

PENGEMBANGAN KOMUNIKASI MULTIKANAL UNTUK MONITORING INFRASTRUKTUR JARINGAN BERBASIS BOT TELEGRAM

Rio Juniyantara Putra¹, Nyoman Putra Sastra², Dewa Made Wiharta³

Jurusan Teknik Elektro, Fakultas Teknik Universitas Udayana

Email: ryojuniyantara@gmail.com¹, putra.sastra@unud.ac.id², wiharta@unud.ac.id³

Abstrak

Penelitian ini mengusulkan suatu solusi untuk mengoptimalkan sistem monitor infrastruktur jaringan di Universitas Udayana. Optimasi sistem monitor dimulai dari pencarian segala informasi tentang perangkat jaringan internet pada server OpenNMS dan Router Mikrotik. Layanan ini dilakukan dengan menggunakan aplikasi Telegram. Pengguna dapat melakukan komunikasi dengan Bot Telegram yang sudah diintegrasikan dengan sistem monitoring agar dapat memberikan informasi tentang jaringan internet di Universitas Udayana. Aplikasi Telegram dipilih karena aplikasi ini gratis, ringan, dan multiplatform serta didukung dengan Bot API yang sangat lengkap dan terus berkembang. Segala bentuk respon sudah dirancang dinamis dengan memanfaatkan sumber daya yang ada, dan sudah dibuat dalam penelitian ini sebagai prototype dengan beberapa perintah.

Kata Kunci : Bot, Sistem Monitoring, Telegram,

1. PENDAHULUAN

Monitoring infrastruktur jaringan merupakan suatu hal yang sangat penting untuk menjaga kelangsungan *Service Level Agreement* (SLA) [1] dari sebuah layanan. Agar mudah, dan efektif dan efisien, maka berbagai model telah ditawarkan oleh berbagai pengembang aplikasi/layanan. dapat tetap terjaga oleh sebuah sistem yang digunakan untuk memantau kondisi perangkat jaringan baik itu *server*, *router*, maupun perangkat jaringan lainnya yang terhubung ke jaringan. Sistem monitoring ini ditargetkan agar sistem dapat mengakses *resource* dan *service* yang ditangani perangkat jaringan, dapat memberikan informasi kepada network administrator bila terjadi masalah dalam jaringan.

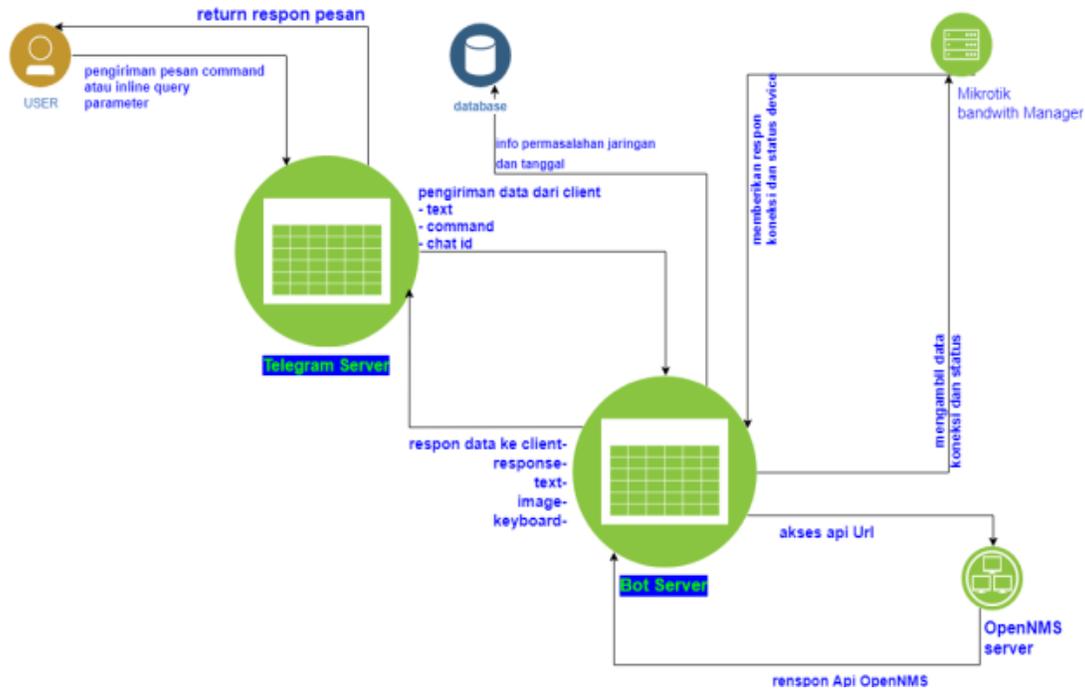
Dalam memantau sebuah jaringan komputer dibutuhkan seorang *network administrator* yang bertanggung jawab atas pengelolaan jaringan internet. Salah satu tanggung jawabnya yaitu memantau jaringan komputer agar tetap berada pada keadaan optimal. Dalam memonitoring suatu jaringan komputer diperlukan ketelitian, agar informasi

yang didapatkan dari hasil monitoring sesuai dengan kondisi sebenarnya.

