

SISTEM KONTROL DAN PEMBERITAHUAN STATUS PELADEN JARAK JAUH BERBASIS CHATBOT

Dian Hanifudin Subhi¹, Aldi Wirawan², Syehfi Amali³
Jurusan Teknologi Informasi, Politeknik Negeri Malang
email: ¹dhanifudin@polinema.ac.id, ²aldirwira@gmail.com, ³syehfi75@gmail.com

(Artikel diterima: Desember 2020, diterima untuk terbit: Januari 2021)

Abstrak – Sistem kontrol dan pemberitahuan status peladen jarak jauh berbasis *chatbot* merupakan pemanfaatan Telegram Messenger untuk melakukan kontrol (*remote server*) pada peladen. Kontrol dapat dilakukan pada lingkungan peladen dengan jaringan lokal (intranet) tanpa perlu melakukan pengaturan VPN (*Virtual Private Network*). Pemberitahuan status peladen dikirimkan secara terjadwal setiap 30 menit sekali kepada pengguna. Pengujian usabilitas dilakukan untuk menguji pendekatan ini sehingga dapat menjadi alternatif solusi bagi praktisi jaringan dalam mengelola peladen. Dalam pemanfaatan penelitian ini didapatkan hasil sebesar 86,41% dari sisi usabilitas penggunaan.

Kata kunci: *remote server, chatbot, telegram, peladen.*

Abstract – The chatbot-based remote server status control and notification system is the use of Telegram Messenger to control (*remote server*) the server. Control can be carried out on a server environment with a local network (intranet) without the need to set up a VPN (*Virtual Private Network*). Server status notifications are sent on a scheduled basis every 30 minutes to users. Usability testing is carried out to test this approach so that it can be an alternative solution for network practitioners in managing servers. In the utilization of this research, the results obtained are 86.41% in terms of usability of use.

Keywords: remote server, chatbot, telegram, server.

I. PENDAHULUAN

Administrator Sistem (*System Administrator*) atau Admin harus dapat memastikan peladen yang dikelola dapat bekerja dengan baik dan maksimal. Informasi berupa keadaan peladen seperti penggunaan CPU, RAM, penggunaan penyimpanan dan layanan-layanan yang berjalan harus dengan cepat didapatkan. Tetapi sebuah peladen tidak selalu dapat diakses dengan mudah, dikarenakan terkadang ada keterbatasan hanya dapat diakses melalui intranet atau VPN (*Virtual Private Network*).

Dengan melihat kendala tersebut maka diperlukan suatu solusi untuk layanan tambahan untuk memudahkan komunikasi antara Admin dengan peladen. Sehingga peladen dapat dengan mudah dikelola walaupun Admin berada pada lokasi yang berbeda dengan akses yang terbatas. Sebagai alternatif, sistem kontrol jarak jauh serta pemberitahuan berbasis chatbot ditawarkan untuk menyelesaikan masalah ini. Penelitian ini berfokus pada bagaimana pengembangan sistem kontrol dan pemberitahuan status peladen berbasis *chatbot* pada *Telegram Messenger* yang terbatas pada satu peladen.

II. TINJAUAN PUSTAKA

A. Remote Server

Remote Server adalah metode untuk mengakses komputer atau peladen dari jarak jauh dengan beberapa metode. *Remote server* dapat dikendalikan dari jarak tertentu dengan menggunakan media kendali jaringan lokal maupun internet baik kabel (*wire*) maupun non-kabel (*wireless*). Subyek dan obyek kendali biasanya perangkat elektronik atau PC/Smartphone, dengan sistem multiplatform (windows, linux, mac, atau *mobile*) [1].

Remote server digunakan dalam melakukan pengendalian peladen secara jarak jauh oleh *System Administrator* untuk melakukan pemantauan peladen untuk melihat apakah terdapat kesalahan pada peladen, misalnya pada peladen terdapat proses yang tidak berjalan dengan normal. Pada umumnya *remote server* menggunakan protokol SSH (*Secure Shell Protocol*) untuk melakukan koneksi antara pengguna dengan server yang akan di kontrol secara jarak jauh oleh *System Administrator*.



