²Program studi Sistem Informasi, STMIK AKAKOM Yogyakarta

Jl Raya Janti No 143 Karangjambe Yogyakarta

Email: ¹dini@akakom.ac.id, ²debbie@akakom.ac.id

Abstrak

Server Monitoring adalah sebuah aplikasi untuk memantau seluruh kinerja yang sedang berlangsung pada suatu server. Server monitoring dapat membantu dalam memahami penggunaan sumber daya sistem, seperti CPU, Penggunaan Memory, input dan output, network, penggunaan disk, dan lain sebagainya. Monitoring terhadap performa server merupakan hal yang penting dalam menjaga stabilitas server. Monitoring merupakan tugas seorang administrator dan dalam hal monitoring server tentunya seorang administrator tidak bekerja 24 jam di depan komputer sehingga perlu adanya sebuah layanan aplikasi monitoring server yang dapat memonitor server secara real time dan dapat diakses di mana saja. Aplikasi di bangun dengan teknologi devops di mana serangkaian pekerjaan yang secara otomatis di proses software development dengan source yang kecil dan stabil (Docker) dikombinasikan dengan load balancer untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang. Berdasarkan hasil pengujian, aplikasi dapat menyediakan sebuah layanan untuk memonitor server yang dapat menampilkan grafik kondisi server dan grafik proses yang berjalan pada server secara real time yang terhubung langsung dengan telegram. Analisis kinerja dapat dilakukan dengan menggunakan parameter performance pada processor, memori dan proses layanan.

Kata Kunci : Load balancing, Monitoring Server, Telegram

PENDAHULUAN

Penelitian ini dilatarbelakangi oleh kesadaran dimana pemanfaatan TI (teknologi Informasi) dapat meningkatkan mutu pelayanan dan efisiensi kerja perusahaan. Kompleksitas perusahaan ikut andil dalam kompleksitas infrastruktur IT yang dibangun, salah satunya adalah sever. Server adalah salah satu perangkat teknologi penting untuk memfasilitasi kegiatan perusahaan karena server mempengaruhi kinerja dalam pengelolaan *data base* dan sistem manajemen perusahaan, bahkan untuk kelancaran kinerja perusahaan, server harus selalu *standby* dan terkoneksi dalam kondisi baik selama 24 jam. Permasalahan yang kerap terjadi di dalam sebuah perusahaan atau institusi yang memiliki *server* adalah kurangnya fleksibilitas dalam proses pengawasan dikarenakan

permasalahan *human error* terutama *server admin* yang bertugas mengamati kinerja *server* kurang mampu dalam mengamati *server* selama 24 jam. Salah satu kendala yang dihadapi perusahaan adalah terbatasnya tenaga administrasi server. Selain itu beragamnya permasalahan error server, dan masih pasifnya sistem monitoring server menjadi kendala tersendiri. Dalam banyak kesempatan, perbaikan error server sering terjadi karena adanya laporan dari pengguna pengguna sehingga sering terjadi *complain user*. Tidak adanya pendistribusian pada server (*load balancer*) secara otomatis pada sistem yang mengakibatkan kinerja sistem menjadi berat dan lambat juga menjadi kendala lain bagi perusahaan.

Untuk itu dibutuhkan sistem monitoring server yang lebih

komprehensif, efektif, dan efisien. Membangun sistem server yang berbasis *microserverice* dengan sistem *load balancing* yang bekerja secara terjadwal dan dapat dikendalikan dengan perangkat *mobile* ataupun *desktop* untuk melakukan aktivitas *monitoring server jarak jauh* menjadi salah satu solusi yang diperlukan. Aplikasi *Telegram* dipilih karena aplikasi ini gratis, ringan, dan *multiplatform* serta didukung dengan *Bot API* yang sangat lengkap dan terus berkembang.

LANDASAN TEORI

Load balancing sangat penting ketika skala yang berada di jaringan *computer* semakin besar dan *lalu lintas data* yang ada dalam jaringan *computer* semakin lama lebih tinggi dan lebih tinggi. Dengan *load balancing* memungkinkan untuk mengakses sumber daya internal jaringan dengan mendistribusikannya ke beberapa *host* sehingga tidak terpusat dan kinerja jaringan secara keseluruhan bisa menjadi lebih stabil.

