

Implementasi *Remote Monitoring* Pada *Virtual Private Server* Berbasis Telegram Bot API (Studi Kasus Politeknik TEDC Bandung)

Mamay Syani¹, Bayu Saputro²

Teknik Informatika, Politeknik TEDC Bandung
Jl.Politeknik-Pesantren KM.2 Cibabat Cimahi Utara Kota Cimahi 40513
¹msyani@poltektedc.ac.id¹, ²bayu190198@gmail.com

Diterima : 23 Februari 2021

Disetujui : 28 Maret 2021

Abstract— *Virtual Private Server (VPS)* merupakan layanan *server* yang menggunakan teknologi *bare metal hypervisor* untuk membagi *hardware server* fisik menjadi beberapa *server* virtual yang di *hosting* di infrastruktur fisik yang sama. Virtualisasi menawarkan kemudahan untuk meng-*host* beberapa *server* pada satu *server* fisik. Setiap *server* dapat memiliki layanan dan sistem operasi yang berbeda satu sama lain. Dimana permasalahan yang sering terjadi di sebuah institusi pendidikan seperti di Politeknik TEDC, dalam implementasi teknologi (*VPS*) kurangnya *awareness* terhadap ancaman dari pihak dalam dan luar, sehingga perlunya pengawasan yang terintegrasi dengan seorang *system administrator*. Melihat permasalahan tersebut dengan menggunakan *tools zabbix* sebagai *network monitoring system dashboard* memiliki fitur *map* dan *grafik* sehingga membantu dalam proses pengawasan, pengaturan dan administrasi pada sistemnya, *zabbix* memiliki fitur *Application Programming Interface (API) Bot* yang diintegrasikan dengan aplikasi telegram, yaitu sistem perpesanan lintas *platform* yang berpusat pada keamanan dan privasi keamanan dan privasi pemakainya berbasis *cloud*. Permasalahan yang sering terjadi di Politeknik TEDC dengan populasi pengguna mahasiswa 1874 orang dan 146 dosen di tambah karyawan dan staff mencapai 100 orang, dalam pemanfaatan akses sistem informasi oleh civitas Politeknik TEDC, seorang admin dalam mengelola (*VPS*) terdapat masalah seperti kurangnya sumber daya manusia terutama seorang *system administrator* yang bertugas mengawasi kinerja *server* sehingga kurang dapat melakukan pengamatan (*VPS*) selama 24 jam. Implementasi *Bot API* sudah banyak digunakan dengan keunggulan dalam keandalan untuk menyediakan data ke *user* yang tidak terbatas oleh waktu. Penggunaan pemantauan mengintegrasikan fungsi *Application Programming Interface (API) telegram* ke dalamnya untuk dapat mengirim pesan dan memeriksa secara *realtime* seperti Informasi *problem alerts, user login, active host, memory, cpu dan hdd status* ditunjukkan dalam satuan *percent*, serta detail total kapasitas, *available dan used*.

Kata Kunci: *Virtual Private Server (VPS), Remote Monitoring, Zabbix, Bot, API, Telegram.*

I. PENDAHULUAN

Virtual Private Server (VPS) adalah layanan *server* yang menggunakan teknologi *bare metal hypervisor* untuk membagi *hardware server* fisik menjadi beberapa *server* virtual yang di *hosting* di infrastruktur fisik yang sama

Politeknik TEDC merupakan salah satu perguruan tinggi vokasi yang berada di wilayah Jawa Barat, dengan jumlah dosen, mahasiswa dan tenaga non kependidikan mencapai 2120

orang berdasarkan dari dari PDPT dan Sister Kemdikbud, dengan di implementasikannya sistem informasi terintegrasi, baik untuk aplikasi e-learning, sistem informasi dosen, sistem informasi akademik, PDPT dan sistem informasi pendukung lainnya dengan teknologi *traditional IT infrastructure*, di gunakan untuk *firewall* menggunakan Mikrotik *RouterOS* sebagai dan *remote monitoring SNMP (Simple Network Management Protocol)* yang digunakan pada

teknologi yang sedang berjalan. Pada infrastruktur jaringan yang digunakan sering terjadi kendala pada layanan seperti aplikasi pada *web server*, terjadi gagal akses, seorang *system administrator* yang bertugas mengawasi *server* di Politeknik TEDC, tidak memiliki seseorang yang khusus di bidang jaringan *server* sehingga *system administrator* yang bertugas untuk menangani infrastruktur jaringan dan *virtual private server* (VPS) Politeknik TEDC masih menggunakan pihak ketiga dan untuk yang menangani IT infrastruktur tidak berada di ruangan *server* ataupun memantau *server* secara langsung.

