Sistem Informatika, Jaringan Komputer, dan Aplikasi
Vol 11 No 2 (2021)

PENGUJIAN PERFOMANCE SERVER TERHADAP *REALTIME DASHBOARD MONITORING* MENGGUNAKAN METODE *SCRIPTING AUTOMATION PROCESS*

[1]**Domo Pranowo Kuswandono** [2]**Mohammad Amri**
[1]Program Studi/Jurusan D3 Manajemen Informatika, STMIK Bani Saleh
[2]Mahasiswa Program Studi/Jurusan S1 Teknik Informatika, STMIK Bani Saleh
[1]domopranowo@stmik-banisaleh.ac.id, [2]muhamri1998@gmail.com

Abstract

Problems that often occur on servers who is in charge of observing server performance is not able to observe the server for 24 hours. The server will fail to perform its main functions and tasks due to full hard disk, high CPU & memory usage, so the server admin needs a quick notification before the problem occurs. The system design using the linear sequential method and the scripting automation process method by utilizing the Bash Shell Script and Python programming languages is expected to adapt to the company's needs while still following existing security requirements. It is expected that the implementation of the system can provide early warning information before decreasing performance server. Impact performance server after implementation monitoring system not significant, CPU load only increased 0.3% if used monitoring system with SNMP protocol

Keywords: *Monitoring server, sequential linear, automation scripting process.*

Abstrak

Permasalahan yang kerap terjadi pada *server* yaitu tugas mengamati kinerja *server* selama 24 jam. Masalah Utama *Server adalah* akan gagal melakukan fungsi dan tugas utamanya dikarenakan *harddisk* penuh, *CPU & memory usage* tinggi sehingga *server* admin membutuhkan notifikasi cepat sebelum masalah tersebut terjadi. Melihat permasalah tersebut, maka dibutuhkan sistem yang dapat melakukan proses realtime *remote monitoring* untuk mengetahui kondisi server secara berkala dan memberikan informasi peringatan dini *(Early Warning System)* ketika server akan mengalami masalah.
Perancangan *system* dengan menggunakan metode sekuensial linear dan metode scripting automation process dengan memanfaatkan bahasa pemprograman Bash Shell Script dan Python Diharapkan sistem dapat memberikan informasi peringatan dini sebelum server mengalami penurunan performa dan dapat memberikan informasi terkini terhadap kondisi server vital. Perfomance server terhadap penggunaan monitoring ini dengan protocol SNMP berdampak sangat ringan terhadap CPU load penambahan beban hanya 0.3%.

Kata Kunci: *Monitoring server,* sekuensial linear, *automation scripting process.*

1. PENDAHULUAN

Data yang besar tentu membutuhkan *server* yang baik untuk memenuhi kebutuhan penyimpanan dan pengelolaan data tersebut. Terdapat beberapa *server* yang berperan vital, seperti *Server Operation Support System (OSS), Server Network Monitoring System (NMS)* dan *server Two Way Active Monitoring Performance (TWAMP)*. Apabila terjadi kesalahan pada *server-server* diatas akan menyebabkan masalah ketersediaan data tertentu berkurang atau hilang. Permasalahan yang kerap terjadi pada *server* bisa terjadi karena kurang mampu mengamati kinerja *server* elama 24 jam. *Server* akan gagal melakukan fungsi dan tugas utamanya dikarenakan *harddisk* penuh, *CPU & memory usage* tinggi sehingga *server* admin membutuhkan notifikasi cepat sebelum masalah tersebut terjadi tanpa adanya *system* pendukung. Melihat permasalah tersebut, maka dibutuhkan sistem yang dapat melakukan proses realtime *remote monitoring* untuk mengetahui kondisi server secara berkala dan memberikan informasi peringatan dini *(Early Warning System)* ketika server akan mengalami masalah. Ada beberapa

metode yang bisa dipakai dalam perancangan peringatan dini *(Early Warning System)* dan realtime *remote monitoring* antara lain adalah dengan menggunakan produk *tool* pihak ketiga *monitoring server* memberikan kemudahan bagi pengguna karena tidak perlu merancang *design tool* namun memiliki kekurangan yaitu minimnya kostumisasi dan fleksibilitas *tool* untuk menyesuaikan dengan kebutuhan data. Perancangan system dengan menggunakan metode scripting automation process dengan memanfaatkan bahasa pemprograman Bash Shell Script dan Python diharapkan dapat menyesuaikan dengan kebutuhan perusahaan dengan tetap mengikuti ketentuan keamanan yang ada.
Pada penelitian kali ini penyusun akan meng-Implementasikan *Early Warning System* dan *Realtime Dashboard Monitoring Server* dengan megunakan metode *Sctipting Automation Process*, dengan harapan dapat memberikan informasi peringatan dini sebelum server mengalami penurunan performa dan dapat memberikan informasi terkini terhadap kondisi server vital pada perusahaan sehingga dapat mengurangi masalah hilang atau berkurangnya data pendukung.