Ketika pengguna mengakses server, server menanggung beban karena harus memproses permintaan dari pengguna. Semakin banyak pengguna mengakses server, semakin banyak proses yang dijalankan server. Jika hanya satu server yang terbebani maka server tidak dapat memberikan layanan kepada banyak pengguna karena kemampuan untuk melakukan pemrosesan terbatas. Solusi ideal adalah membagi beban di beberapa server, sehingga pengguna layanan tidak hanya terkonsentrasi pada satu perangkat. Teknik ini disebut dengan *load balancing* [1]. *Load balancing* mengacu pada pendistribusian beban (tugas kerja) secara merata ke beberapa unit operasi sehingga unit-unit tersebut dapat bekerja sama. *Load balancing* dibangun di atas struktur jaringan asli. Ini menyediakan cara yang transparan, murah dan efisien untuk

memperluas *bandwidth server*, dan perangkat jaringan. Selain itu dapat meningkatkan kemampuan pemrosesan data jaringan dan meningkatkan *throughput*, serta meningkatkan ketersediaan dan fleksibilitas jaringan [2]. *Load balancing* dapat memastikan kehandalan layanan, yang artinya mempercayai sistem untuk terus memberikan layanan terbaik kepada pengguna sehingga memungkinkan pengguna untuk menyelesaikan pekerjaan dengan lancar melalui layanan yang ada. Selain itu, *load balancing* dapat memberikan skalabilitas dan ketersediaan. Jika hanya ada satu server dalam jaringan *computer* maka kemungkinan terjadinya masalah sangat tinggi. Jika server tiba-tiba *crash* maka layanan kepada pengguna akan terganggu. Dengan menambahkan server dan membangun server *farm*, skalabilitas akan meningkat dan faktor ketersediaan juga akan meningkat. *Load balancing* juga digunakan untuk mengurangi waktu eksekusi 2 beban pada *traffic linev* dan memastikan bahwa semua sumber daya yang ada dalam sistem dimanfaatkan secara optimal sehingga dapat memaksimalkan *throughput* [3]. Jenis *load balancing*:

1. *Load balancing Software/hardware*. Solusi penyeimbangan beban *software* adalah satu atau lebih *software* tambahan yang diinstal pada sistem operasi untuk merespon satu atau lebih server untuk *load balancing*, seperti *DNS Load balance*, *checkpoint Firewall-1 Connect Control*, dan lain-lain. Keuntungan jenis ini antara lain konfigurasi yang sederhana, penggunaan yang fleksibel, biaya rendah, dan memenuhi persyaratan umum *load balancing*.
2. *Load balancing local/global*. *Load balancing* dibagi menjadi *load balancing local* dan *load balancing global* dari struktur geografis aplikasi,

yang dikenal juga sebagai load balancing regional. Load balancing local adalah penyeimbangan beban dari beberapa kumpulan server local. Load balancing global melakukan penyeimbangan beban untuk grup server di lokasi geografis yang berbeda dan struktur jaringan yang berbeda [2].

Load balancing memiliki beberapa metode yang dapat digunakan dalam permasalahan pada penggabungan *server*. Namun, beberapa kekurangan pada jaringan adalah terputusnya *server* secara tiba-tiba atau tidak tahu kapan secara pasti akan terputus. Dari hal tersebut membutuhkan *monitoring* secara *real-time* untuk mengetahui ketika *server* terputus. Hal tersebut bisa diatasi dengan menggunakan beberapa aplikasi android yang bersifat *open source* seperti Telegram, Telegram merupakan aplikasi *chat* yang bersifat *open source* yang dapat diaplikasikan pada perangkat lain dengan membuat bot dan memanfaatkan API dari bot tersebut maka akan lebih mudah dalam mengirim informasi jaringan secara *real-time* [4].

Telegram adalah aplikasi perpesanan dengan fokus pada kecepatan dan keamanan, sangat cepat, sederhana, dan gratis. Telegram dapat digunakan di semua perangkat secara bersamaan dan pesan akan disinkronkan dengan mulus di sejumlah ponsel, tablet, atau komputer yang digunakan. Beberapa alasan menggunakan telegram, antara lain [5]:

- i. simple, telegram sederhana dan mudah digunakan
- ii. Privat, telegram terenkripsi dan memiliki fitur penghancur otomatis
- iii. Tersinkron, telegram dapat diakses di beberapa perangkat bersamaan
- iv. Cepat, telegram mengirim pesan lebih cepat dari aplikasi lain
- v. Tangguh, telegram tanpa batas ukuran untuk pesan dan media
- vi. Terbuka, telegram memiliki API dan kode sumber terbuka untuk semua
- vii. Aman, telegram menjaga pesan selalu aman dari serangan peretas

Dikutip dari telegram.org, bot seperti program kecil yang berjalan tepat di dalam Telegram. Mereka dibuat oleh pengembang pihak ketiga menggunakan Telegram Bot API. Bot Telegram merupakan akun khusus yang tidak memerlukan nomor telepon tambahan untuk didaftarkan ke Server Telegram. Akun ini berfungsi sebagai antarmuka antara kode program dengan server Telegram. Telegram merupakan salah satu aplikasi yang mendukung adanya bot ini. Dengan adanya bot ini dapat memudahkan pengguna membuat semacam aplikasi chatting [6].

