

Rancang Bangun Sistem Visualisasi Data *Monitoring Server* Dengan Teknologi *Chatbot*

I Putu Wijaya Adi Candra^{a1}, I Made Agus Dwi Suarjaya^{a2}, I Made Sukarsa^{b3}

Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

e-mail: ¹adicandraofficial@gmail.com, ²agussuarjaya@it.unud.ac.id, ³sukarsa@unud.ac.id

Abstrak

Teknologi internet yang berkembang membutuhkan sumber daya manusia untuk melakukan pengelolaan informasi yang umumnya menggunakan skema jaringan komputer *client-server*. Skema tersebut membutuhkan server yang bekerja memberikan layanan secara waktu nyata, namun terkadang terdapat permasalahan seperti membludaknya klien dan keadaan seperti terputusnya koneksi dan padamnya listrik. Administrator yang harus memantau keadaan server memiliki keterbatasan waktu dan tidak bisa melakukan perbaikan secara cepat apabila tidak terhubung langsung ke jaringan atau berada pada depan layar komputer. Sistem visualisasi data monitoring server dengan teknologi chatbot ditawarkan untuk mengatasi keterbatasan tersebut. Metode pengambilan data monitoring menggunakan metode *agentless monitoring*. Pengujian sistem visualisasi data monitoring server dengan teknologi chatbot dilakukan dengan membandingkan kesesuaian data hasil monitoring. Chatbot yang dibangun dapat memberikan data monitoring server berbasis Linux dengan tingkat kesesuaian respon sebesar 29 respon sesuai dan 16 respon tidak sesuai dan rata-rata waktu eksekusi perintah yaitu 4.8 detik

Kata kunci: Server, Monitoring, Chatbot, Visualisasi, Agentless

Abstract

The growing internet technology requires many human resources to manage information that generally uses a *client-server* computer network scheme. The scheme necessitates a server that operates in real time to provide services, but there are occasionally issues, such as a large number of client requests and circumstances such as network disconnection and power outages. Administrators who have to monitor the state of the server have limited time and cannot make quick troubleshooting decisions if they are not directly connected to the network or in front of a computer. A server data visualization monitoring system with a chatbot is offered to overcome these limitations. Method of monitoring data using *agentless monitoring*. The monitoring server data visualization system with chatbot technology is tested by comparing the suitability of the monitoring data. The chatbot can provide Linux-based server monitoring data with a conformity response rate of 29 appropriate responses and 16 inappropriate responses and an average command execution time of 4.8 seconds.

Keywords: Server, Monitoring, Chatbot, Visualization, Agentless

1. Pendahuluan

Perkembangan teknologi jaringan internet membutuhkan banyak orang untuk mengelola sumber daya informasi menggunakan skema *client-server* [9]. Skema *client-server* menuntut komputer server untuk selalu memberikan layanan *real-time* selama 24 jam, namun kadang terdapat kejadian-kejadian seperti membludaknya *client* yang mengakses server dalam satu waktu atau adanya *force majeure* seperti padamnya listrik atau terputusnya koneksi ke jaringan publik atau privat.

Administrator mempunyai keterbatasan dalam memantau 24 jam secara terus-menerus lalu lintas data dan keadaan dari layanan server dan terdapat kemungkinan besar terjadinya permasalahan layanan jaringan ketika administrator tidak berada di depan komputer untuk melakukan tindakan perbaikan. Hal-hal tersebut menjadi dasar sehingga diperlukan sebuah Sistem Visualisasi Data *Monitoring Server* yang bekerja untuk memantau dan menyampaikan

hasil *monitoring* dalam bentuk visual dan teks kepada administrator secara langsung ke *smartphone*.

Teknologi *monitoring* jaringan sebelumnya menggunakan beberapa media antarmuka atau *interface* untuk menyampaikan hasil dari *monitoring* dan sebagai sistem peringatannya, seperti penggunaan SMS *gateway* dan *website* [6] [9]. Penelitian yang akan diajukan menggunakan pendekatan teknologi lain yang lebih baru yaitu menggunakan *chatbot*. Pertukaran data hasil *monitoring* yang terjadi antara layanan jaringan yang diawasi dengan pengguna menggunakan API atau *Application Programming Interface* pihak ketiga yang juga menjadi fondasi dari modul *chatbot*. Penggunaan API tersebut menjadikan pertukaran data menjadi *less investment* atau dengan kata lain tidak diperlukan peralatan baru dalam membuat suatu sistem yang juga berarti biaya dapat ditekan [4].

Chatbot dipilih karena dewasa ini tren penggunaan aplikasi perpesanan pada perangkat *mobile* bertumbuh dengan cepat. *Chatbot* juga dipilih karena merupakan aplikasi yang dekat dengan penggunaannya dan administrator relatif cepat dan efisien dalam menerima informasi *monitoring* [7]. Administrator melalui informasi layanan dan sumber daya jaringan dari *smartphone*-nya saja diharapkan dapat menjadi dasar dalam melakukan tindakan pengambilan keputusan untuk manajemen jaringan khususnya *server*.

Penelitian ini hanya menggunakan API (*Application Programming Interface*) dari Telegram sebagai fondasi layanan *chatbot* yang dapat diakses melalui *smartphone*, komputer tablet, komputer personal, dan laptop. *Monitoring* pada *server* akan dilakukan menggunakan sistem operasi berbasis Linux untuk proses pengambilan data sumber daya dari layanan-layanan *server*, visualisasi dalam bentuk grafik data *monitoring server*, dan pembuatan modul *chatbot* secara *soft real time*.