Solusi yang akan di implemnsasikan yaitu menggunakan *tools* zabbix yang salah satu *software open source* yang dapat digunakan untuk memonitor jaringan, dan status dari berbagai *network services*, *servers* dan *network hardware* lainnya. Pemantuan jaringan dengan kemampuan untuk merekam statistik jaringan dengan konsep *Simple Network Management Protocol* (SNMP). Dengan zabbix, pengguna dapat mudah mengetahui *status server*, kondisi jaringan dan mendapatkan notifikasi jika terjadi gangguan, dengannya fitur zabbix *alerts* dan sudah mendukung *polling* dan *trapping*.

Berdasar uraian diatas pada penulisan ini bermaksud melakukan penelitian dengan judul "Implementasi *Remote Monitoring* Pada *Virtual Private Server* Berbasis Telegram Bot API (Studi Kasus Politeknik TEDC Bandung)".

II. LANDASAN TEORI

A. Penelitian Terkait

Menurut Anggiat Cokrojoyo, Justinus Andjarwirawan, dan Agustinus Noertjahyana dalam penelitian mengenai pembuatan Bot Telegram untuk, mengambil informasi dan jadwal film menggunakan API mengatakan bahwa dengan bot maka pengguna hanya butuh untuk mengirimkan perintah ke akun Bot dan jadwal jam tayang atau informasi yang diinginkan akan langsung diberikan ke pengguna yang membutuhkan.[1]

Menurut David Karismata W. Tion dan Indrastanti R. Widiyari dalam penelitian mengenai perancangan Bot untuk *remote monitoring* pada *server* menggunakan Telegram Bot API, untuk mengatasi *human error* dalam proses pengawasan terutama *server*, seorang *system administrator* yang bertanggung jawab dalam mengamati kinerja *server* selama 24 jam, dibutuhkan layanan yang dapat melakukan

proses *remote monitoring* untuk menambah fleksibilitas serta dapat melakukan monitoring terhadap *server* secara *real time* apabila terjadi masalah pada jaringan Bot akan mengirim notifikasi kepada pengelola layanan jaringan.[2]

Menurut Dwi Wijonarko dalam penelitiannya mengenai zabbix network monitoring sebagai perangkat monitoring jaringan di SKPD Kota Malang dengan menggunakan zabbix, proses monitoring jaringan komputer dapat dilakukan tanpa harus berada dalam lingkup jaringan, bisa dimonitor jarak jauh dimanapun dan kapanpun selama *agent* masih terhubung ke internet.[8]

B. Virtual Private Server

Sebuah teknologi virtualisasi sistem operasi dan perangkat lunak yang memungkinkan sebuah mesin dengan kapasitas besar dibagi ke beberapa virtual mesin. Tiap virtual mesin ini melayani sistem operasi dan perangkat lunak secara *dedicated*. VPS juga dapat diartikan sebagai sebuah metode untuk mempartisi atau membagi *resource* sebuah *server* menjadi beberapa *server* virtual. *Virtual machine* tersebut memiliki kemampuan menjalankan sistem operasi sendiri seperti layaknya sebuah *server*. Bahkan pengguna dapat mengelola sebuah *virtual machine* secara terpisah, dari *hypervisor* os *server* utama. Untuk mengendalikan VPS (*Virtual Private Server*) pada sistem operasi Windows menggunakan protocol RDP (*Remote Desktop Protocol*) dengan port TCP/UDP 3389, sedangkan sistem operasi Linux menggunakan SSH (*Secure shell*) dengan port default TCP/UDP 22, apabila menggunakan VPS dilengkapi dengan *control panel* untuk mengelola *script*, *users*, pemrosesan, *file* sistem *backup restore* dan fitur lainnya, VPS bekerja seperti sebuah *server* yang terpisah memiliki *processes*, *users*, *files* dan menyediakan *full root access*. Setiap VPS mempunyai *IP address*, *port number*, *tables*, *filtering* dan *routing rules* sendiri. VPS juga dapat melakukan konfigurasi *file* untuk sistem dan *software service*.