METODE PENELITIAN

Penelitian ini menggunakan metode PPDIIO (*Prepare, Plan, Design, Implement, Operate, Optimize*). Metode PPDIIO adalah metode yang digunakan oleh Cisco pada perancangann jaringan. Kemudian dalam penelitian ini digunakan dalam perancangan bot server sebagai media remote monitoring yang ditunjukkan pada gambar 1.

Gambar 1. Skema Metodologi PPDIIO (Cisco)



Pembuatan Monitoring Server Menggunakan Bot Telegram Dengan Load Balancing Microservice Docker membutuhkan hardware dan software. Tabel 1 menunjukkan spesifikasi kebutuhan software yang digunakan.

Tabel 1. Spesifikasi kebutuhan software

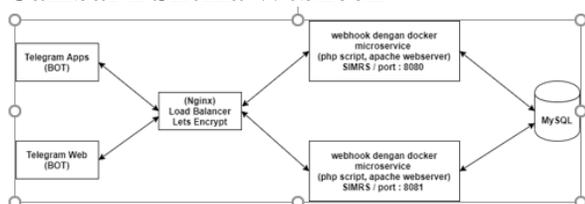
Perangkat Lunak	Versi	Perangkat Lunak	Versi
Ubuntu Server	18.04	Docker	19.03
PHP	7.0.33	Nginx	1.10.3
Apache	2.4.27	Google Chrome	83.0.4
MySQL	5.0.12		

Proses yang akan dilakukan adalah :

- User client telegram harus melakukan register terlebih dahulu untuk dapat mengakses bot monitoring, dengan menginputkan nama dan email address
- Setiap client telegram yang sudah terdaftar dalam database system monitoring, bisa mengakses bot, baik di smartphone maupun web desk
- Setiap user yang sudah terdaftar dalam system bot monitoring akan mendapatkan push notifikasi secara 10 detik di saat server mengalami down/up secara otomatis

Skema pengembangan webhook dapat dilihat Gambar 2. Skema pengembangan webhook pada gambar 2 menjelaskan bagaimana alur kerja dalam mekanisme proses webhook. User di berikan dua pilihan dapat menggunakan desktop maupun mobile, request dari user langsung di teruskan oleh load balancer. Web proxy yang di kelola dengan menggunakan load balancer ini dapat secara otomatis menentukan port mana yang akan di pilih ketika terjadi overload pada sistem (port 8080, port 8081). Setiap proses registrasi akan mendapatkan sebuah chat id dari telegram sendiri. Bot telegram akan menentukan setiap quest dari user dengan berdasarkan chat id masing-masing.

Gambar 2 Skema Webhook



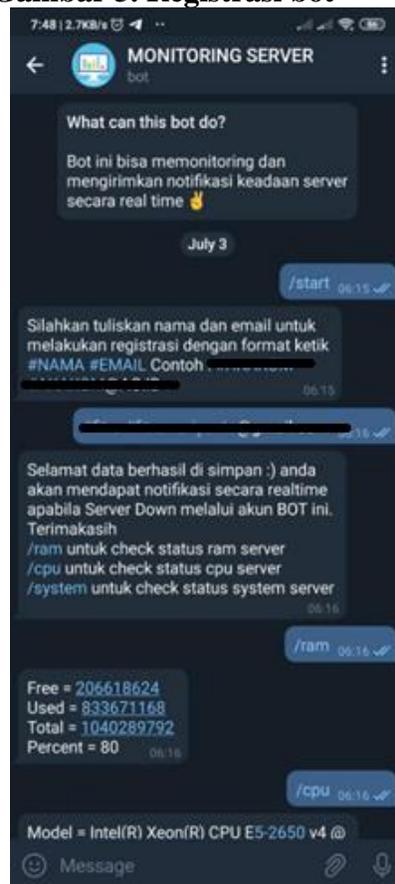
HASIL DAN PEMBAHASAN

Webhook secara umum biasanya dilakukan dengan cara mengupload file engine bot ke sebuah webhosting. Syarat dan ketentuan web hosting yang bisa dipakai adalah, bahwa web tersebut wajib memiliki Domain dan Server yang sudah terkoneksi SSL ber-HTTPS.