Sistem monitoring infrastruktur jaringan pada Universitas Udayana saat ini masih menggunakan sistem monitoring berbasis web, OpenNMS itu sendiri merupakan merupakan aplikasi manajemen jaringan berbasis *free/open source*[2], dan terintegrasi dengan banyak perangkat jaringan. Network administrator sangat sering menerima keluhan dari pengguna karena jaringan yang bermasalah, dan setiap terjadi permasalahan mengharuskan *network administrator* membuka aplikasi web monitoring untuk mencari informasi terkait permasalahan yang terjadi dan mengharuskan menggunakan jaringan intranet.

Melihat permasalahan ini, maka dalam penelitian ini ditawarkan sebuah solusi untuk pemberian alert dan pesan menggunakan layanan Telegram Bot, yang akan memberikan *custom tool* dan *social service* untuk network administrator yang diintegrasikan dengan sistem monitoring infrastruktur jaringan di Universitas Udayana. Layanan yang disediakan berupa pencarian

informasi infrastruktur jaringan seperti server, router, dan switch.



Gambar 1 Arsitektur Sistem

Layanan ini akan dilakukan dengan aplikasi *messenger* Telegram. *Network administrator* akan berkomunikasi dengan Bot Telegram yang dirancang sesuai dengan kebutuhan agar memberikan informasi perangkat dan permasalahan lainnya.

Telegram Bot merupakan aplikasi pihak ketiga yang berjalan didalam sistem Telegram [3], dan merupakan salah satu dari layanan yang disediakan oleh aplikasi Telegram. Telegram dipilih karena aplikasi ringan, gratis, multiplatform, dan juga memiliki BOT API yang lengkap dan terus berkembang, sehingga memungkinkan untuk membuat Bot yang dinamis, pintar dan dapat merespon pesan dari *network administrator*. Bot dirancang memberikan respon yang dinamis dengan memanfaatkan sumber daya dari infrastruktur jaringan Universitas Udayana. Tentunya diharapkan mampu memberikan kemudahan bagi *network administrator* untuk mengakses informasi tentang infrastruktur jaringan maupun permasalahan yang terjadi dengan mudah.

2. DESAIN SISTEM

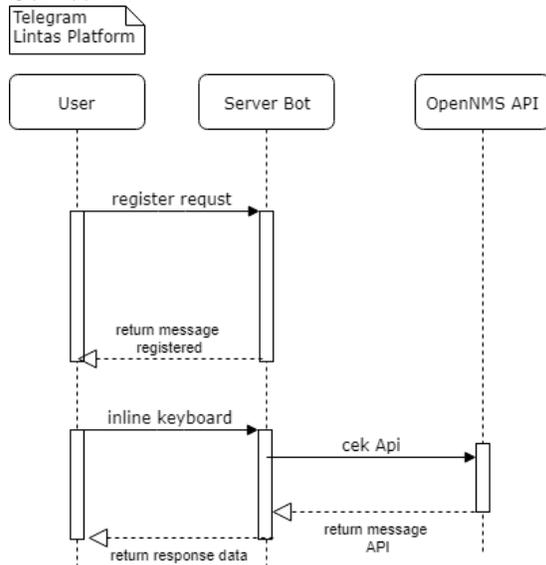
Penelitian ini memanfaatkan fitur Bot dari Telegram yang nantinya digunakan untuk merespon pesan dari *network administrator*. Arsitektur dari Telegram Bot dapat dilihat pada Gambar 1. *Network administrator* mengirim pesan dari perangkat yang digunakan ke akun Telegram Bot. Pesan akan diterima oleh *Telegram server* dan dilanjutkan ke *Bot server*. *Bot server* akan memproses pesan tersebut untuk dapat merespon *request* yang dikirimkan oleh *network administrator* berupa *command* atau *inline query*. Respon akan dikirimkan kembali ke *client* melalui *Telegram server*. Setiap teks yang dikirimkan akan bertindak sebagai *command/perintah* yang berpengaruh terhadap bentuk respon ke *client*.