Dengan VPS pengguna tidak perlu lagi merawat *server virtual*, karena penyedia VPS akan merawat secara berkala serta melakukan *upgrade operating system*, *system backup*, dan sebagainya secara berkala.[5]

C. Remote Monitoring

Remote monitoring atau RMON adalah suatu teknik untuk melakukan pemantauan kondisi jaringan. Pemantauan jaringan ini sangat terkait dengan *traffic* atau lalu lintas data pada

infrastruktur jaringan baik menggunakan *traditional IT infrastructure* dan. *Remote Monitoring* atau *RMON* pada awalnya dikembangkan untuk mengatasi masalah manajemen segmen situs dan *Local Area Network (LAN)* dari lokasi terpusat. Standar *RMON* menentukan sekelompok fungsi dan statistik yang dapat dipertukarkan antara *network probes* yang kompatibel dengan *RMON* dan *console managers*. *RMON* melakukan deteksi kesalahan jaringan yang luas dan menyediakan data penyesuaian kinerja untuk *NAs*.

RMON mengumpulkan sembilan jenis informasi, termasuk *bytes sent, packets sent, packets dropped dan statistics by host*. *NAs* menggunakan *RMON* untuk menentukan lalu lintas pengguna jaringan atau tingkat *bandwidth* dan informasi akses situs web. *RMON* menggunakan perangkat jaringan tertentu, seperti *server*, dan berisi aplikasi manajemen jaringan yang berfungsi sebagai *client*. *RMON* mengontrol jaringan dengan menggunakan *protocol remote TCP/UDP* dan aplikasinya secara bersamaan. Ketika sebuah paket jaringan ditransmisikan, *RMON* memfasilitasi penglihatan status paket dan memberikan informasi lebih lanjut, dalam hal suatu paket diblokir, dihentikan atau hilang.[6]

D. Telegram

Telegram adalah aplikasi pesan *instant* berbasis cloud yang fokus pada kecepatan dan keamanan. Telegram dirancang untuk memudahkan pengguna saling berkirim pesan teks, audio, video, gambar dan stiker dengan aman. Secara default, seluruh konten yang ditransfer akan dienkripsi berstandar internasional. Dengan demikian, pesan yang terkirim sepenuhnya aman dari pihak ketiga, bahkan dari telegram sekalipun. Bukan hanya teks, gambar dan video, telegram juga bisa jadi sarana user mengirimkan dokumen, file multimedia, berkas, berbagi lokasi *real-time* dan kontak yang tersimpan di perangkat ke orang lain. asal, orang yang dituju juga mempunyai aplikasi dengan akun telegram terdaftar di perangkatnya. Telegram juga merupakan aplikasi yang ringan, cepat, tidak ada iklan, dan gratis selamanya. Lalu *user* bisa menggunakan telegram dengan menggunakan PC dan laptop via *web browser*. Kelebihan lainnya dari telegram yaitu dapat dimanfaatkan di dunia bisnis dan komunitas karena berbasis *cloud*.

Karena telegram berbasis *cloud*, maka penggunaanya dapat mengakses pesan dari perangkat yang berbeda secara bersamaan dan membagikan jumlah berkas yang tak terbatas hingga ukuran 1,5GB. Berkas ini dapat diatur untuk disimpan di dalam perangkat atau hanya di *cloud*.[7]

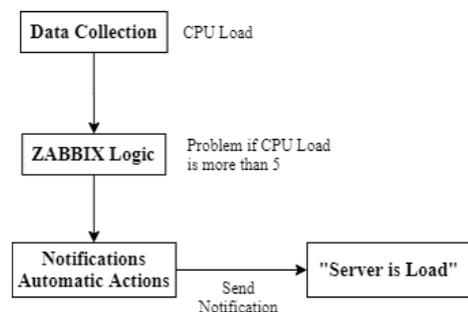
E. Zabbix

Zabbix adalah salah satu *software open source* yang dapat digunakan untuk memonitor jaringan, dan status dari berbagai *network services, servers* dan *network hardware* lainnya. Software diciptakan oleh Alexei Vladishev. dengan zabbix, pengguna dapat mudah mengetahui status *server*, kondisi jaringan dan mendapatkan notifikasi jika terjadi gangguan dan masalah, zabbix juga sudah mendukung *polling* dan *trapping*.