Gambar 1. Arsitektur telegram bot

B. Telegram Messenger

Telegram Messenger adalah aplikasi *messenger smartphone* yang bertujuan untuk mengirim dan menerima pesan teks dan multimedia ke dan dari pengguna pribadi. Ada banyak aplikasi *messenger*, namun yang membedakannya adalah fitur keamanan melalui enkripsi data dan kemampuan membuat Telegram Bots dengan beberapa fungsi melalui kode pemrograman menggunakan Telegram Bot API [2].

Telegram Messenger dipilih karena merupakan aplikasi yang mudah digunakan, gratis dan dapat digunakan di banyak platform seperti Android, IOS, Windows, MacOS, Linux, dan Web Desktop. Pengguna dapat mengirimkan pesan singkat seperti foto, video, stiker, pesan suara kepada user lain. *Telegram Messenger* menyediakan fasilitas pengembangan chatbot yang cukup lengkap. Selain itu juga terdapat fasilitas untuk membangun desain antar muka berupa tombol, tautan, perintah dan lain-lain.

C. Telegram Bot

Telegram Bot dibuat untuk memiliki fungsi yang sama dengan pengguna pada umumnya, tetapi akan dikendalikan oleh mesin dan dapat memiliki fitur kecerdasan buatan [2].

Dari Gambar 1 merupakan alur secara garis besar dalam telegram bot, Smartphone akan mengirimkan request pesan lalu akan diproses oleh telegram bot, lalu telegram bot akan memproses perintah tersebut ke server untuk diproses, lalu server akan mengirimkan hasil dari perintah tersebut dan akan dikirimkan ke Telegram Bot, lalu bot akan mengirimkan kepada user yang mengirimkan request tersebut.

Ada dua cara dalam pembuatan Bot Telegram yaitu dengan menggunakan metode long-polling dan webhook. Dengan metode long-polling, maka server akan mengecek secara periodik ke bot apakah ada pesan yang masuk. Jika ada pesan yang masuk maka server akan melakukan eksekusi berdasarkan pesan request yang dikirim pengguna, jika tidak ada pesan maka kondisi serve idle. Dalam metode Long-Polling Bot akan dijalankan langsung melalui server, sedangkan dalam metode Webhook bot akan berada dalam hosting dan wajib menggunakan https [3], [4].

D. Node.js

Node.js didesain untuk membangun aplikasi jaringan yang dapat berskalakan kecil atau besar [4]. Node.js adalah sebuah platform baru yang menarik yang dikembangkan oleh Ryan Dahl, yang memungkinkan pengembang bahasa pemrograman JavaScript untuk membuat server dengan kinerja yang sangat tinggi dengan memanfaatkan mesin V8 JavaScript Google, dan asynchronous I/O [5].

Node.js digunakan dalam pembuatan layanan bot untuk dijalankan pada peladen yang akan digunakan untuk melakukan kontrol jarak jauh dengan pemanfaatan Telegram Bot API, agar peladen dapat mengetahui terdapat perintah masuk apa saja yang dapat dilakukan oleh peladen. Node.js dapat digunakan sebagai runtime javascript berbasis *Asynchronous event-driven*, Node.js dirancang untuk membangun aplikasi jaringan.

E. SQLite

SQLite merupakan paket perangkat lunak yang bersifat *public domain* yang menyediakan sistem manajemen basis data relasional atau RDBMS. Sistem basis data relasional

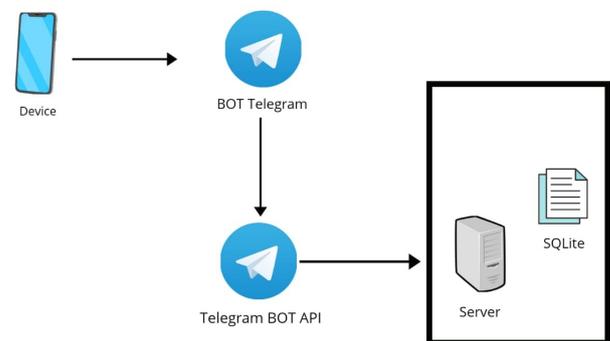
digunakan untuk menyimpan record yang didefinisikan oleh pengguna pada ukuran tabel yang besar dan memproses perintah query yang kompleks dan menggabungkan data dari berbagai tabel untuk menghasilkan laporan dan rangkuman data [6].