2. Metode Penelitian

Metode penelitian adalah salah satu aspek dasar dalam pengerjaan sebuah tugas akhir. Metodologi penelitian terdiri atas tahapan-tahapan yang terurut dan menggambarkan bentuk dari tugas akhir. Metodologi penelitian ini dilakukan dalam lima tahapan yaitu, identifikasi masalah, observasi dan studi literatur, perancangan dan implementasi, pengujian, dan dokumentasi.

2.1 Gambaran Umum Sistem

Gambaran umum dari Rancang Bangun Sistem Visualisasi Data *Monitoring Server* dapat dibagi menjadi dua modul utama sebagai berikut:

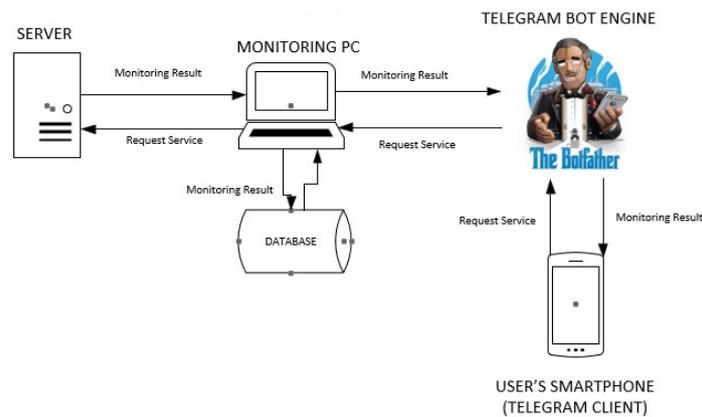
1. Modul *Monitoring*

Modul *monitoring* berfungsi untuk mengambil data *monitoring* dari *server* maupun node yang terhubung dengan satu jaringan. Metode pengambilan data *monitoring* adalah menggunakan metode *agentless monitoring* dengan bantuan *monitoring PC*.

2. Modul *Chatbot*

Modul *chatbot* adalah modul yang mengatur penerimaan perintah dan pengiriman hasil *monitoring* jaringan. Modul *chatbot* menggunakan *Application Programming Interface* (API) yang disediakan oleh aplikasi *instant messaging* Telegram. API tersebut memungkinkan dibangunnya sistem *chatbot* yang reliabilitasnya dapat dipercaya serta kaya akan fitur. Fitur-fitur lainnya yang diatur sebagai bagian yang melengkapi sistem seperti pengaturan hak akses, penyampaian data yang informatif dan cepat seperti menggunakan grafik dan gambar, serta penyediaan *commands* atau perintah-perintah yang sudah didefinisikan sebelumnya.

Modul-modul tersebut serta keterkaitannya satu sama lain sebagai satu kesatuan sistem dapat dijelaskan sebagai berikut.

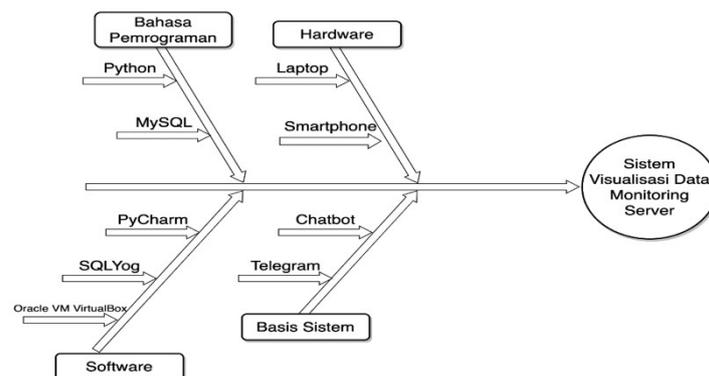


Gambar 1. Gambaran Umum Sistem

Gambar 1 merupakan gambaran umum dari Sistem Visualisasi Data *Monitoring Server*. *User* yang telah memasang aplikasi Telegram dan menambahkan *chatbot monitoring* akan melakukan *request* dari *service* yang disediakan. Sistem yang dibuat akan melakukan *monitoring* status dari *web server* Apache, dan *database server* MySQL serta memantau keadaan dari sumber daya perangkat keras seperti CPU, RAM, *harddisk*, dan paket data yang keluar-masuk. Bot Telegram akan melakukan menjalankan *engine monitoring* yang terdapat pada *monitoring PC* (*Personal Computer*). *Monitoring PC* adalah sebuah PC biasa yang akan menjalankan *script monitoring*, *task scheduling*, dan sebagai tempat *database* yang menampung hasil *monitoring* ditempatkan. *Monitoring PC* haruslah PC yang selalu dalam keadaan menyala dan terus melakukan *monitoring* ke *server*.

2.2 Fishbone Diagram

Fishbone diagram atau diagram Ishikawa adalah diagram yang berfungsi sebagai alat bantu dalam mengetahui segala kemungkinan sumber masalah pada aplikasi atau sistem yang dibuat. *Fishbone* diagram dari Sistem Visualisasi Data *Monitoring Server* dijelaskan pada Gambar 2 sebagai berikut.