1.1 IDENTIFIKASI MASALAH

a. Bagaimana meng-implementasikan sistem yang memberikan notifikasi peringatan dini sebelum server mengalami penurunan performa?
b. Bagaimakan meng-implementasikan sistem yang memberikan informasi realtime kondisi server?
c. Apakah sistem yang di implementasikan dapat menjalankan fungsi tanpa mengurangi performa server yang di monitoring?

1.2 IDENTIFIKASI MASALAH

Berdasarkan identifikasi diatas maka rumusan masalah yang dipilih pada penelitian ini adalah bagaimana merancang dan membangun sistem yang memberikan peringatan dini pada penurunan performa *server* dan memberikan kondisi *realtime server*

1.3 TUJUAN PENELITIAN

Tujuan penelitian dari pembuatan *Early Warning System* dan *Realtime Dashboard Monitoring Server* ini adalah:

a. Memberikan informasi *early warning alert* kepada *server admin* agar dapat mempercepat penanganan server
b. Memberikan informasi kondisi server secara *realtime* dengan *Dashboard Monitoring Server.*
c. Memberikan dampak performance server setelah system dijalankan

2. METODE

Dalam pengembangan Pengujian Perfomace server terhadap Realtime Dashboard Monitoring ini ada beberapa hal atau kegiatan yang dilakukan,
Menggunakan model sekuensial linier menunjukkan sistematis, pendekatan sekuensial untuk pengembangan perangkat lunak yang dimulai pada tingkat sistem dan kemajuan melalui analisis, desain, coding, pengujian, dan dukungan.
Metode *Scripting Automation Process*, Scripting adalah kode yang digunakan untuk mengotomatiskan proses yang seharusnya perlu dijalankan langkah demi langkah.

2.1 Monitoring Server

Monitoring server adalah kegiatan atau penggunaan suatu sistem yang secara konstan mengawasi *server* dan memberikan notifikasi kepada *administrator server* apabila terjadi sesuatu yang tidak diinginkan pada server tersebut. Dalam melakukan monitoring server terdapat beberapa bagian dalam server yang perlu di perhatikan dan awasi, berikut adalah beberapa parameter yang dapat mempengaruhi performa server:

a. Mendapatkan nilai RAM . LoadRAM merupakan salah satu perangkat keras komputer yang penting dalam membantu meningkatkan kinerja dari sebuah komputer. RAM merupakan kependekan dari *Random Access Memory*. Untuk mendapatkan informasi penggunaan RAM pada sistem operasi linux dapat menggunakan *command* “free -m”. Sedangkan untuk mendapatkan presentase penggunaan RAM diperlukan formula yang digabungkan dengan *command parsing* di linux/unix yaitu “AWK”.

b. Mendapatkan nilai CPU Load : CPU berarti kapasitas dari penggunaan komputer. CPU load dapat di deteksi melalui *command* "top" di sistem operasi linux/unix, yang mana penggunaan dari CPU itu sendiri nantinya akan dinyatakan dalam bentuk persen (%). Sama dengan RAM *Load*, untuk mendapatkan presentase penggunaan CPU load dibutuhkan command “awk” untuk parsing agar didapatkan hasil yang diinginkan.

c. Mendapatkan nilai Disk Capacity Usage : Selain RAM *Load* dan CPU *Load*, kapasitas hardisk juga sangat berpengaruh dalam performa sebuah *server*. Kapasitas hardisk yang tinggi bahkan penuh dapat menyebabkan berhentinya semua

service yang berjalan dalam sebuah *server*. Di sistem operasi linux/unix dapat menggunakan *command* "df -kh" dengan kembali menambahkan *command* "awk" untuk melakukan parsing sesauai keinginan.

d. Mendapatkan nilai Disk Inodes Usage : Inode (Index Node) adalah konsep dasar dalam sistem file Linux dan UNIX. Setiap objek dalam sistem file diwakili oleh sebuah inode. Setiap inode diidentifikasi oleh nomor inode unik dalam sistem file. Inode juga dikenal sebagai nomor indeks. Semua informasi yang mengidentifikasi file dan atributnya disimpat di dalam inode. Jumlah Inode menunjukkan jumlah file dan folder pada disk. Kapasitas inode dapat di cek menggunakan perintah "df -i". Apabila Inode usage pada directory penuh maka sistem tidak akan bisa menambahkan file baru pada direktori tersebut. Sehingga selain harus mengamati disk capacity usage, Inode usage merupakan parameter penting yang perlu lakukan pengamatan secara periodik.