Tools ini menggunakan GPL license *open source*, sudah memiliki fitur *GUI* yang memudahkan pemakaian para pengguna, beberapa fitur yang ada berupa map dan grafik sehingga membantu pengaturan administrasi dengan mudah. Dengan menampilkan map jaringan komputer admin kelola beserta dengan statusnya. Maka admin pun akan tahu kondisi jaringan jika terjadi masalah. warna hijau menandakan status kondisi normal sedangkan warna merah jika terjadi masalah.

Untuk tipe laporan, pengguna dapat membuatnya sesuai waktu yang diinginkan, per minggu, bulan atau sesuai jangka waktu lainnya. Dilihat dari sisi keamanannya, software ini sudah mempunyai sistem *otentifikasi* dengan *IP address* dan memberikan informasi masalah dengan cepat, misalkan dengan e-mail atau sms dan Bot API.

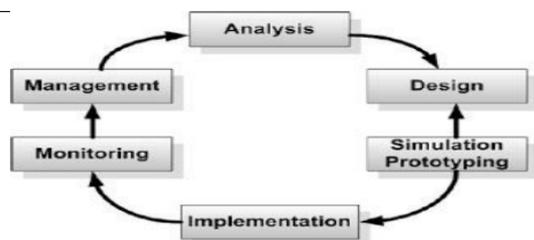
Salah satu cara kerja zabbix akan memonitoring beban dari *CPU* yang sedang dipantau dan apabila beban dari *CPU* melebihi standar yang telah diatur maka zabbix akan mengirim aksi notifikasi secara otomatis.[3]



Gambar 1. Cara Kerja Sistem Zabbix

III. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini menggunakan metode *NDLC* (*Network Development Life Cycle*).[9]



Sumber : (<https://www.research-methods-information-technology.html.co.id/>)

Gambar 2. Alur Network Development Life Cycle

Tahapan pada *Network Development Life Cycle* (NDLC).[9]

1. Tahap Analysis
Tahap awal ini dilakukan analisa kebutuhan, analisa permasalahan yang muncul, analisa keinginan user, dan analisa topologi atau jaringan yang sudah ada saat ini.
2. Tahap Design
Tahap desain ini akan membuat gambar desain topologi jaringan yang akan dibangun. Desain bisa berupa design structure topology, design access data, desain tata layout perkabelan, dan sebagainya yang akan memberikan gambaran jelas tentang proyek yang akan dibangun.
3. Tahap Simulation Prototype
Pada tahap ini beberapa network engineer akan membuat dalam bentuk simulasi dengan bantuan tools khusus di bidang network seperti BOSON, Packet Tracer, NETSIM, dan sebagainya.
4. Tahap Implementation
Dalam tahap implementasi network engineer akan menerapkan semua yang telah direncanakan dan di desain sebelumnya.
5. Tahap Monitoring
Tahapan monitoring merupakan tahapan yang penting agar jaringan dapat berjalan sesuai dengan keinginan dan tujuan dari user pada tahap awal analisis
6. Tahap Management
Pada tahap ini suatu kebijakan perlu dibuat untuk membuat atau mengatur agar sisem yang telah dibangun dapat berjalan dengan baik dan dapat berlangsung lama.

A. Analisis Sistem

Berdasarkan data yang diperoleh, infrastruktur jaringan yang digunakan Politeknik TEDC Bandung telah menggunakan fiber optik dengan provider *ASTINet* dengan untuk server *dedicated* memiliki 5 *IP public* 1:1 *downstream* dan *upstream* 3 MBps , yaitu untuk *IP Public* yang digunakan sebagai *gateway*, *IP server* PDPT Kemdikbud, *IP server* akademik lama, *IP server* akademik baru, dan untuk layanan *Sister dosen*. Teknologi fiber optik yang digunakan adalah GPON ZTE dengan menggunakan MikroTik RouterBoard 2011UiAS-2HnD sebagai *Firewall*.

Server *dedicated* yang digunakan yaitu Dell PowerEdge T30 Mini Tower Server, Dell PowerEdge T20 Mini Tower Server, Rainer Server dengan sistem operasi yang digunakan adalah Windows yang di pasang *Hypervisor* Oracle VirtualBox untuk menjalankan Linux Ubuntu Server,

B. Analisis Kebutuhan

Berdasarkan hasil observasi yang dilakukan di Politeknik TEDC Bandung, maka sistem yang akan dibangun memiliki spesifikasi perangkat keras seperti ditunjukkan pada tabel 1 dan perangkat lunak ditunjukkan pada tabel 2.