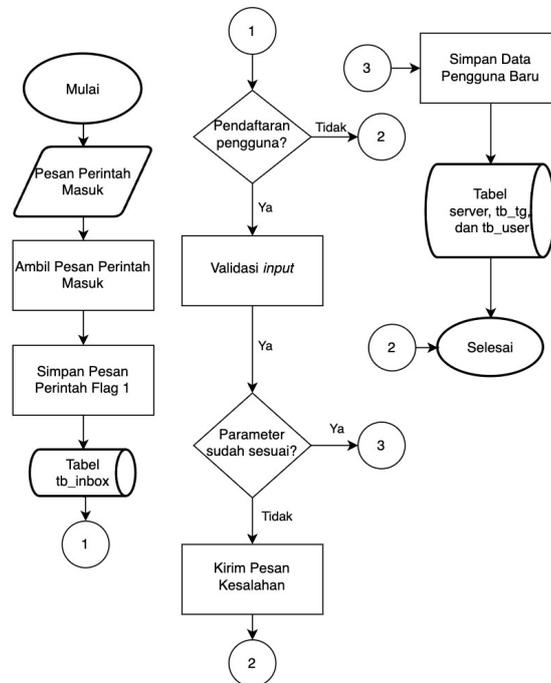


Gambar 2. Fishbone Diagram

Gambar 2 merupakan rancangan *fishbone* diagram yang dibuat sebagai penunjang dalam merancang dan membangun Sistem Visualisasi Data *Monitoring Server*. Atribut yang diperlukan antara lain *Hardware*, *Software*, Bahasa Pemrograman, dan Basis Sistem. *Hardware* atau perangkat keras yang diperlukan adalah Laptop sebagai media virtualisasi sistem operasi dan *monitoring PC* dan *smartphone* sebagai Telegram *client*. *Software* atau perangkat lunak yang digunakan antara lain PyCharm, SQLYog, serta Oracle VM *VirtualBox*. Bahasa Pemrograman yang digunakan adalah Python dan MySQL sebagai RDBMS. Basis Sistem yang dibuat adalah *chatbot* pada aplikasi Telegram.

2.3 Diagram Alir Penerimaan Pesan (*Message Inbox*)

Flowchart atau diagram alir penerimaan pesan menggambarkan proses penerimaan pesan perintah dari pengguna ke *chatbot monitoring server*. *Flowchart* penerimaan pesan ditunjukkan pada Gambar 3.



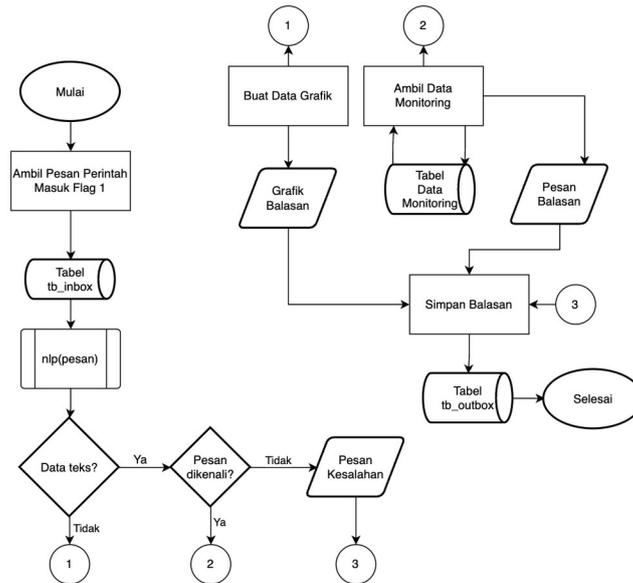
Gambar 3. *Flowchart* Penerimaan Pesan

Gambar 3 menjelaskan bahwa *Engine* pada sistem *monitoring server* akan mengambil pesan perintah masuk melalui API (*Application Programming Interface*) Telegram menggunakan *package Telepot* serta melakukan pengecekan apakah pengguna ingin melakukan pendaftaran dengan perintah */register*, apabila iya maka akan dilakukan validasi input pada parameter pendaftaran (seperti *IP Address*, *username*, dan *password*), apabila tidak maka pesan perintah akan disimpan pada basis data. *Engine* akan melakukan pengecekan terhadap status pengguna, apakah pengguna biasa atau *Super User*.

Super User juga dapat melakukan perubahan data nilai ambang batas pada tabel *threshold_alert* sebagai nilai acuan untuk peringatan titik kritis otomatis pada penggunaan sumber daya *server*. Data pengguna baru akan sudah disetujui akan disimpan ke dalam tabel *tb_tg* untuk menyimpan data pengguna Telegram, tabel *server* untuk menyimpan data *server*, dan *tb_user* untuk menyimpan *username* dan *password* SSH. Pesan akan disimpan pada tabel *tb_inbox* dengan status dalam *field flag* bernilai 1, nilai 1 berarti pesan perintah sudah disimpan dalam basis data namun belum diproses oleh *engine*.

2.4 Diagram Alir Pemrosesan Pesan (*Message Process*)

Flowchart pemrosesan pesan menggambarkan proses pengolahan pesan perintah dari pengguna yang nantinya akan diolah di *engine chatbot monitoring server*. *Flowchart* penerimaan pesan ditunjukkan pada Gambar 4.

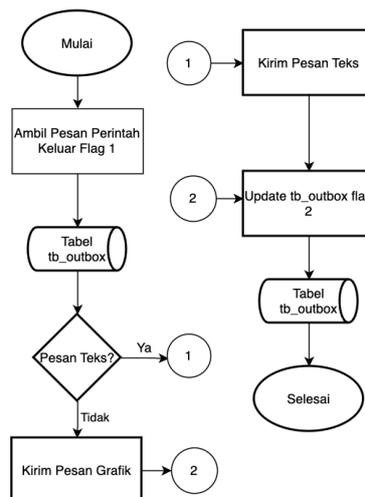


Gambar 4. Flowchart Pemrosesan Pesan

Gambar 4 menjelaskan bahwa *Engine* akan mengambil pesan perintah pada tabel *tb_inbox* dengan status dalam *field flag* bernilai 1. *Engine* akan melakukan pengecekan apakah data *monitoring* yang diminta adalah data dalam bentuk teks atau visualisasi grafik. Pesan perintah akan diproses sehingga menghasilkan balasan yang sesuai. Pesan perintah akan disimpan pada tabel *tb_outbox* dengan *field flag* bernilai 1, nilai 1 berarti pesan perintah sudah disimpan dalam basis data namun belum diproses oleh *engine*. Pesan perintah pada tabel *tb_inbox* akan diperbarui dengan *flag* bernilai 0, nilai 0 berarti pesan perintah sudah selesai diproses oleh *engine*.