SQLite digunakan dalam pembuatan layanan bot untuk menyimpan data pengguna, kata perintah yang akan diblokir, dan pengguna untuk penerima notifikasi pemberitahuan status server [7] – [10].

III. METODE PENELITIAN

A. Arsitektur dan Desain Sistem

Admin mengelola peladen melalui gawai pintar (*mobile*). Pada gawai yang telah terpasang aplikasi *Telegram Messenger*, Admin mengirimkan pesan kepada Bot Telegram yang mendukung perintah-perintah yang telah didefinisikan. Perintah ini yang kemudian akan diteruskan ke peladen (*server*) melalui Telegram Bot API. Peladen mencocokkan perintah, apakah termasuk dalam daftar perintah yang diblokir pada SQLite. Balasan pesan kemudian akan disampaikan kembali ke Admin. Penjelasan alur arsitektur sistem, dapat dijabarkan pada Gambar 2.



Gambar 2. Arsitektur sistem kontrol jarak jauh

Sistem pemberitahuan status pada peladen akan dijadwalkan setiap 30 menit sekali. Penjadwalan akan memanfaatkan format pola cron yang disisipkan ke dalam program Node.js. Pemberitahuan hanya akan dikirimkan kepada pengguna yang telah melakukan *subscription* atau langganan.

Dalam penerapan sistem, dibutuhkan penyimpanan data ke dalam basis data. Dibutuhkan tiga buah tabel yaitu *users*, *subscribe* dan *blacklist*. Pada tabel *users* berisi data pengguna sistem. Untuk tabel *subscribe* berisi data pengguna yang dapat menerima pemberitahuan sistem. Sedangkan *blacklist* berisi data perintah-perintah yang diblokir. Gambaran lebih jelas, dapat dilihat pada Gambar 3.

IV. HASIL DAN PEMBAHASAN

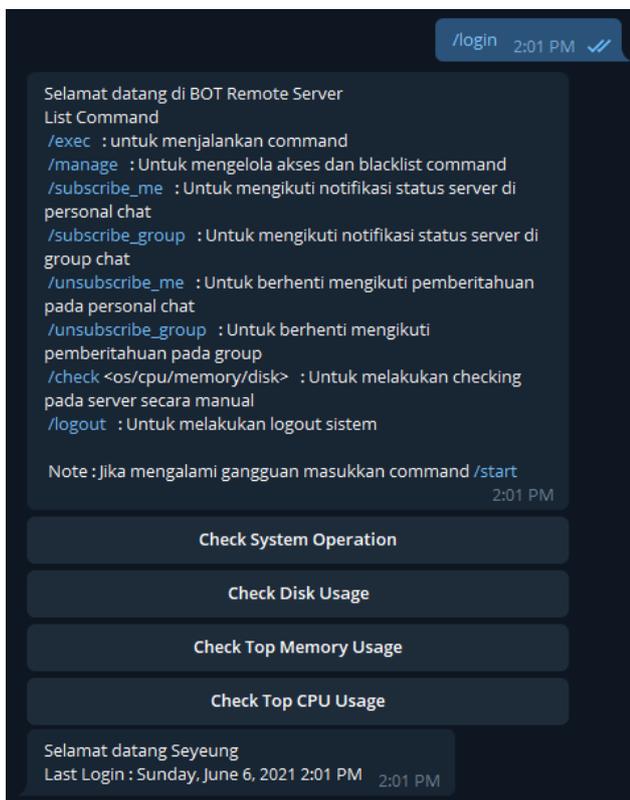
A. Pengujian

Pengujian sistem dilakukan pada aplikasi *Telegram Messenger*. Pengguna akan mendapatkan tampilan pada Gambar 4 setelah berhasil melalui proses login. Tersedia beberapa perintah yang didukung, pengguna dapat menekan tombol atau memasukkan perintah sesuai dengan deskripsi panduan. *Chatbot* akan memberikan balasan sesuai dengan masukan perintah. Dengan pendekatan ini Admin dapat mengelola peladen dengan hanya bermodalkan gawai.