2.2 Kebutuhan Perangkat

Spesifikasi minimum perangkat keras yang akan digunakan sebagai server. CPU : 2 GHz dual core processor, VGA Card dengan minimum resolusi 1024x768, HDD : 50GB, RAM 2 GB

Kebutuhan perangkat lunak yang dibutuhkan untuk membangun program pada penelitian ini: Ubuntu Server 20.04 LTS, Database server (MySql Server), Python dipilih sebagai bahasa pemprograman yang akan digunakan untuk merancang tampilan antarmuka. Library yang akan digunakan pada implementasi sistem khususnya pada *realtime dashboard monitoring* adalah Pandas (digunakan untuk manipulasi dan analisis data) dan Dash (digunakan sebagai framework pembuatan grafik). Perl merupakan bahasa pemprograman yang bisa digunakan untuk melakukan parsing data text dari log data server yang dimonitoring. Mengurai file teks adalah salah satu alasan Perl menjadi alat penambangan data dan script yang baik. Bash (Bourne shell) digunakan sebagai bahasa pemprograman utama dari *backend* sistem yang dibuat.

2.3 Pengambilan Sample

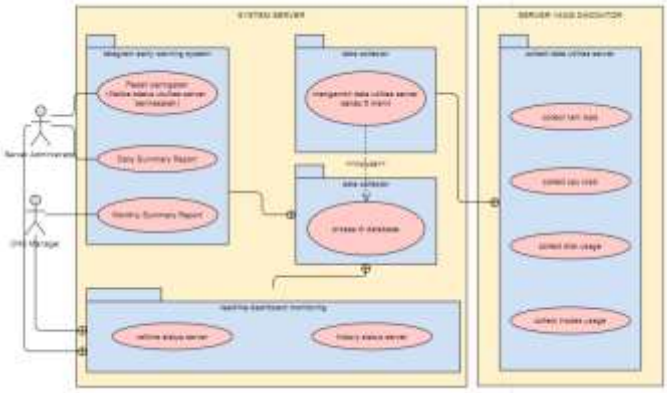
Data Utilisasi server sebelum implementasi system. Dikarenakan belum adanya data historis utilisasi server yang akan di monitoring, maka pengambilan sampel hanya dilakukan sekali sebelum sistem diimplementasikan, untuk memberikan gambaran kondisi server sebelum implementasi.

Penarikan/Pengambilan data dilakukan dengan menjalankan command utilisasi server di beberapa server yang akan di monitoring.

Jumlah sample akan diambil dari 3 Sample Server yang akan dimonitoring.

2.4 Perancangan Sistem

use case dalam pembuatan implementasi early warning system and realtime dashboard monitoring server menjelaskan peran Server administrator dan CNS manager sebagai user dan system server serta server yang dimonitor sebagai pelaksana proses



Berikut merupakan penjelasan beberapa activity yang ada pada use case diagram diatas:

a. a. Mengambil data di utilitas server (di sisi System server). Untuk mengambil data utulitas server di semua server yang terdaftar, akan dibuat script yang akan mengambil list server yang akan di monitoring dari database, lalu menggunakan bash script dan except script akan dijalankan pengambilan log utilitas server dengan meggungakan protocol ssh dari masing masing server yang di monitoring. Hasil log yang sudah di ambil akan di lakukan proses parsing untuk mengambil info info yang di perlukan untuk dimasukkan ke parameter parameter yang ada di database. Script yang dibuat akan dijadwalkan running setiap 5 menit.

b. b. Pesan Peringatan. Pengecekan terhadap semua parameter diambil dari rata rata nilai setiap parameter oleh script bash setiap 10 menit untuk mengambil data parameter 20 menit terakhir. Apabila nilai yang dihasilkan Parameter RAM Load, CPU Load, Disk Usage, dan Inodes Usage lebih dari atau sama dengan 90% maka dikategorikan sebagai *Spike Load*. Apabila selama 20 menit terdapat 3 Spike Load (Max 4 Spike Load) maka server tersebut dapat dikategorikan berstatus *WARNING* pada parameter RAM Load, CPU Load, Disk Usage, dan Inodes Usage Load, jika tidak

maka normal. Untuk status koneksi, apa bila terdapat 3 Status koneksi gagal selama 20 menit maka server dapat dikategorikan berstatus "*Connection Problem*" jika tidak maka normal. Apabila pada pengecekan yang dilakukan setiap 10 menit terdapat salah satu atau kedua status yang tidak normal, maka sistem akan mengirim pesan peringatan otomatis ke grup telegram agar dapat segera di lakukan corrective maintenance oleh server administrator.