2.5 Diagram Alir Pengiriman Pesan (*Message Outbox*)

Flowchart atau diagram alir pengiriman pesan menggambarkan proses pengiriman pesan balasan yang berupa hasil *monitoring server* atau hasil dari proses manajemen pengguna dari *chatbot monitoring server* kepada pengguna yang melakukan permintaan layanan. *Flowchart* pengiriman pesan ditunjukkan pada Gambar 5.



Gambar 5. Flowchart Pengiriman Pesan

Gambar 5 menjelaskan bahwa *Engine* akan mengambil pesan hasil perintah pada tabel *tb_outbox* dengan *field flag* bernilai 1. Pesan hasil perintah akan dikirimkan kepada pengguna

yang sesuai menggunakan API Telegram. Pesan hasil perintah pada tabel *tb_outbox* akan diperbarui dengan *flag* bernilai 2, dimana nilai 2 berarti pesan hasil perintah sudah berhasil dikirimkan.

3. Studi Literatur

Kajian pustaka digunakan menjelaskan beberapa teori pendukung yang akan menunjang dalam proses pembuatan Rancang Bangun Sistem Visualisasi Data *Monitoring Server* dengan Teknologi *Chatbot* juga penelitian-penelitian terdahulu yang memiliki kaitan dengan penelitian ini.

3.1 Dasar Monitoring

Monitoring atau pemantauan adalah sebuah proses yang meliputi kegiatan pengumpulan, analisis, pelaporan, dan tindakan terkait informasi dari suatu sistem. *Monitoring* diperlukan untuk mendeteksi kesalahan yang terjadi pada sistem secepat mungkin, sehingga dapat dilakukan tindakan perbaikan atau pencegahan untuk mengurangi risiko yang terjadi. Kegiatan untuk memantau kondisi dari suatu perangkat *server* pada jaringan dikenal dengan *monitoring server* [3]. Tujuan dari *monitoring* adalah untuk memastikan bahwa kebijakan yang diterapkan sesuai dengan tujuan awal, menemukan kesalahan secepat mungkin untuk mengurangi risiko yang lebih besar, dan melakukan tindakan perbaikan terhadap kebijakan jika hasil *monitoring* mendukung tindakan tersebut [5].

3.2 Chatbot

Chatbot merupakan gabungan dua kata dalam bahasa Inggris, yaitu "*chat*" yang berarti percakapan dalam bentuk teks maupun suara, dan "*bot*" yang berarti program komputer yang mengandung data. Apabila diberikan masukan, *chatbot* akan memberikan jawaban atas masukan tersebut. *Chatbot* memiliki arti sebagai program komputer yang dapat melakukan percakapan dengan cerdas dengan pengguna melalui teks maupun suara secara singkat [2]. *Chatbot* dapat mengerti perintah dari pengguna dan melakukan komputasi dari perintah tersebut, lalu mengeksekusi sesuai dengan perintah. Hasilnya adalah jawaban dari *chatbot* yang menyerupai percakapan antar manusia. Keunggulan dari *chatbot* adalah antarmuka yang menyerupai percakapan yang sering dilakukan manusia, serta dapat difokuskan pada fitur kecerdasan tertentu, sehingga informasi dapat dengan cepat dan efektif diterima oleh pengguna.

3.3 Natural Language Processing

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang mempelajari cara komputer dan manusia berkomunikasi dengan bahasa alami. Model komputasi ini memfasilitasi komunikasi antara manusia dan komputer dalam pencarian informasi dan memungkinkan dialog yang menggunakan bahasa alami. Tujuan dari studi yang menggunakan metode NLP adalah untuk membuat mesin dapat menjawab pertanyaan dan memahami kalimat lisan atau tulisan seperti yang dilakukan manusia. Arsitektur standar untuk pemrosesan NLP meliputi pengenalan ucapan, pemahaman bahasa, manajemen dialog, berkomunikasi dengan sistem eksternal, mengembalikan contoh respon, dan pengucapan respon.

3.4 Metode Agentless

Metode *agentless* adalah metode *monitoring* jaringan yang menggunakan bantuan sebuah node/komputer untuk berkomunikasi dengan *server* atau node lain. Cara kerja dari metode ini adalah komputer akan terus melakukan *request* ke *server* yang akan dipantau untuk mendapatkan nilai dari sumber daya secara *soft real time* atau dengan kata lain, sistem tidak memiliki tenggat waktu untuk menyelesaikan tugas pengambilan hasil *monitoring* dan sistem dapat melanjutkan fungsinya apabila ada keterlambatan dalam pengiriman hasil *monitoring*. Keuntungan dari metode ini adalah sebagai solusi *back up* apabila *node* dalam keadaan mati dan sebagai solusi apabila *server* tidak dimungkinkan untuk memasang dan mengelola *monitoring agent*. Metode ini berbeda dengan metode yang mengandalkan sebuah agent yang ditanam pada *server target*, seperti penerapan *Simple Network Management Protocol* (SNMP) sebagai protokol umum yang dipergunakan untuk mendapatkan informasi dari status jaringan

yang dikelola. SNMP mengandalkan 3 bagian yaitu *agent*, *manager*, dan *Manager Information Base (MIB)* [1] [6].