Bot ditanam pada sebuah URL, jadi ketika ada pesan masuk, pesan tersebut akan diposting atau diteruskan ke URL yang telah

di setting untuk selanjutnya diproses oleh bot. Tentunya untuk menggunakan metode ini harus mempunyai layanan VPS atau web service untuk tempat project bot. Untuk dapat terkoneksi dengan bot, Langkah pertama yang harus di lakukan adalah dengan mencari bot pada kolom pencarian telegram di smartphone. apabila sudah mendapati bot tersebut dapat di lanjutkan dengan klik tombol start. Gambar 3 menunjukkan registrasi bot yang dilakukan. Melanjutkan dari isi perintah *start*, bot akan mengarahkan untuk mengisikan nama dan email agar terdaftar sebagai bot member, yang berfungsi apabila server mengalami down pada salah satu server, bot akan mengirimkan notifikasi pada akun telegram yang sudah terdaftar. Data member user akan tersimpan sebagai chat id pada database, yang berfungsi menyimpan semua record yang nanti nya akan di kembalikan sebagai proses reply apabila terdapat request dari bot tersebut.

Gambar 3. Registrasi bot



Analisis kinerja dapat dilakukan dengan menggunakan parameter performance pada processor, memori dan proses layanan. Bot diatur untuk dapat mengakses server baremetal secara langsung dengan beberapa syntax sebagai berikut,

- a. / **ram** untuk mengecek status ram yang di gunakan pada server
- b. / **cpu** untuk mengecek status informasi detail processor yang di gunakan pada server
- c. / **system** untuk mengecek status system yang terinstall pada server

Contoh informasi baremetal pada aplikasi telegram dapat dilihat pada Gambar 4.

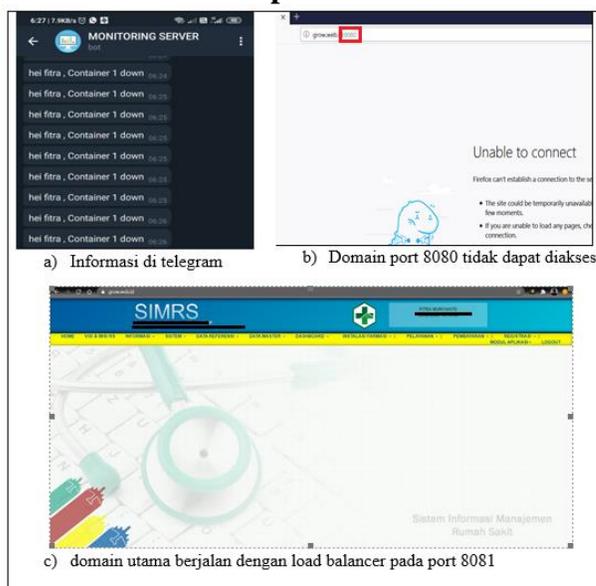
Gambar 4. Status server dalam telegram



Ujicoba dilakukan dengan memberikan beban request pada single container. Pada saat servis container dengan port 80 pada server

dihentikan maka secara otomatis akun telegram akan mendapatkan pesan otomatis dari bot, memberitahukan bahwa container 1 mengalami down berkala setiap 10 detik sekali. Domain dengan port 8080 tidak dapat di akses, dan di lirikan ke port yg lain secara otomatis oleh loadbalancer. Penggambaran kondisi ini dapat dilihat pada Gambar 5. Ketika port 8080 mengalami down, domain utama tetap berjalan dengan load balancer pada port 8081. Hal yang sama juga alan terjadi pada saat service Container 2 dihentikan. Secara otomatis akun telegram akan mendapatkan pesan otomatis dari bot, memberitahukan bahwa container 2 mengalami down berkala setiap 10 detik sekali. Domain dengan port 8081 tidak dapat di akses, dan di lirikan ke port 8080 secara otomatis melalui loadbalancer. Ketika port 8081 mengalami down, domain utama tetap berjalan dengan load balancer pada port 8080.