Untuk menjalankan layanan yang disediakan bot, pengguna harus mengirimkan beberapa perintah yang sudah disediakan. Contoh beberapa perintah sebagai beri

1. /start - perintah pertama yang akan dikirimkan pengguna jika pertama kali menggunakan layanan bot.
2. /opennms - perintah untuk menampilkan menu dari layanan OpenNMS.
3. /mikrotik - perintah ini digunakan untuk menampilkan menu dari layanan mikrotik.
4. ./reset – perintah ini digunakan untuk mereset tombol dari tombol menu.

2.1 Sequence Diagram Akses Data API OpenNMS

Pada bagian ini jenis layanan yang disediakan oleh Bot server adalah pengaksesan API OpenNMS dengan menjalankan perintah command atau inline query, dan selanjutnya akan mengambil informasi dari respon API tersebut. Setelah itu data akan dilempar ke aplikasi Telegram client dalam kasus ini yang berperan sebagai network administrator. Rancangan sequence diagram akses API server ditunjukkan pada Gambar 2.

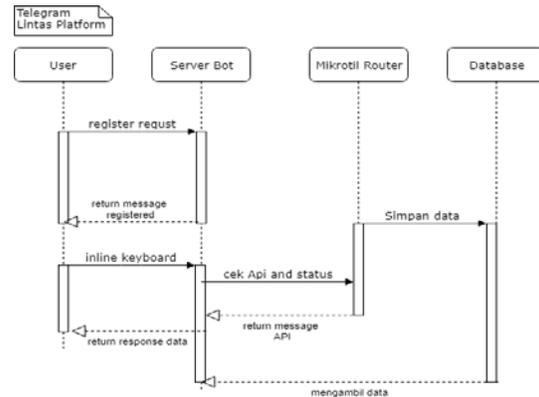


Gambar 2 Sequence Akses Data API OpenNMS

2.2 Sequence Diagram Akses API Router Mikrotik

Pada bagian ini jenis layanan yang disediakan oleh Bot server adalah mengakses api mikrotik untuk mengambil informasi tentang informasi sistem pada router Mikrotik dan status koneksi dengan menjalankan perintah command atau inline

query, dan menyimpan informasi tersebut ke dalam database. Selanjutnya client/network administrator akan mendapatkan informasi dari respon API tersebut melalui client Telegram yang dimiliki. Rancangan sequence diagram akses data api mikrotik ditunjukkan pada Gambar 3.



Gambar 3 Sequence Akses Data API Mikrotik dan database

2.3 Referensi

2.3.1 NMS

Network Monitoring System adalah sistem ekstra atau kumpulan dari beberapa sistem yang memiliki tugas untuk mengamati atau memonitor sistem-sistem jika kemungkinan terjadinya masalah-masalah pada sistem tersebut agar dapat dideteksi secara dini.[4]

2.3.2 DBMS

DBMS adalah Kumpulan file yang saling berkaitan satu dengan yang lainnya bersama dengan program untuk pengelolaannya. Satu Database Management System (DBMS) memiliki satu koleksi data yang saling berelasi dan satu program untuk mengakses data dari koleksi data tersebut. DBMS terdiri dari beberapa database dan set program pengelola untuk melakukan kegiatan seperti menambah data, menghapus data, mengambil dan membaca data.[5]

2.3.3 Node JS

Node.js adalah sistem perangkat lunak yang didesain untuk keperluan pengembangan aplikasi web. Aplikasi ini ditulis dalam bahasa JavaScript maupun Typescript, menggunakan basis event dan asynchronous I/O. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada

peramban, Node.js merupakan bahasa pemrograman *server side* yang dieksekusi sebagai aplikasi *server*. Aplikasi ini dibangun dari V8 JavaScript Engine buatan Google dan beberapa modul bawaan yang sudah terintegrasi.[6]

2.3.4 RESTful API

RESTful API merupakan salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* dari *user request* suatu halaman web, maka *server* akan mengirimkan *state* halaman web yang sekarang ke *browser*. Bernavigasi melalui *link endpoint* yang disediakan sama halnya dengan mengganti *state* dari halaman web. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas yang sesuai, seolah-olah terjadi perpindahan *state* satu sama lain.[7]

3 HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian

Pengujian Telegram Bot pada penelitian ini dilakukan dengan menggunakan metode black box dengan mencoba semua perintah yang sudah diatur dalam Bot dengan menyesuaikan dengan fungsinya masing-masing. Beberapa contoh *screenshot* pengujian Telegram Bot dapat dilihat pada Gambar 9.