3.5 Linux

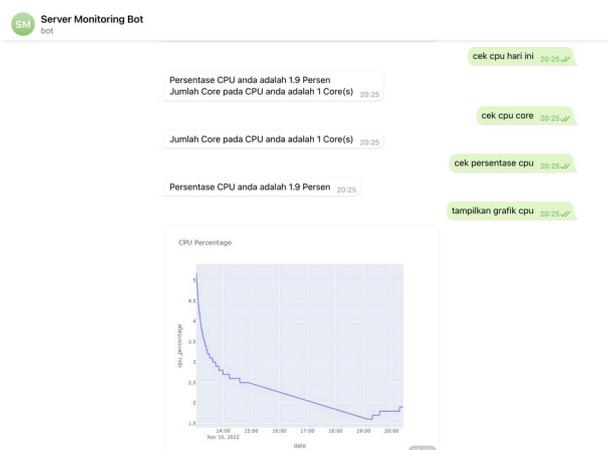
Linux merupakan salah satu varian dari sistem operasi Unix yang merupakan sistem operasi sumber terbuka yang dapat diperoleh gratis dan kodenya dapat diperoleh dibawah lisensi GNU (*GNU is Not Unix*), *General Public License (GPL)*. Linux merupakan sistem operasi yang dikembangkan oleh komunitas pengguna dan pengembang. Sistem operasi ini dirancang untuk digunakan pada berbagai perangkat seperti komputer pribadi, *server*, *supercomputer*, telepon seluler, perangkat IoT dan lain-lain. Linux memiliki banyak distribusi atau varian yang disesuaikan dengan kebutuhan pengguna. Beberapa distribusi Linux yang populer diantaranya Ubuntu, Fedora, Debian, dan Mint. Linux juga memiliki banyak fitur yang disesuaikan dengan kebutuhan pengguna seperti keamanan, performa, dan kemudahan penggunaan. Sistem operasi ini juga menawarkan banyak aplikasi yang bisa digunakan secara gratis dan kode sumber yang dapat dikembangkan oleh siapapun.

4. Hasil dan Pembahasan

Bagian ini akan diuraikan hasil dan analisis dari pengujian yang dilakukan pada Rancang Bangun Sistem Visualisasi Data *Monitoring Server* menggunakan Teknologi *Chatbot*. Hasil tersebut merupakan hasil pengujian dari aplikasi yang telah dikembangkan dan analisis dari aplikasi tersebut.

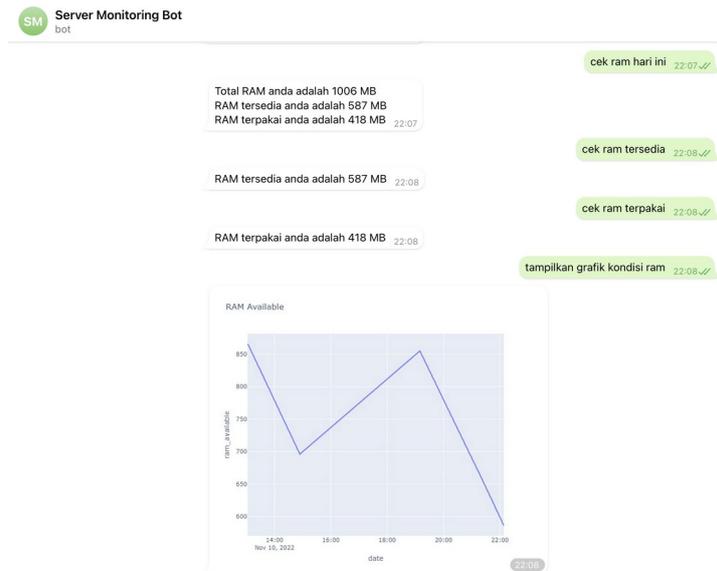
4.1 Implementasi Antarmuka *Chatbot*

Pengguna dapat berinteraksi dengan *chatbot* dengan antarmuka aplikasi Telegram. *Chatbot* dapat menampilkan hasil *monitoring* dari CPU, RAM, media penyimpanan, jaringan, basis data, *service*, serta status *port* jaringan dalam bentuk visualisasi data dan teks. Berikut adalah implementasi antarmuka dari *chatbot monitoring server*.



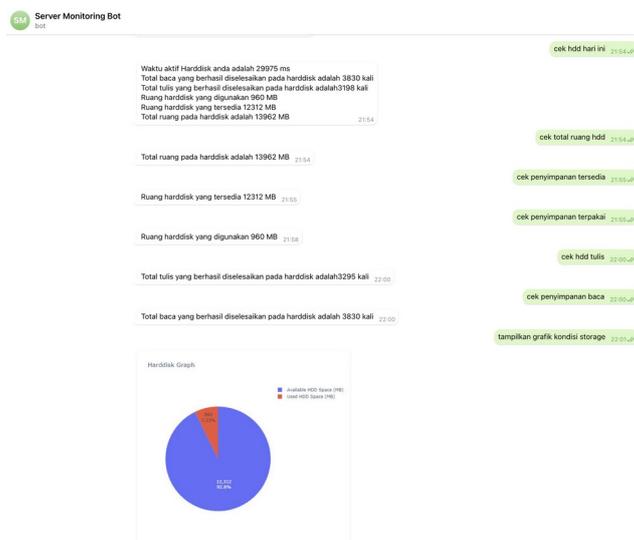
Gambar 6. Antarmuka *Monitoring CPU*

Gambar 6 menjelaskan tentang beberapa perintah yang didukung oleh sistem *monitoring server* untuk sumber daya CPU. Perintah yang didukung oleh sistem *monitoring server* untuk CPU adalah melakukan *monitoring* secara menyeluruh menurut rentang waktu, pengecekan jumlah inti (*core*) pada CPU, persentase penggunaan CPU, dan grafik dari persentase penggunaan CPU.