c. Daily Summary Report. Salah satu fungsi dalam sistem monitoring ini adalah memberikan pesan periodik tiap hari ke grup telegram server admin agar dapat menjadi bahan analisa preventive maintenance terhadap kondisi aktual server. Pesan yang dikirim merupakan nilai rata rata tiap parameter yang ada dalam periode waktu 1 hari. Untuk mendapatkan informasi tersebut dibutuhkan script Bash dan PHP. Script bash berfungsi untuk menjalankan query dan membuat file text(.txt) berisi informasi yang menjadi pesan yang akan dikirim melalui telegram. Sementara dibutuhkan script PHP untuk mengirim file text(.txt) yang sudah di buat oleh script Bash untuk dikirim ke grup telegram memlalui bot Telegram.

d. Monthly Summary Report. Selain memberikan pesan periodik tiap 1 hari ke grup telegram, sistem ini juga akan memberikan pesan periodik rangkuman kondisi lebih detail server dalam 1 bulan. Selain menampilkan nilai rata rata tiap parameter, juga ditampilkan nilai maksimal dan minimal tiap parameter dalam satu bulan, dan juga jumlah pesan peringatan yang terjadi dan dikirim di dalam satu bulan. Untuk aktifitas kali ini script yang dibutuhkan juga sama dengan scipt untuk membuat rangkuman kondisi server tiap hari yang sebelumnya di bahas dengan beberapa tambahan parameter.

e. Realtime Status Server. Melihat status realtime server merupakan salah satu fungsi utama dari realtime dashboard monitoring. Dengan adanya pengambilan status utilitas server tiap 5 menit, dan data di simpan di database. Maka data tersebut dapat diolah dan ditampilkan dalam bentuk grafik pada web, untuk memudahkan melakukan analisa pada status aktual server.

f. History Status Server. Selain menampilkan status realtime server, data utilitas yang juga dapat disimpan dan dapat di buka pada menu History Status server pada dashboard. Pada menu ini user dapat melihat status utulitas server di waktu tertentu yang bisa di tentukan sendiri untuk melakukan analisa pada keadaan server di masa lampau.

2.5 Teknis Analisis

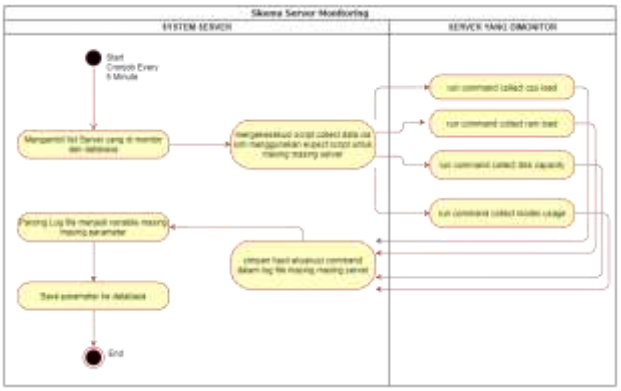
Pada proses analisis dilakukan untuk menciptakan solusi automasi pada pengecekan parameter setiap server yang akan dimonitoring. Solusi yang di harapkan adalah kostumisasi yang tinggi terhadap masing masing parameter yang diinginkan serta meminimalisir konfigurasi yang di perlukan pada sisi server yang akan dimonitoring. Dengan itu ditemukan solusi automasi menggunakan bash dan expect sehingga untuk dapat mengambil data parameter yang dibutuhkan hanya memerlukan ssh akses ke server (ip address, username, password dan port). Dengan solusi automasi tersebut server yang akan di monitoring tidak memerluka konfigurasi apapun disisi server. Tentu hal in akan mempermudah control terhadap masalah yang akan muncul pada sistem yang akan di implementasikan.

Selain pada pengambilan parameter status server, diperlukan juga perancangan terhadap bagaimana menyiapkan *report alert* yang cepat dan tepat apabila status server terindikasi mengalami masalah. Selain itu juga di perlukan pembuatan laporan status server secara berkala setiap hari dan tiap bulannya. Dari sisi visualisasi data pada dashboard realtime dan historical, diharapkan data dan dashboard dapat mendukung user dalam menggunakan tool yang diimplementasikan agar dapat sesuai fungsi yang diharapkan.