users		subscribe		blacklists	
id	int	id	int	id	int
username	varchar(255)	username	varchar(255)	command	varchar(255)
id_telegram	varchar(255)	id_telegram	varchar(255)		
isAdmin	boolean	createdAt	timestamp		
isAllowed	boolean	updatedAt	timestamp		
last_login	date				
createdAt	timestamp				
updatedAt	timestamp				

Gambar 3. Implementasi desain basis data

Sistem dapat juga menangani perintah custom sesuai dengan kebutuhan Admin. Pada Gambar 5 disajikan tampilan pengecekan layanan status SSH (*Secure Shell Protocol*) Server. Admin mengirimkan pesan mentah (*raw*) contoh: *systemctl status ssh*. Chatbot akan membalas berupa informasi hasil perintah yang dimasukkan.



Gambar 4. Tampilan awal perintah yang didukung system

Informasi pemberitahuan terkait kesehatan peladen, dapat juga didapatkan oleh Admin. Informasi akan disampaikan secara otomatis oleh Chatbot. Contoh tampilan luaran pemberitahuan ke pengguna, dapat dilihat pada Gambar 6.

Pengujian usabilitas sistem dilakukan dengan mengambil data survei berdasarkan skenario yang telah ditetapkan pada 15 pengguna dengan 2 orang mempunyai latar belakang Admin. Terdapat 11 pertanyaan terkait kasus sistem dengan batasan nilai 1-5. Pada Tabel 1 terdapat hasil

rekap hasil survei. Dari hasil pengujian didapatkan nilai 86,41% dari segi usabilitas penggunaan.

Tabel 1. Hasil survey sistem

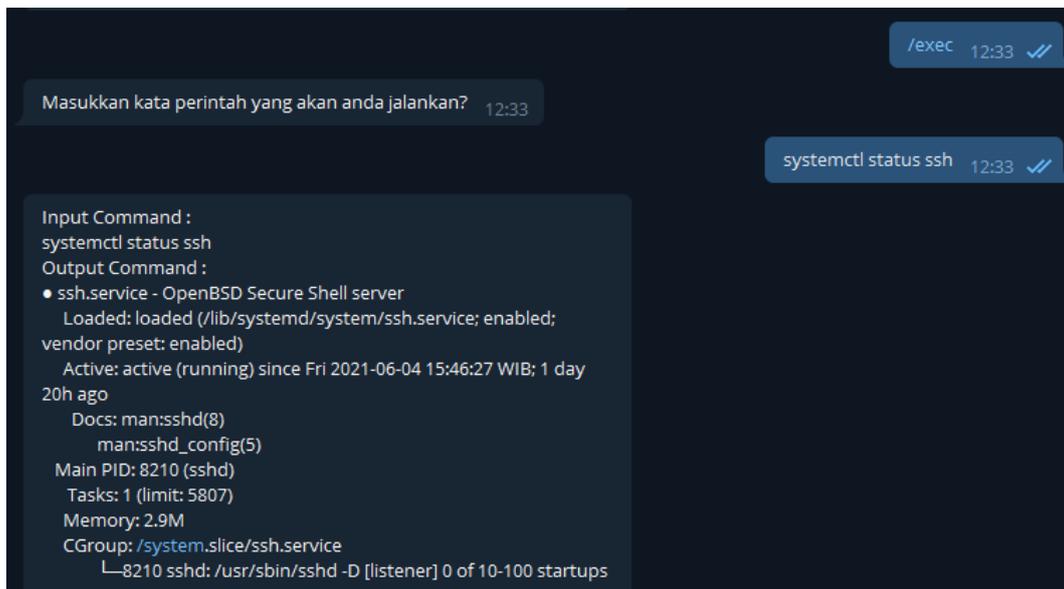
Q	Nilai					Total
	5	4	3	2	1	
Q1	4	9	2	0	0	62,00
Q2	11	3	1	0	0	70,00
Q3	5	7	3	0	0	62,00
Q4	7	7	1	0	0	66,00
Q5	8	7	0	0	0	68,00
Q6	4	9	2	0	0	62,00
Q7	10	3	2	0	0	68,00
Q8	5	9	1	0	0	64,00
Q9	4	9	2	0	0	62,00
Q10	6	8	1	0	0	65,00
Q11	6	7	2	0	0	64,00
Rata-Rata						64,81
Persentase (%)						86,41