Tabel 1 Perangkat keras VPS Bot API

No	Perangkat	VPS
1	Processor	2 vCPUs
2	Memory	4GB
3	HDD	80GB

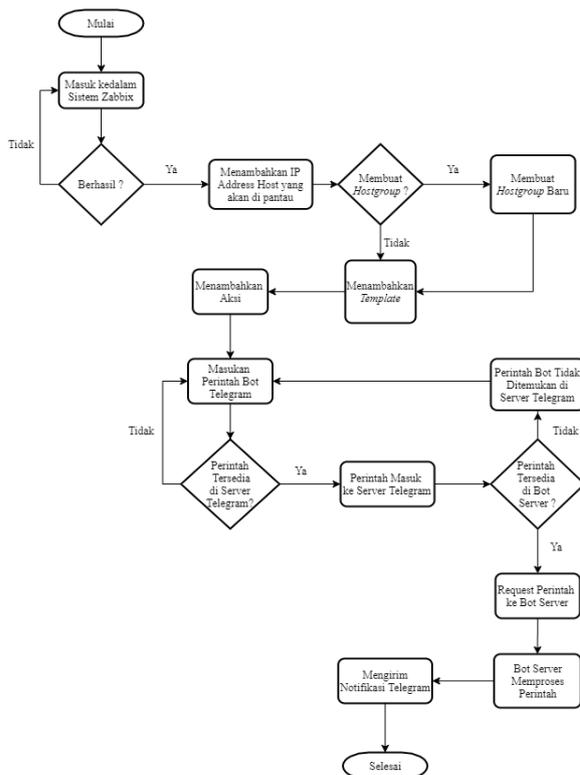
Tabel 2 Perangkat lunak VPS Bot API

No	Software	Versi	keterangan
1	Ubuntu Server	16.04.5	Sistem operasi yang akan digunakan di VPS
2	Zabbix Server	4.0	Aplikasi Monitoring Server
3	MySQL	10.1.29	Sistem penyimpanan data konfigurasi

C. Usulan Pemecahan Masalah

Dengan hasil penelitian yang dilakukan di Politeknik TEDC maka diajukan usulan untuk mengatasi permasalahan yang ditemukan yaitu dengan mengimplementasikan zabbix sebagai sistem pemantau kinerja *server* untuk mempermudah admin dalam proses pemantuan kinerja *server* dengan bantuan *Bot API* Telegram untuk notifikasi sistem, solusi yang dapat diberikan adalah:

- Sistem pemantauan yang di implementasikan dapat mempermudah dalam melakukan pemantauan kinerja *server* dengan *web GUI*.
- Saat terjadi masalah pada sistem akan mengirim notifikasi jenis masalah yang terjadi kepada admin jaringan melalui aplikasi telegram.
- Admin jaringan dapat *memonitoring* kinerja *server* dengan menggunakan kata kunci yang sudah terprogram pada *Bot API Telegram*.



Gambar 3. Flowchart Sistem Keseluruhan

Pada gambar 3 menggambarkan dari keseluruhan rancangan yang akan diimplementasikan pada sistem zabbix *server*, dimulai dari masuk kedalam sistem Zabbix melalui *web Graphical User Interface (GUI)* sampai mengirimkan notifikasi kepada admin jaringan berupa pesan menggunakan media *Bot API telegram*.

D. Daftar Kata Kunci Telegram Monitoring

Ada beberapa kata kunci yang sudah dirancang oleh penulis dan dapat digunakan oleh admin jaringan untuk *memonitoring* keadaan *server* ditunjukkan pada tabel 3.

Tabel 3 Daftar Kata Kunci *Bot API Telegram*

No	Kata Kunci	Penjelasan
1	<code>/users</code>	Kata kunci ini untuk menampilkan semua nama pengguna yang sudah terdaftar pada sistem zabbix.
2	<code>/host</code>	Kata kunci ini untuk menampilkan nama klien, alamat ip <i>address</i> , dan <i>port</i> yang sudah terdaftar pada sistem zabbix.
3	<code>/cpu</code>	Kata kunci ini untuk menampilkan informasi tentang kinerja memori yang sedang berjalan pada server.
4	<code>/hdd</code>	Kata kunci ini untuk menampilkan informasi tentang penggunaan media penyimpanan yang dipakai.