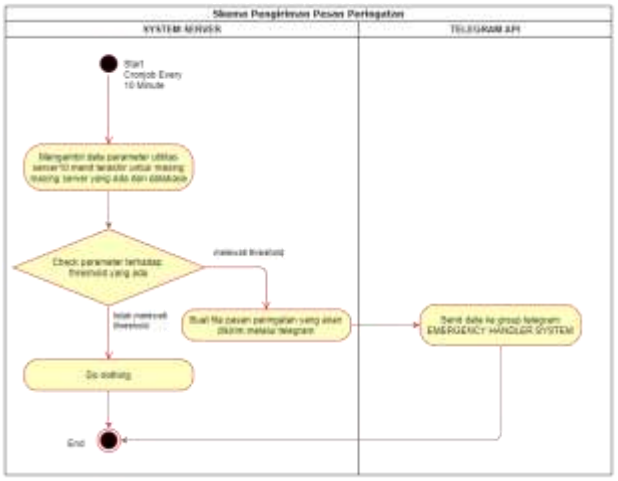
3. HASIL DAN PEMBAHASAN

3.1 Activity Diagram

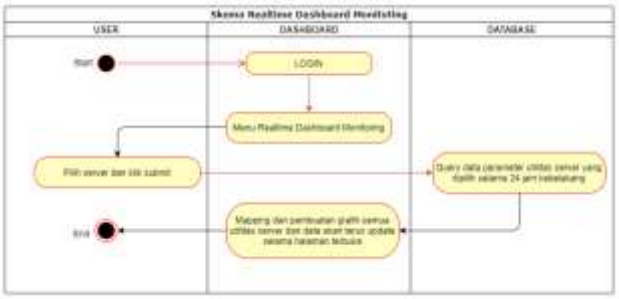
a. Mengambil data di utilitas server (di sisi System server



b. Pesan Peringatan



c.Real Time Status server



3.2 Implementasi Database

Table tb_server

Table tb_dashboard berfungsi untuk menyimpan list server lengkap dengan IP Address, username, dan port untuk SSH connection.

Table tb_cpu_ram_load

Table tb_cpu_ram_load berfungsi untuk menyimpan data utilitas masing-masing server yang di monitoring seperti cpu dan ram load. Table ini juga menyimpan informasi status ssh ke server yang di monitoring.

Table tb_disk_capacity

Table tb_disk_capacity berfungsi untuk menyimpan data utulitas disk capacity masing masing server.

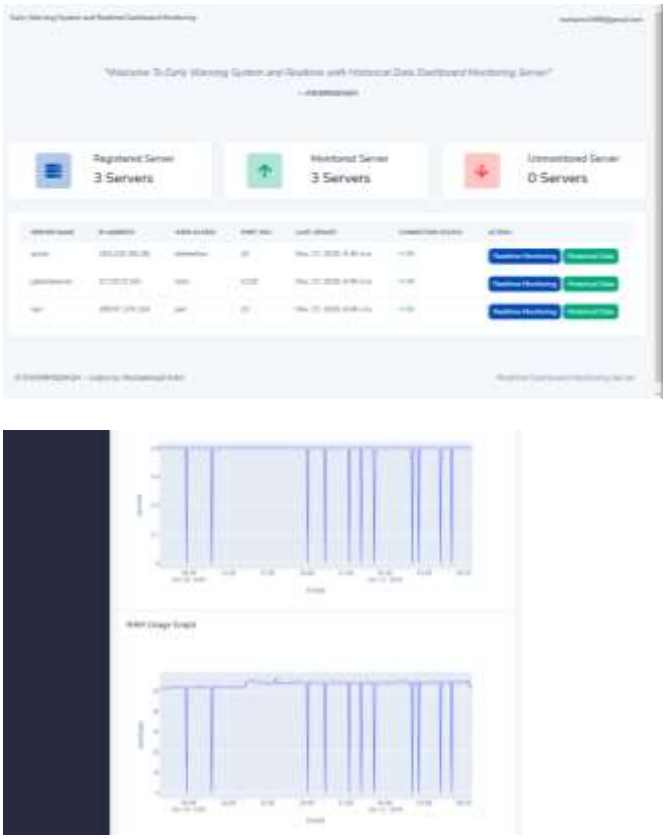
Table tb_inodes_usage

Table tb_inodes_usage berfungsi untuk menyimpan data utilitas inodes_usage masing masing server.