No	Nama Pengujian	Bagian Yang Dujai	Tindakan Pengujian	Jenis Pengujian	Teknik Pengujian	Hasil Pengujian
1	Perintah memulai	Perintah /start	Klik Perintah /start	Sistem	Black-Box	Berhasil
2	Perintah menu OpenNMS	Perintah /opennms	Klik Perintah /opennms	Sistem	Black-Box	Berhasil
3	Perintah menu Mikrotik	Perintah /mikrotik	Klik Perintah /mikrotik	Sistem	Black-Box	Berhasil
4	Perintah menu OpenNMS node	Perintah /nodes	Klik perintah /nodes	Sistem	Black-Box	Berhasil
5	Perintah menu OpenNMS event	Perintah /events	Klik perintah /event	Sistem	Black-Box	Berhasil
6	Perintah menu OpenNMS alarm	Perintah /alarms	Klik Perintah /alarms	Sistem	Black-Box	Berhasil
7	Perintah menu Mikrotik ip address	Perintah /mikroipaddress	Klik Perintah /mikroipaddress	Sistem	Black-Box	Berhasil
8	Perintah menu Mikrotik dhcp server	Perintah /mikrodhcpserver	Klik Perintah /mikrodhcpserver	Sistem	Black-Box	Berhasil
9	Perintah menu Mikrotik dhcp client	Perintah /mikrodhcpclient	Klik Perintah /mikrodhcpclient	Sistem	Black-Box	Berhasil
10	Perintah menu Mikrotik dns	Perintah /mikrodns	Klik Perintah /mikrodns	Sistem	Black-Box	Berhasil
11	Perintah menu Mikrotik ip pool	Perintah /mikroippool	Klik Perintah /mikroippool	Sistem	Black-Box	Berhasil
12	Perintah menu Mikrotik ip route	Perintah /mikroiproute	Klik Perintah /mikroiproute	Sistem	Black-Box	Berhasil
13	Perintah menu Mikrotik queue simple	Perintah /mikroqueueimple	Klik Perintah /mikroqueueimple	Sistem	Black-Box	Berhasil
14	Perintah menu Mikrotik queue tree	Perintah /mikroqueuetree	Klik Perintah /mikroqueuetree	Sistem	Black-Box	Berhasil
15	Perintah menu Mikrotik ip queue interface	Perintah /mikroqueueinterface	Klik Perintah /mikroqueueinterface	Sistem	Black-Box	Berhasil
16	Perintah menu Mikrotik router board	Perintah /mikrorouterboard	Klik Perintah /mikrorouterboard	Sistem	Black-Box	Berhasil

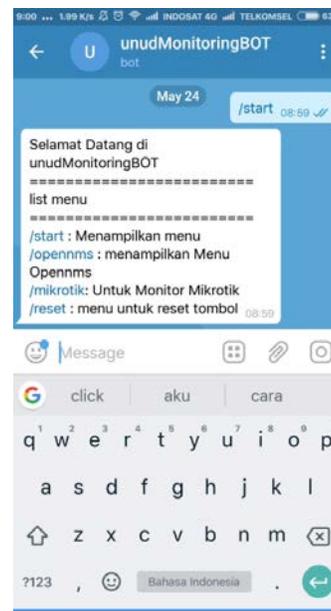
Gambar 4 Hasil Pengujian

3.2 Pembahasan dan Gambaran Umum Sistem Keseluruhan

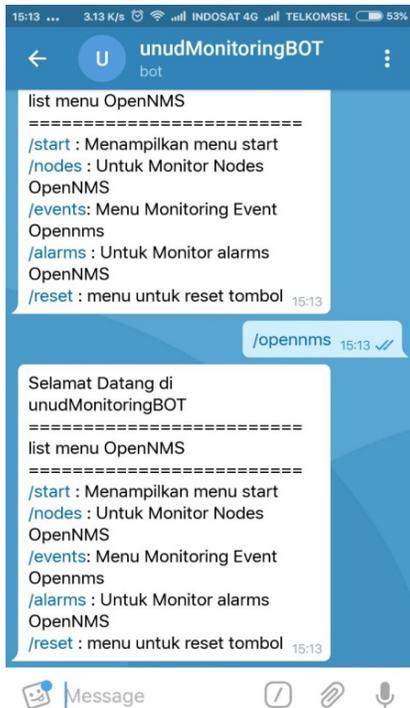
Aplikasi Bot dibangun dengan menggunakan bahasa pemrograman JavaScript pada platform Node JS. Komunikasi dari *server* ke *Bot server* menggunakan metode *long polling*. Beberapa contoh *screenshot* penggunaan Bot oleh *network administrator* dapat dilihat pada Gambar 5 sampai dengan Gambar 9.