Gambar 7. Antarmuka Monitoring RAM

Gambar 7 menjelaskan tentang beberapa perintah yang didukung oleh sistem *monitoring server* untuk sumber daya RAM. Perintah yang didukung oleh sistem *monitoring server* untuk RAM adalah melakukan *monitoring* secara menyeluruh menurut rentang waktu, pengecekan kapasitas RAM yang tersedia, kapasitas RAM yang terpakai, dan grafik dari kapasitas RAM yang tersedia dalam rentang waktu tertentu.



Gambar 8. Antarmuka Monitoring Media Penyimpanan

Gambar 8 menjelaskan tentang beberapa perintah yang didukung oleh sistem *monitoring server* untuk sumber daya media penyimpanan (*storage*). Perintah yang didukung oleh sistem *monitoring server* untuk *storage* adalah melakukan *monitoring* secara menyeluruh menurut rentang waktu, pengecekan total ruang pada *storage*, kapasitas *storage* yang tersisa, kapasitas *storage* yang terpakai, total tulis (*write*) pada *storage*, total baca (*read*) pada *storage*, dan grafik dari kapasitas *storage* yang tersedia dan terpakai dalam bentuk *pie chart*.

4.2 Pengujian Sistem

Pengujian sistem dilakukan dengan mengirimkan 25 contoh perintah kepada *chatbot* terkait proses *monitoring server* dengan keluaran dalam bentuk teks dan 20 contoh perintah

terkait proses *monitoring server* dengan keluaran dalam bentuk grafik visualisasi data. Respon yang diberikan oleh *chatbot* akan dianalisis untuk mengetahui kesesuaian hasil yang didapatkan apakah sesuai atau tidak beserta waktu yang diperlukan untuk memproses perintah hingga sampai ke pengguna. Hasil rekapitulasi pengujian perintah dengan keluaran teks dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Perintah dengan Keluaran Teks

No	Fitur	Perintah	Respon	Waktu/Status	Keterangan
1	CPU	Lakukan cek core pada cpu	Jumlah Core pada CPU anda adalah 1 Core(s)	00:02.19 Sesuai	-
2	CPU	Berapa persentase cpu yang telah terpakai?	Pesan salah	00:01.66 Tidak sesuai	Tidak ada kata kunci "Cek"
3	CPU	berapa core cpu?	Pesan salah	00:02.43 Tidak sesuai	Tidak ada kata kunci "Cek"
4	RAM	Ram udah terpakai berapa, minta tolong cek	RAM terpakai anda adalah 257 MB	00:02.13 Sesuai	-
5	RAM	Tampilkan ram nya dong	Pesan salah	00:03.03 Tidak Sesuai	Tidak ada kata kunci "Cek"
6	RAM	tampilkan kapasitas ram?	Total RAM anda adalah 1006 MB, RAM tersedia anda adalah 30 MB, RAM terpakai anda adalah 975 MB	00:02.14 Sesuai	-
7	Storage	Cek total ruang yang tersedia tanggal 3 dong	Total ruang pada <i>harddisk</i> adalah 13962 MB	00:02.46 Sesuai	-
8	Storage	Bisa di cek penyimpanan yang terpakai berapa?	Ruang <i>harddisk</i> yang digunakan 966 MB	00:02.59 Sesuai	-
9	Storage	Cek berapa kapasitas penyimpanan	Waktu aktif <i>Harddisk</i> anda adalah 15969 ms Total baca yang berhasil diselesaikan pada <i>harddisk</i> adalah 1091 kali Total tulis yang berhasil diselesaikan pada <i>harddisk</i> adalah 1867 kali Ruang <i>harddisk</i> yang digunakan 966 MB Ruang <i>harddisk</i> yang tersedia 12306 MB Total ruang pada <i>harddisk</i> adalah 13962 MB	00:02.01 Sesuai	-

No	Fitur	Perintah	Respon	Waktu/Status	Keterangan
10	Database	Cek waktu nyala pada basis data	MySQL server telah berjalan selama 50 detik	00:03.88 Sesuai	-
11	Database	Cek berapa jumlah tables yang terbuka pada database	Jumlah tabel yang sedang terbuka sebanyak 1 tabel	00:02.86 Sesuai	-
12	Database	Tabel apa saja yang terbuka?	Pesan tidak dimengerti	00:02.92 Tidak sesuai	Perintah tidak diketahui
13	Network	Tolong dong, cek network keluar, makasi	Data yang keluar dari server sebesar 1408.4 Kilobytes	00:02.31 Sesuai	-
14	Network	Cek ntwork dong	Pesan tidak jelas	00:02.25 Tidak sesuai	Salah ketik pada kata kunci
15	Network	Lihat penggunaan network	Data yang diterima oleh server sebesar 5962.9 Kilobytes, Data yang keluar dari server sebesar 6106.5 Kilobytes	00:01.66 Sesuai	-
16	Port Status	Lakukan cek pada port 80, apakah terbuka atau tertutup?	port: 80 Tertutup	00:08.02 Sesuai	-
17	Port Status	Port 20 terbuka atau tertutup ya? tolong cek	port: 20 Tertutup	00:16.04 Sesuai	-
18	Port Status	Kondisi port 3306 gimana?	Kondisi port 3306 gimana?	00:02.58 Tidak sesuai	Tidak ada kata kunci "Cek"
19	Service	Jalankan perintah service httpd start	Service httpd berhasil dijalankan	00:29.08 Sesuai	-
20	Service	Tolong stop service httpd, makasi	Service httpd berhasil dimatikan	00:25.42 Sesuai	-
21	Service	Restart service httpd nya dong	Pesan salah	00:01.26 Tidak sesuai	Perintah restart tidak tersedia
22	Manajemen pengguna	defaultip	Default IP address Anda adalah 192.168.100.5.	00:04.05 Sesuai	-
23	Manajemen pengguna	showallip	Ip address terdaftar pada akun Anda adalah: 192.168.100.5.	00:03.24 Sesuai	-
24	Manajemen pengguna	Tampilkan default ip nya	Pesan tidak jelas	00:02.64 Tidak sesuai	Perintah manajemen pengguna menggunakan pola statik