Table tb_report

Table tb_alert_hist berfungsi untuk menyimpan data bot telegram dan grup telegram yang akan dikirimkan data Report baik alert, daily, ataupun Monthly.

3.3 Tampilan Realtime Dashboard Monitoring



Gambar 3.1 Realtime Dashboard

Pengambilan Data Utilitas Server
Pada proses pengambilan data utilitas server, dibutuhkan beberapa script automasi terpisah antara lain:

1. Script runcollector.sh

Script runcollecttor.sh merupakan script yang di tulis menggunakan bahasa Bash shell script yang berfungsi untuk mengambil list server yang terdaftar pada database dan mememanggil script bash selanjutnya yaitu collectlog.sh untuk masing masing server secara paralel.

- Mengambil data list server dari database.
- Memanggil collectlog.sh secara pararel.

```
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )"

my_touch $DIR/etc/server.list
/usr/bin/mysql -u root -pLast_12321 -
e "use db_ewsmsdash;select servername,ipaddress,username,password,port
 from tb_server" | awk 'BEGIN{OFS=";"} {print $1,$2,$3,$4,$5}' | grep -
v "servername;ipaddress" >> $DIR/etc/server.list

for i in `cat $DIR/etc/server.list`
do
    SERVERNAME=`echo $i | awk -F';' '{print $1}'`
    IPADDR=`echo $i | awk -F';' '{print $2}'`
    USERNAME=`echo $i | awk -F';' '{print $3}'`
    PASSWD=`echo $i | awk -F';' '{print $4}'`
    PORT=`echo $i | awk -F';' '{print $5}'`
    $DIR/collectlog.sh $SERVERNAME $IPADDR $USERNAME $PASSWD $PORT&
Done
```

2. Script collectlog.sh

Script collectlog.sh merupakan script yang ditulis menggunakan bahasa Bash Shell script yang berfungsi untuk mengambil log dari server yang di monitoring dengan script lain yaitu collector.exp. Hasil dari proses ambil log dari collector.exp akan dilakukan proses parsing pada collectlog.sh dan parserlog.pl untuk selanjutnya disimpan pada database.

- Menjalankan collector.exp dan simpan log.
- Melakukan parser log dan running parser.pl Import hasil parser ke database.

```
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )"
SERVERNAME=$1
IPADDR=$2
USERNAME=$3
PASSWD=$4
PORT=$5
NOW=`date "+%FT%H:%M:00"`
my_touch $DIR/log/${SERVERNAME}.log
```

3. Script collector.exp

Script collector.exp merupakan script yang ditulis menggunakan library expect pada bash shell scripting. Script ini berfungsi untuk melakukan akses ssh dan menjalankan beberapa command pada server yang akan dimonitoring lalu outputnya dijadikan log.

```
/usr/bin/expect $DIR/collector.exp $IPADDR $USERNAME $PASSWD $PORT >> $
DIR/log/${SERVERNAME}.log
```

4. Script parserlog.pl

Script parserlog.pl merupakan script yang ditulis menggunakan bahasa pemprograman perl. Script ini dibutuhkan untuk melakukan parsing pada data log, khususnya data disk usage dan inodes usage. Berikut adalah log perbandingan log sebelum dan hasil parsing dari parserlog.pl.

```
/usr/bin/expect $DIR/collector.exp $IPADDR $USERNAME $PASSWD $PORT >> $
DIR/log/${SERVERNAME}.log
```

Pesan peringatan telegram

Berikut adalah tahapan pengambilan data untuk menentukan apakah parameter pada server tersebut mengalami masalah atau tidak:

a. Parameter CPU, RAM, dan Connection Status

Dilakukan Query yang berfungsi untuk mengambil data server 20 menit kebelakang dan menggolongkan pada kondisi spike apabila memiliki value diata 90, apabila terjadi spike 3 kali pada waktu 20 menit maka parameter tersebut digolongkan sedang dalam keadaan warning.

```
######## MEM LOAD & CPULOAD ########
MEMLOAD=`cat $DIR/log/${SERVERNAME}.log | grep "MemLoad:" | egrep -
v "free|awk" | awk '{print $NF}' | sed 's/%//g' | tr -d '\r'`
CPULOAD=`cat $DIR/log/${SERVERNAME}.log  | grep "Cpu" | grep -
v top | awk '{print $2}' | sed 's/%us,//g'`
```

b. Disk Usage

Dilakukan Query yang berfungsi untuk mengambil data percentage disk capacity server 20 menit kebelakang dan menggolongkan pada kondisi spike apabila memiliki value diata 90, apabila terjadi spike 3 kali pada waktu 20 menit maka parameter tersebut digolongkan sedang dalam keadaan warning.

c. Inodes Usage

Dilakukan Query yang berfungsi untuk mengambil data percentage inodes usage server 20 menit kebelakang dan menggolongkan pada kondisi spike apabila memiliki value diatas 90, apabila terjadi spike 3 kali pada waktu 20 menit maka parameter tersebut digolongkan sedang dalam keadaan warning.