IV. HASIL DAN PEMBAHASAN

A. Kode Program 1

Inisialisasi Token dan Koneksi *Database Zabbix Server*

```

1. from _future_ import print_function
2. import psutil
3. import telebot
4. import mysql.connector
5. mydb = mysql.connector.connect(
6.     host = 'localhost', user = 'root',
7.     passwd = 'password', database = 'zabbixdb')
8. sql = mydb.cursor()
9. api = '1221808221:AAHRTOAp4ybERrrYdA
    qbb2fJBH71eDXE1Gk'
10. chat_id = '1251599434'
11. bot = telebot.TeleBot(api)
    
```

Sesuai dengan perancangan sistem yang telah dibuat, untuk menghubungkan *Bot API Telegram* dengan sistem dibutuhkan *token API* yang merupakan kunci untuk mengakses bot. *Chat id* adalah identitas yang digunakan untuk bot mengirim pesan dan *mydb* digunakan sebagai kunci untuk mengakses *database zabbix server*.

B. Kode Program 2

Koneksi *Bot API Telegram* dengan Sistem *zabbix server*

```

1. zbx_tg_prefix = "zbx_tg"
2. zbx_tg_tmp_dir = "/var/tmp/" +
    zbx_tg_prefix
3. zbx_tg_signature = False
4. zbx_tg_update_messages = True
5. zbx_tg_matches = {
6.     "problem": "PROBLEM: ",
7.     "ok": "OK: " }
8. zbx_server =
    
```

```
"http://192.168.137.11/zabbix/"
9. zbx_api_user = "api"
10. zbx_api_pass = "api"
11. zbx_api_verify = True
12. zbx_server_version = 4
```

Kode program 2 menjelaskan fungsi untuk bot telegram dapat mengakses ke sistem zabbix server sehingga Bot API telegram dapat mengirim notifikasi kepada admin jaringan ketika terjadi masalah dengan server.

C. Kode Program 3 Fungsi Users

```
1. @bot.message_handler(commands=['users'])
2. def users(message):
3. sql.execute("select alias from users")
4. hasil_user = sql.fetchall()
5. pesan_users = ''
6. for x in hasil_user:
7. pesan_users = pesan_users + str(x) + '\n'
8. bot.reply_to(message, pesan_users)
```

Kode program 3 menjelaskan fungsi ketika admin jaringan ingin mengetahui nama pengguna yang sudah terdaftar untuk mengakses sistem zabbix server dan Bot API telegram akan mengirim informasi tersebut dengan mengakses database sistem.

D. Kode Program 4 Fungsi Host

```
1. @bot.message_handler(commands=['host'])
2. def interface(message):
3. sql.execute("select host, ip, port from hosts
inner join interface on
hosts.hostid=interface.hostid;")
4. hasil_interface = sql.fetchall()
5. pesan_interface = ''
6. for x in hasil_interface:
7. pesan_interface = pesan_interface + str(x)
+ '\n'
```

Kode program 4 merupakan fungsi untuk menampilkan alamat ip address dan port klien yang sudah terhubung dengan sistem zabbix server dengan mengakses database sistem.

E. Kode Program 5 Fungsi CPU

```
1. @bot.message_handler(commands=['cpu'])
2. def interface(message):
3. cpu = psutil.virtual_memory()
4. total = cpu.total
```

```
5. percent = cpu.percent
6. avlable = cpu.available
7. used = cpu.used
8. bot.reply_to(message, """"
9. PERCENT = {}
10. TOTAL = {}
11. AVAILABLE = {}
12. USED = {}
13. """".format(percent, total, avlable, used))
```

Kode program 5 merupakan fungsi untuk menampilkan informasi dari kinerja memory server yang sedang berjalan.