No	Fitur	Perintah	Respon	Waktu/Status	Keterangan
25	Manajemen pengguna	Tampilkan seluruh ip	Pesan salah	00:02.67 Tidak sesuai	Perintah manajemen pengguna menggunakan pola statik

Tabel 1 merupakan data hasil *monitoring* yang diuji coba pada *chatbot*. *Chatbot* mampu memberikan hasil *monitoring* dari *server* target berbasis Linux. Perintah yang diujikan berjumlah 25 data dengan 16 respon sesuai dan 9 respon yang tidak sesuai. Berikut hasil rekapitulasi data pengujian dengan keluaran garfik visualisasi data akan ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Perintah dengan Keluaran Garfik Visualisasi Data

No	Fitur	Perintah	Respon	Waktu/Status	Keterangan
1	CPU	Tampilkan grafik dari cpu dong	[Visualisasi Data] CPU.png	00:05.23 Sesuai	-
2	CPU	Bagaimana ya visualisasi keadaan cpu?	[Visualisasi Data] CPU.png	00:08.56 Sesuai	-
3	CPU	Hari ini grafik cpunya baik?	[Visualisasi Data] CPU.png	00:05.90 Sesuai	-
4	CPU	Tampilan grafik cpu nya tolong ditunjukkan	Persentase CPU anda adalah 4.7 Persen, Jumlah Core pada CPU anda adalah 1 Core(s)	00:02.72 Tidak Sesuai	Terdapat kata kunci "tampil"
5	CPU	Buatkan gambar penggunaan cpu	[Visualisasi Data] CPU.png	00:04.70 Sesuai	-
6	RAM	Tampilkan grafik dari kondisi ram dong	[Visualisasi Data] RAM.png	00:04.96 Sesuai	-
7	RAM	Tolong tunjukkan gambar ram	[Visualisasi Data] RAM.png	00:04.53 Sesuai	-
8	RAM	Tunjukkan penggunaan ram hari ini	Pesan salah	00:02.65 Tidak Sesuai	Tidak terdapat kata kunci "grafik"
9	RAM	Visualisasikan ram minggu ini yang dipakai	[Visualisasi Data] RAM.png	00:05.43 Sesuai	-
10	RAM	Bisa tampilkan grafik ram digunakan	Total RAM anda adalah 1006 MB, RAM tersedia anda adalah 500 MB, RAM terpakai anda adalah 505 MB	00:02.49 Tidak Sesuai	Terdapat salah ketik pada kata kunci "grafik"
11	Storage	Bisa tampilan grafik storage yang digunakan	Waktu aktif <i>Harddisk</i> anda adalah 1269024 ms Total baca yang berhasil diselesaikan pada <i>harddisk</i> adalah 1821 kali Total tulis yang berhasil diselesaikan pada <i>harddisk</i> adalah 149607 kali Ruang <i>harddisk</i> yang	00:02.91 Tidak sesuai	Terdapat kata kunci "tampil"

No	Fitur	Perintah	Respon	Waktu/ Status	Keterangan
			digunakan 1552 MB Ruang <i>harddisk</i> yang tersedia 11720 MB Total ruang pada <i>harddisk</i> adalah 13962 MB		
12	Storage	Hari ini penyimpanan tolong di visualisasikan	[Visualisasi Data] Storage.png	00:05.18 Sesuai	-
13	Storage	Minggu ini gambarkan penyimpanan yang dipakai	[Visualisasi Data] Storage.png	00:03.54 Sesuai	-
14	Storage	Grafik <i>storagenya</i> tolong ditunjukkan	[Visualisasi Data] Storage.png	00:03.84 Sesuai	-
15	Storage	Tunjukkan penggunaan penyimpanan	Pesan tidak dimengerti	00:03.84 Tidak sesuai	Tidak terdapat kata kunci "grafik"
16	Networ k	Tampilkan grafik jaringan	[Visualisasi Data] Network.png	00:04.57 Sesuai	-
17	Networ k	Visualisasi kan jaringan yang diterima	[Visualisasi Data] Network.png	00:04.85 Sesuai	-
18	Networ k	Tunjukan data yang dikeluarkan hari ini	Pesan salah	00:02.17 Tidak sesuai	Data yang diminta tidak spesifik
19	Networ k	Tunjukan <i>network</i> hari ini	[Visualisasi Data] Network.png	00:02.13 Sesuai	-
20	Networ k	Tolong Grafik jaringan yang diterima ditampilkan	Data yang diterima oleh <i>server</i> sebesar 10750.81 Kilobytes	00:02.67 Tidak sesuai	Terdapat kata kunci "tampil"

Tabel 2 merupakan data hasil *monitoring* yang diuji coba pada *chatbot* dengan keluaran grafik visualisasi data. Perintah yang diujikan berjumlah 20 data dengan 13 respon sesuai dan 7 respon yang tidak sesuai. Terdapat beberapa faktor yang menyebabkan ketidaksesuaian antara respon dan perintah. Faktor pertama yaitu kesesuaian padanan kata kunci yang menentukan apakah layanan *monitoring* yang diakses tersedia atau tidak. Semakin banyak padanan yang didaftarkan yang terkait sumber daya *server* yang dapat di-monitor maka akan semakin kecil kemungkinan *chatbot* untuk mengeluarkan jawaban yang tidak sesuai.