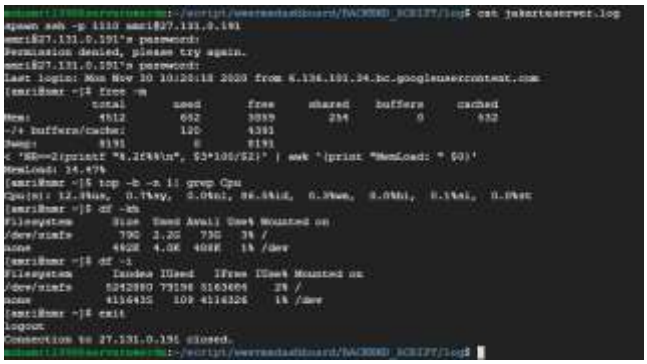
```
######## DISK USAGE & INODES USAGE ########
cd $DIR/log/
/usr/bin/perl $DIR/parserlog.pl ${SERVERNAME}.log $DIR $NOW

my_touch $DIR/importdb/${SERVERNAME}_diskusage.csv
cat $DIR/importdb/${SERVERNAME}_diskusage.tmp | egrep -
v "df|exit|Filesystem|Cpu" | sed 's/,$//g;s/%//g' >> $DIR/importdb/${SE
RVERNAME}_diskusage.csv
rm $DIR/importdb/${SERVERNAME}_diskusage.tmp

my_touch $DIR/importdb/${SERVERNAME}_inodesusage.csv
cat $DIR/importdb/${SERVERNAME}_inodesusage.tmp | egrep -
v "df|exit|Filesystem" | sed 's/,$//g;s/%//g;s/,-
,/,0,/g' >> $DIR/importdb/${SERVERNAME}_inodesusage.csv
rm $DIR/importdb/${SERVERNAME}_inodesusage.tmp
```

4.2 Pengujian Sistem Realtime dan Historical Data

Pengecekan data update terakhir di setiap server. Pada pengecekan ini dilakukan perbandingan dari

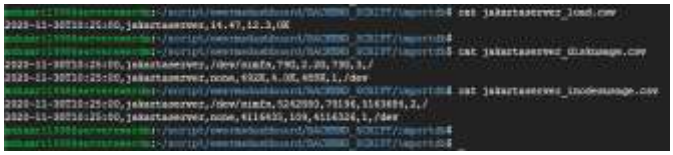
aktual data, hasil parsing data, dan tampilan dashboar data di waktu yang sama.

a.Aktual Data



Gambar 3.2 Tampilan Aktual data pada log

b.Data hasil parsing



Gambar 3.3 Tampilan data hasil parsing

c.Tampilan pada dashboard

Gambar 3.4 Tampilan dashboard CPUload

Gambar 3.5 Tampilan dashboard memory load

Gambar 3.6 Tampilan dashboard Disk capacity

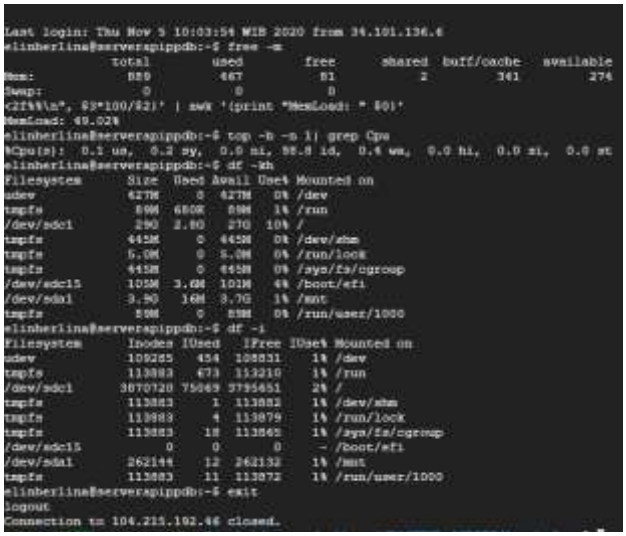
Gambar 3.7 Tampilan dashboard Inodes usage