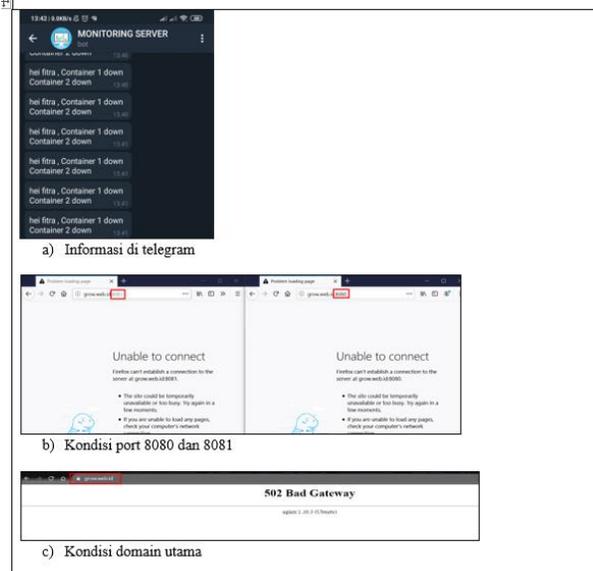
Gambar 5. Kondisi port 8080 dihentikan



Ketika semua port mengalami down baik port 8080 dan 8081, bot akan mengirimkan notifikasi setiap 10 detik mengindikasikan bahwa kedua container mengalami down dan tidak dapat di akses. Domain utama memberikan report bad gateway 502, yang berarti server awal

mengirimkan respons tidak valid ke server lainnya yang bertugas sebagai gateway atau proxy. Kondisi ini dapat dilihat pada Gambar 6.

Gambar 6. Kondisi saat semua port down



Setelah dilakukan uji coba aplikasi yang dibangun dapat membantu sysadmin dalam mendapatkan informasi seputar server dengan cepat, dimana dan kapan saja. Dapat membantu sysadmin dalam memajemen sistem server sesuai dengan image yang di build pada microservice docker guna mengoptimalkan server. Kelebihan monitoring server menggunakan telegram berbasis load balancer sendiri yaitu terjadwal rapi melalui sistem scheduler untuk pembagian overload sistem berdasarkan port, dan dapat memberikan notifikasi akurat dengan realtime melalui push notifikasi telegram maupun email bagi user yang sudah terdaftar dalam sistem. Selain itu, kelebihan lainnya adalah memberikan ruang space lebih luas pada server untuk mengelola data dengan adanya microservice docker, karena docker tidak membutuhkan sumber komputasi yang besar. Fungsi lain daripada Docker container juga dapat digunakan untuk membangun, mempersiapkan dan menjalankan aplikasi, dapat membuat aplikasi dari bahasa pemrograman yang berbeda pada lapisan apapun. Aplikasi dapat di bungkus dalam

container, dan aplikasi dapat berjalan pada lingkungan apapun dimana saja. Aplikasi monitoring ini mengimplementasikan Load Balancer. Sebagai catatan khusus, Pembatasan hak akses terhadap bot perlu di lakukan agar tidak di salah digunakan dalam melakukan monitoring server. Pembagian user biasa dan admin di bedakan ketika melakukan pendaftaran dengan memberi tagar admin pada akhir session.

PENUTUP

Kesimpulan

Berdasarkan hasil pengumpulan dan analisis data serta proses pembuatan aplikasi, maka dapat diambil kesimpulan sebagai berikut:

1. Memanfaatkan pesan instan sebagai alternatif yang efisien untuk membantu sysadmin dalam mendapatkan informasi seputar server dimana dan kapan saja.
2. Dengan menggunakan aplikasi sistem informasi berbasis Bot Telegram, Sysadmin terbantu dalam mendapatkan informasi server secara cepat dan akurat.
3. Dengan menerapkan loadbalancer pada docker container, sysadmin terbantu dalam proses waktu maintenance system. Memberikan ruang waktu dalam proses pembenahan sistem yang overload.
4. Dengan menggunakan docker container sistem berjalan lebih ringan di bandingkan menggunakan multi virtual machine.

DAFTAR PUSTAKA

- [1] Rijayana, I., 2005, Teknologi Load Balancing untuk Mengatasi Beban Server, *Seminar Nasional Aplikasi Teknologi Informasi 2005*, Yogyakarta, 18 Juni 2005.
- [2] Liang, S., Cheng, J., Zhang, J., Research on Data Load Balancing Technology of Massive Storage System for Wearable Devices, 2002, *Digital Communications and Networks*, November 2020.
- [3] Mustofa, A., Ramayanti, D., 2020, Implementasi Load Balancing dan

-
- Failover to Device Mikrotik Router menggunakan Metode NTH (Studi Kasus : PT. Gojek Indonesia), *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, Vol 7 , No. 1, Februari 2020, hlm 139-144.
- [4] Oktaviantom M. I., dan Prayogi, Y. R., 2019, Sistem Monitoring Jaringan *Load balancing* dengan Metode *Equal Cost Multipath* (ECMP) menggunakan Media Telegram, *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, Volume 4, Nomor 2, Desember 2019.
- [5] Telegram.org.
- [6] Hakim, D. K., dan Nugroho, S. A., 2019. Implementasi Telegram Bot untuk Monitoring Mikrotik Router, *Jurnal Sainteks*, Vol 16, No 2..

HALAMAN INI SENGAJA DIKOSONGKAN