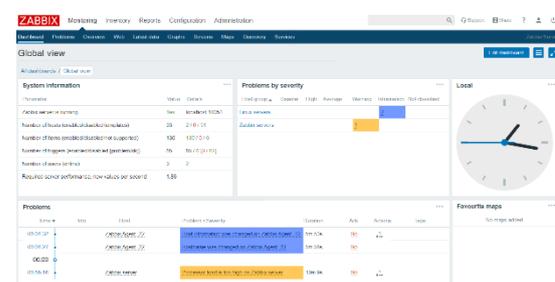
F. Kode Program 6 Fungsi HDD

```
1. @bot.message_handler(commands=['hdd'])
2. def interface(message):
3. hdd = psutil.disk_usage('/')
4. percent = hdd.percent
5. total = hdd.total
6. used = hdd.used
7. free = hdd.free
8. bot.reply_to(message, ""
9. PERCENT = {}
10. TOTAL = {}
11. USED = {}
12. FREE = {}
13. """".format(percent, total, used, free))
```

Kode program 6 berfungsi untuk menampilkan informasi tentang pemakaian harddisk atau media penyimpanan data server.

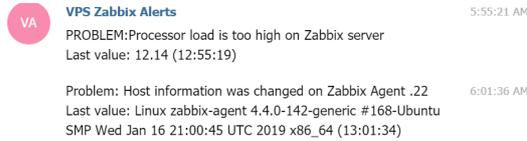
G. Pengujian Notifikasi Bot API Telegram

Untuk pengujian notifikasi Bot API telegram penulis menguji dengan membuat stres server dan client, penulis juga mengganti hostname client. Pada gambar 4 memperlihatkan bahwa zabbix server kinerja terlalu tinggi dan zabbix agent telah mengganti informasi (hostname).



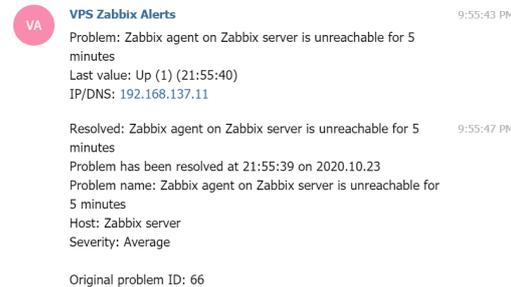
Gambar 4. Tampilan Menu Dashboard Ketika Terjadi Masalah

Pada gambar 5 menampilkan bot telegram mengirim pesan yang berisi informasi yaitu nama masalah yang sedang terjadi dan waktu terjadinya masalah tersebut.



Gambar 5. Tampilan Pesan Notifikasi Bot Telegram

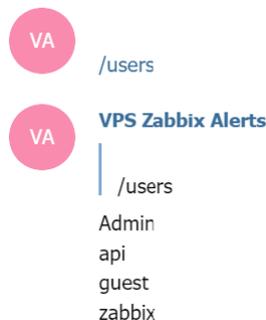
Pada gambar 6 menampilkan tampilan apabila masalah yang terjadi sudah selesai diperbaiki maka Bot telegram akan mengirim pesan yang berisi bahwa masalah yang terjadi pada server atau klien sudah diperbaiki.



Gambar 6. Tampilan Pesan Notifikasi Bot Telegram Masalah Selesai Diperbaiki

H. Pengujian Program Bot API Telegram

Pada gambar 7 menampilkan tampilan nama pengguna yang sudah terdaftar pada sistem zabbix server dengan menggunakan kata kunci `/users` di percakapan dengan Bot telegram.



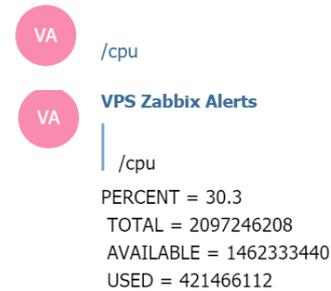
Gambar 7. Pengujian Kata Kunci `/users` di Telegram

Pada gambar 8 menampilkan tampilan nama klien, alamat `ip address`, dan `port` yang digunakan di sistem zabbix server dengan menggunakan perintah `/host` di percakapan dengan Bot telegram.



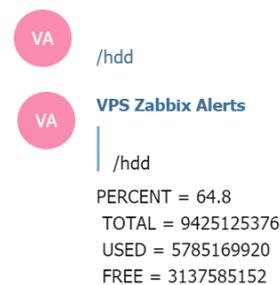
Gambar 8. Pengujian Kata Kunci `/host` di Telegram

Pada gambar 9 menampilkan tampilan dari berapa persen memori yang sedang berjalan, total memori yang dipasang di server, total memori yang tersedia, dan total `memory` yang terpakai oleh server.