3.5 Dampak implementasi sistem pada server termonitor

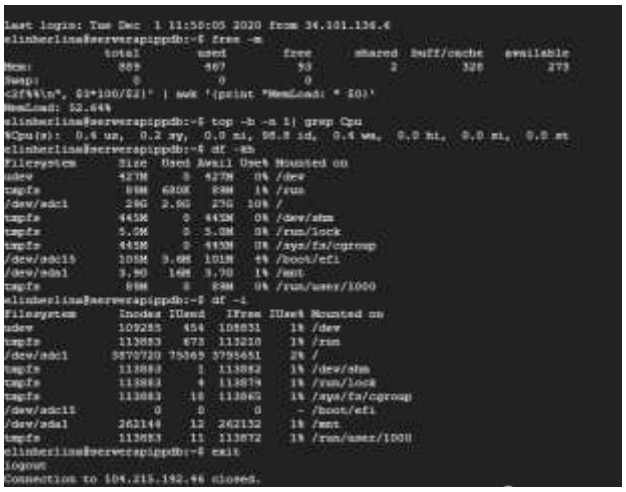
Berikut merupakan dampak implementasi sistem pada server termonitor

Servername	Parameter	Sebelum	Sesudah	Selisih
azure	Mem Load	49.02%	52.64%	3.62%
	CPU Load	0.1%	0.4%	0.3%
vps	Mem Load	23.71%	21.49%	-2.22%
	CPU Load	0.1%	0.1%	0%

Tabel 3.1 Hasil pengujian performance server



Gambar 3.8 Status server tanpa monitoring system



Gambar 3.8 Status server setelah monitoring system

PENUTUP

Simpulan

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut :

1. Sistem yang diimplementasi dapat memberikan notifikasi peringatan dini sebelum *server* mengalami penurunan perfomance.
2. Sistem yang diimplementasi dapat memberikan informasi realtime kondisi *server* dan menyimpan data kondisi tersebut untuk kebutuhan analisa.
3. Sistem yang diimplementasi tidak mengurangi performa *server* yang di *monitoring* dengan selisih perbandingan performa sebelum dan sesudah implementasi sistem pada mem load yaitu (-2.22%-3.62%) dan CPU load yaitu (0%-0.3%).

Saran

Berikut adalah saran untuk pengembangan lebih lanjut terhadap penelitian ini sebagai berikut:

1.Diharapkan untuk pengembangan kedepannya selain memberikan info peringatan dini pada server bermasalah, sistem juga dapat memberikan *automation preventive action* sehingga masalah bisa langsung diatasi oleh sistem dan status server dapat menjadi normal kembali secara otomatis.

2. Sistem dapat ditambah parameter yang akan di monitor untuk melengkapi analisa performance server seperti delay, latency dan troughput server.

3. Diharapkan untuk kedepanya dengan data yang semakin berkembang semakin banyak juga visualisasi grafik data yang bisa di tampilkan pada dashboard

DAFTAR PUSTAKA

[1] Agus Mulyanto. 2009. Yogyakarta: Pustaka Pelajar *Sistem Informasi Konsep Dan Aplikasi*. Yogyakarta: Pustaka Pelajar.

[2]Atmojo, Yohanes Priyo. 2018. “Bot Alert Snort Dengan Telegram Bot API Pada Intrusion Detection System : Studi Kasus IDS Pada Server Web.” : 176–80.

[3] Bungun, Burhan. 2019. *Penelitian Kualitatif*. Jakarta: Kencana Prenda Media Group.

[4] Django Software Foundation, 2020. Django Documentation - Design Philosophies. [Online]

[5] Hartono, Jogiyanto. 2005. *Analisis Dan Desain Sistem Informasi Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis III*. Yogyakarta: ANDI.

[6]Mokhamad Nizar Zulfikar. 2019. “Sistem Reporting Keamanan Pada Jaringan Cloud Computing Melalui Bot Telegram Dengan Menggunakan Teknik Intrussion Detection and Prevention System.” 5(2): 49–57.

[7] Panjaitan, Febriyanti et al. 2019. “Pemanfaatan Notifikasi Telegram Untuk Monitoring Jaringan Febriyanti.” *Simetris : Jurnal Teknik Mesin, Elektro dan Ilmu Komputer* 10(2): 725–32.

[8] R.H. Sianipar & Hamzan Wadi. 2015. *Pemrograman Python (Teori Dan Implementasi)*. Rifai, Bakhtiar, Nanang Nuryadi, and Amarulloh Ripai. 2019.