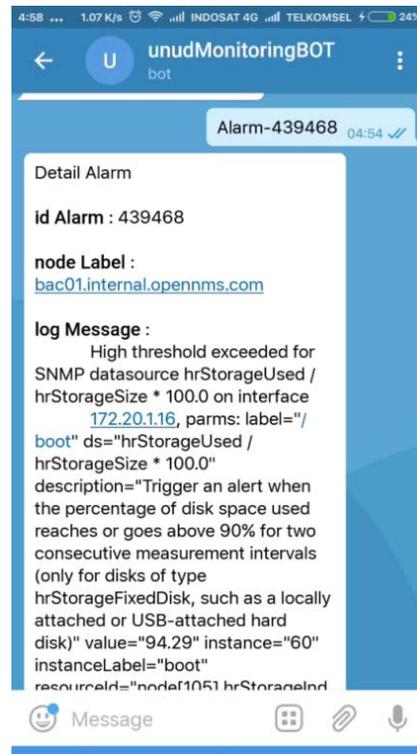
Gambar 5 Daftar Perintah



Gambar 6 Menu start



Gambar 7 Menu OpenNMS



Gambar 9 Detail Alarm OpenNMS



Gambar 8 Daftar Alarm OpenNMS

4 SIMPULAN

4.1 SIMPULAN

Simpulan yang dapat diambil berdasarkan hasil dari pengujian dan pembahasan yang telah dilakukan adalah sebagai berikut:

Aplikasi ini dirancang dengan menggunakan Bot Telegram dan diintegrasikan dengan server OpenNMS dan Router Mikrotik, sehingga pengguna dapat memonitoring server maupun perangkat jaringan lainnya dengan mudah. Bot Telegram untuk layanan monitoring telah berhasil dibuat, terdapat beberapa *command* yang dibuat dalam penelitian ini sebagai *prototype* awal. *Output* yang diberikan oleh Bot berupa teks dan tombol. Tidak menutup kemungkinan untuk dapat memberikan *output* berupa dokumen, gambar, audio, video dan format lainnya.

Untuk mengintegrasikan API OpenNMS dan Router Mikrotik dengan *backend system* Telegram yaitu dengan merancang sistem yang bisa mengakses informasi data OpenNMS dan Router Mikrotik dan

mengintegrasikan sistem tersebut dengan Bot Telegram.

4.2. SARAN

Aplikasi monitoring ini dapat dikembangkan agar dapat dijalankan di media sosial lainnya seperti Line Bot, Messenger Bot, dan lain - lainnya.

Pengujian pada penelitian ini terbatas hanya dengan beberapa pengguna. Diharapkan untuk pengujian lebih lanjut bisa menggunakan pengguna yang lebih banyak.

Aplikasi monitoring terbatas hanya diintegrasikan dengan OpenNMS dan Router Mikrotik, diharapkan untuk kedepannya dapat diintegrasikan dengan sistem monitoring lainnya seperti cacti, zabbix dan lainnya.

5 DAFTAR PUSTAKA

- [1] Waspada I, Analisa Best Practice Service Level Management (SLM) CISCO Menggunakan Kriteria Kelengkapan Dari Thomas SCHAAF, Jurnal Masyarakat Informatika, 2010;2:1.
- [2] Rusmana M U, Pengembangan Aplikasi Sistem Monitoring Jaringan Menggunakan OpenNMS Berbasis *Smartphone* Android, Jurnal Ilmiah Komputer dan Informatika, 2016;1:2.
- [3] <https://core.telegram.org/bots>, diakses tanggal 15 Agustus 2018.
- [4] Amnur H,Deddy, Perancangan dan Implementasi Network Monitoring Sistem Menggunakan Nagios dengan Email dan SMS Alert, *Poli Rekayasa*, 2014;10:1.
- [5] Kristanto H, Konsep dan Perancangan Database, II, Yogyakarta, ANDI, 2009
- [6] Iqbal M, Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan *Node.js*, *Jurnal Teknik ITS*, 2012;1:A-243.
- [7] Rismayani, Sistem Manajemen Administrasi Kependudukan Tingkat Pedesaan Berbasis RESTful API di Kabupaten Bone, Jurnal Pekommas, 2017;2:9.