V. KESIMPULAN

Kesimpulan dari penelitian yang didapatkan adalah admin dapat melakukan kontrol jarak jauh pada peladen dengan menggunakan tampilan menu berupa tombol atau perintah mentah (*raw*). Informasi terkait keadaan peladen dapat disampaikan pada Admin berbasis *chatbot*. Hasil pengujian usabilitas menunjukkan nilai sebesar 86,41% dalam penggunaan sistem kontrol dan pemberitahuan berbasis *chatbot*. Pengembangan lebih lanjut pada penelitian ini dapat digunakan untuk pengelolaan pada jumlah peladen lebih dari satu. Selain itu dapat juga dikembangkan untuk dukungan peladen selain sistem operasi Linux.

VI. REFERENSI

- [1] Sudiana, & Handayani, T. (2015). Perancangan Sistem Monitoring Server Data Base Secara Remote Control Berbasis Vpn (Virtual Private Network). Prosiding Seminar Nasional XI "Rekayasa Teknologi Industri Dan Informasi 2016 Sekolah Tinggi Teknologi Nasional Yogyakarta", 88–98.
- [2] de Oliveira, J. C., Santos, D. H., & Neto, M. P. (2016). Chatting with Arduino platform through Telegram Bot. 2016 IEEE International Symposium on Consumer Electronics (ISCE), 131–132.



```
Input Command :
systemctl status ssh
Output Command :
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled;
  vendor preset: enabled)
  Active: active (running) since Fri 2021-06-04 15:46:27 WIB; 1 day
  20h ago
  Docs: man:sshd(8)
  man:sshd_config(5)
  Main PID: 8210 (sshd)
  Tasks: 1 (limit: 5807)
  Memory: 2.9M
  CGroup: /system.slice/ssh.service
└─8210 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Gambar 5. Contoh hasil perintah custom



```
Selamat Siang, Syehfi.
=====
Server Time : Sunday, June 6, 2021 1:30 PM
Manufacturer : LENOVO / 80E1
Bios : LENOVO / A2CN45WW(V2.13)
CPU : AMD A6-6310 APU with AMD Radeon R4 Graphics 1.38 Ghz
CPU Speed : 1.38,1.75,1.28,1.42
CPU Temperature :
Total Memory : 5135233024 bytes
Used / Free Memory : 3840180224 bytes / 1295052800 bytes
Total Swap : 7999582208 bytes
Used / Free Swap : 0 bytes / 7999582208 bytes
```

Gambar 6. Contoh tampilan informasi pemberitahuan

- [3] Nufusula, R., & Susanto, A. (2018). Rancang Bangun Chat Bot Pada Server Pulsa Menggunakan Telegram Bot API. *JOINS (Journal of Information System)*, 3(1), 80–88.
- [4] OpenJS Foundation. (2020). About Node.js®. <https://nodejs.org/en/about/>
- [5] Visionmedia. (2020). Mastering Node. <http://visionmedia.github.io/masteringnode/book.html>
- [6] Kreibich, J. (2010). Using SQLite (M. Loukides (ed.); First Edit). O'Reilly Media, Inc. <http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf>
- [7] C. Huda, F. A. Bachtiar and A. A. Supianto, "Reporting Sleepy Driver into Channel Telegram via Telegram Bot," 2019 International Conference on Sustainable Information Engineering and Technology (SIET), 2019, pp. 251-256.
- [8] D. Korotaeva, M. Khlopotov, A. Makarenko, E. Chikshova, N. Startseva and A. Chemysheva, "Botanicum: a Telegram Bot for Tree Classification," 2018 22nd Conference of Open Innovations Association (FRUCT), 2018, pp. 88-93.
- [9] Z. N. Fikile Gasa, P. A. Owolawi, T. Mapayi and K. Odeyemi, "MobileNet Neural Network skin disease detector with Raspberry pi Integrated to Telegram," 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), 2020, pp. 1-5.
- [10] M. Nazarkevych, P. Levush and B. Pankovych, "Improving the Efficiency of Information Collection Based on the Development of a Chatbot with a Parser," 2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT), 2021, pp. 125-129.