Tabel 1 dan 2 juga menunjukkan waktu yang diperlukan bagi *engine chatbot* untuk memproses atau mengeksekusi perintah dan mengirimkan melalui antarmuka Telegram. Rata-rata waktu yang diperlukan untuk memproses seluruh perintah yang diujikan adalah 4.8 detik. Rata-rata waktu yang dibutuhkan untuk memproses perintah dengan hasil berupa teks adalah 3.36 detik. Rata-rata waktu yang diperlukan untuk memproses perintah dengan hasil *monitoring* berupa visualisasi data adalah 4.35 detik, sedikit lebih lama dibandingkan dengan hasil berupa teks dikarenakan adanya fungsi tambahan untuk menghasilkan visualisasi data berupa grafik. Waktu tertinggi yang diperlukan untuk mengeksekusi perintah adalah perintah untuk *monitoring service*. Hal tersebut terjadi karena *engine* menjalankan perintah Linux secara langsung pada *server* dan diperlukan waktu untuk menerima balikan data dari *server*, sehingga diperlukan waktu rata-rata 18.3 detik.

4.3 Analisa Hasil Pengujian

Analisa hasil dari pengujian pada *chatbot* menghasilkan beberapa peluang kesalahan seperti perintah yang tidak tersedia, kesalahan terjadi saat pengguna menjalankan perintah yang tidak mengandung kata kunci atau ketika perintah yang dijalankan tidak tersedia pada *chatbot*, sehingga *chatbot* memberikan hasil bahwa pesan salah atau tidak jelas. Kesalahan mengenali kata kunci sehingga memberikan hasil *monitoring* yang tidak sesuai dengan yang dibutuhkan oleh pengguna. Kesalahan karena internet yang tidak stabil sehingga *chatbot* Telegram tidak bisa menerima perintah dan mengirimkan hasil *monitoring* kepada pengguna dengan baik.

5. Kesimpulan

Kesimpulan yang diperoleh dari penelitian yang telah dilakukan adalah bahwa Rancang Bangun Sistem Visualisasi Data *Monitoring Server* dengan Teknologi *Chatbot* dibangun dengan menggunakan bahasa pemrograman Python dan metode *agentless* yang mampu memberikan hasil *monitoring server* Linux dalam bentuk teks maupun grafik visualisasi data. Sistem visualisasi data *monitoring server* dengan teknologi *chatbot* diimplementasikan dengan tiga *engine* yaitu, *engine* penerimaan pesan, *engine* pemrosesan pesan, dan *engine* pengiriman pesan. *Chatbot* menggunakan aplikasi *instant messaging* Telegram dalam berkomunikasi dengan pengguna *Chatbot* memberikan 29 respon yang sesuai dan 16 respon tidak sesuai dari uji coba yang dilakukan dengan total 45 perintah, dengan rata-rata waktu eksekusi perintah sebesar 4.8 detik.

Daftar Pustaka

- [1] Abidin, M. H. S. dan Ardian, Y. (2015) "Rancang Bangun Aplikasi *Monitoring Network* Berbasis Web Menggunakan Html5," Teknik Informatika, hal. 1–7.
- [2] Akhsan, A. A. dan Faizah (2017) "Analisis dan Perancangan Interaksi *Chatbot* Reminder dengan User-Centered Design," Jurnal Sistem Informasi, 13(2), hal. 78–89. doi: <http://dx.doi.org/10.21609/jsi.v13i2.555>.
- [3] Christopher, K. et al. (2022) "Analisis Dan *Monitoring* Layanan Internet Kepada Pelanggan Menggunakan Aplikasi Berbasis Open Source (Studi Kasus: Pt Jaya Kartha Solusindo)," JITTER-Jurnal Ilmiah Teknologi dan Komputer, 3(1).
- [4] Hanafi, A., Sukarsa, I. M. dan Wiranatha, A. A. K. A. C. (2017) "Pertukaran Data Antar Database dengan Menggunakan Teknologi API," Lontar Komputer, 8(1), hal. 22–30.
- [5] Khan, R. dan Khan, S. U. (2017) "Design and implementation of an automated network *monitoring* and reporting back system," Journal of Industrial Information Integration, (September), hal. 1–11. doi: 10.1016/J.JII.2017.11.001.
- [6] Nugroho, M., Affandi, A. dan Rahardjo, D. S. (2014) "Rancang Bangun Aplikasi *Monitoring* Jaringan Menggunakan SNMP (Simple Network Management Protocol) dengan Sistem Peringatan Dini dan Mapping Jaringan," Jurnal Teknik Pomits, 3(1), hal. 35–39.
- [7] Paliwahet, I. N. S., Sukarsa, I. M. dan Putra, D. (2017) "Pencarian Informasi Wisata Daerah Bali menggunakan Teknologi *Chatbot*," Lontar Komputer, 8(3), hal. 144–153.
- [8] Piarsa, I. N. dan Togantara, S. (2012) "Sistem *Monitoring* Spesifikasi dan Utilitas Host di Jaringan Komputer Berbasis Web," 3(2), hal. 179–187.
- [9] Taufan Dwi Prayogo, Kushartantya, H. A. W. (2011) "Sistem *Monitoring* Jaringan Pada Server Linux Dengan Menggunakan Sms Gateway," Fmipa Undip, 2, hal. 63–72. Tersedia pada: <http://ejournal.undip.ac.id/index.php/jmasif/article/view/2648/2355>.
- [10] Wikantya, I. M. A., Suarjaya, I. M. A. D. dan Raharja, I. M. S. (2021) "Design and Building Mikrotik Based Billing System and *Chatbot* as Media Transaction Online," Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi), 9(2), hal. 165. doi: 10.24843/jim.2021.v09.i02.p07.
- [11] P, E. N. S. C. dan Afrianto, I. 2015. Rancang Bangun Aplikasi *Chatbot* Informasi Objek Wisata Kota Bandung dengan Pendekatan Natural Language Processing. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, Vol. 4, No. 1.