Gambar 9. Pengujian Kata Kunci `/cpu` di Telegram

Pada gambar 10 menampilkan tampilan berapa persen penggunaan `harddisk server`, total `harddisk` yang digunakan di server, dan total `harddisk` yang tersisa untuk menyimpan data.



Gambar 10. Pengujian Kata Kunci `/hdd` di Telegram

V. SIMPULAN

Setelah melakukan pengujian terhadap sistem zabbix secara langsung, maka dapat disimpulkan bahwa dengan memanfaatkan aplikasi *Bot API* telegram bisa membantu *system administrator* jaringan ketika tidak berada di tempat pengawasan server dan dapat mengetahui keadaan server yang diawasi dengan aplikasi telegram yang di *install* di *smartphone* tanpa membuka PC (*Personal Computer*) ataupun

membuka laptop, hal ini ditunjukkan dengan sistem monitoring jaringan menggunakan sistem zabbix baik zabbix *server* ataupun zabbix *agent* berhasil dibangun. Pada pengujian sistem zabbix apabila terjadi masalah maka *dashboard* zabbix akan menampilkan informasi *client* yang mengalami masalah dan apabila masalah tersebut termasuk dengan pemicu notifikasi maka akan terkirim kepada system administrator dengan menggunakan perintah pada Bot *API* telegram. Pada pengujian perintah pada program bot telegram berhasil menampilkan beberapa informasi dari *server* yaitu nama *user* yang dapat mengakses sistem zabbix *server*, informasi nama *client* beserta alamat *IP address client*, informasi yang di tampilkan yaitu kinerja pemakaian *ram* atau *memory* dari *server* utama, lalu menampilkan informasi total penggunaan media penyimpanan data pada *harddisk* dan penggunaan *uptime Central Processing Unit (CPU)* kinerja pada *server* dan *logs* pada *server*.

DAFTAR PUSTAKA

- [1] Cokrojoyo, A., Andjarwirawan, J., & Noertjahyana, A. (2017). Pembuatan Bot Telegram Untuk Mengambil Informasi dan Jadwal Film Menggunakan PHP. *Jurnal Infra*, 5(1), 224–227, Program Studi Teknik Informatika Fakultas. <http://studentjournal.petra.ac.id/index.php/teknik-informatika/article/view/5163>
- [2] David Karismata W. Tion, & Wideasari, I. R. (2016). *Perancangan Bot untuk Remote Monitoring pada Server menggunakan Telegram Bot API*. [https://repository.uksw.edu/bitstream/123456789/11449/2/T1_672012109_Full text.pdf](https://repository.uksw.edu/bitstream/123456789/11449/2/T1_672012109_Full%20text.pdf)
- [3] Diro, S. (2015). *System Monitoring Virtual Menggunakan Zabbix*. Retrieved January 22, 2019, <https://www.slideshare.net/kasbecker5/system-monitoring-zabbix>
- [4] ICT. (2014). Pengertian RMON (Remote Network Monitoring). Retrieved January 22, 2019, from <https://www.proweb.co.id/articles/ict/rmon.html>
- [5] Lukman. (2017). Apa itu Virtual Private Server (VPS)? Dan Apa Aja Sih Kegunaannya? Retrieved January 22, 2019, from <https://www.cloudmatika.co.id/2017/04/25/apa-itu-virtual-private-server/>
- [6] Techopedia. (2019). Remote Monitoring (RMON). Retrieved January 21, 2019, from <https://www.techopedia.com/definition/24844/remote-monitoring-rmon>
- [7] Winarso, B. (2016). Sejarah dan Fitur-fitur Andalan Aplikasi Telegram. Retrieved September 8, 2019, from <https://dailysocial.id/post/apa-itu-telegram>
- [8] Wijonarko, D. (2014). Zabbix Network Monitoring Sebagai Perangkat Monitoring Jaringan Di. *Jurnal ELTEK*, 12(1), 27–38. Retrieved from <http://eltek.polinema.ac.id/index.php/eltek/article/view/31/31>
- [9] Anggorowati, O. P. D., Kurniawan, M. T., & Yunan, U. (2015). Desain dan Analisa Infrastruktur Jaringan Wireless di PDII-LIPI Jakarta dengan Menggunakan Metode Network Development Life Cycle (NDLC). *eProceedings of Engineering*, 2(2).