

**SKRIPSI**  
**SISTEM MONITORING DAN ALERTING**  
**PADA IMPLEMENTASI HIGH AVAILABILITY WEB SERVER**  
**DENGAN CLUSTER DATABASE BERBASIS DOCKER SWARM**



**ARISMAN**

**NIM : 185411140**

**PROGRAM STUDI INFORMATIKA**  
**PROGRAM SARJANA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**  
**2023**

**SKRIPSI**  
**MONITORING SISTEM DAN ALERTING**  
**PADA IMPLEMENTASI HIGH AVAILABILITY WEB SERVER**  
**DENGAN CLUSTER DATABASE BERBASIS DOCKER SWARM**

**Diajukan sebagai salah satu syarat untuk menyelesaikan studi**

**Program Sarjana**  
**Program Studi Informatika**  
**Fakultas Teknologi Informasi**  
**Universitas Teknologi Digital Indonesia**  
**Yogyakarta**

**Disusun Oleh**

**ARISMAN**

**NIM : 185411140**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**PROGRAM SARJANA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**  
**2023**

## **PERNYATAAN KEASLIAN SKRIPSI**

Dengan ini saya menyatakan bahwa naskah skripsi ini belum pernah diajukan untuk memperoleh gelar Sarjana Komputer di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara sah diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 24 Januari 2023



Arisman

NIM: 185411140

## **HALAMAN PERSEMBAHAN**

Puji syukur saya panjatkan kehadiran Allah SWT yang telah memberikan rahmat, taufik dan hidayah-Nya kepada kita semua. Skripsi ini saya persembahkan kepada :

- Allah SWT Pencipta semesta alam yang telah memberiku hidup dan berkah dan rizkinya
- Orang Tua yang telah mendoakan dan memberi dukungan penuh
- Adi Kusjani, S.T., M.Eng. yang telah menjadi dosen pembimbing yang sabar dalam membimbing saya.

## **MOTTO**

“Kamu tidak harus menjadi hebat untuk memulai, tetapi kamu harus mulai untuk menjadi hebat.” – Zig Ziglar.

## KATA PENGANTAR

Puji syukur kita panjatkan kehadiran Allah SWT, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan karya skripsi yang berjudul “Sistem Monitoring dan Alerting pada Implementasi High Availability Web Server dengan Cluster Database Berbasis Docker Swarm” dengan sebaik-baiknya. Skripsi ini dibuat untuk memenuhi syarat menyelesaikan Program Studi Informatika Program Sarjana Fakultas Teknologi Informasi di Universitas Teknologi Digital Indonesia Yogyakarta.

Semoga skripsi ini tidak hanya bermanfaat bagi penulis, namun juga bagi pihak-pihak yang membacanya. Penulis menyadari bahwa didalam penulisan skripsi ini masih terdapat kekurangan atau belum sempurna. Oleh karena penulis mengharapkan segala kritik dan saran dari pembaca untuk memberikan masukan-masukan yang bersifat membangun untuk kesempurnaan didalam penulisan ini.

Dalam penyelesaian skripsi ini penulis tidak terlepas dari bantuan, bimbingan, petunjuk, dan saran dari semua pihak yang telah membantu dalam penyelesaian skripsi ini. Oleh karena itu penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Ir. Totok Suprawoto, M.M., M.T. selaku Rektor Universitas Teknologi Digital Indonesia.
2. Dini Fakta Sari, S.T., M.T. selaku ketua Program Studi Informatika Universitas Teknologi Digital Indonesia.

3. Adi Kusjani, S.T., M.Eng. selaku dosen pembimbing skripsi yang senantiasa memberikan bimbingan dan arahan dalam mengerjakan skripsi hingga selesai
4. Dosen Program Studi Informatika Universitas Teknologi Digital Indonesia yang telah memberikan ilmu.
5. Orang tua dan keluarga yang selalu memberikan semangat dan dorongan dalam penyelesaian skripsi ini
6. Nur Alifah yang selalu memberikan semangat dan selalu menemani dalam pengerjaan skripsi

Yogyakarta, Januari 2023

Arisman

## DAFTAR ISI

SKRIPSI.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN PERSEMBAHAN.....	v
MOTTO.....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xi
INTISARI.....	xv
ABSTRACT .....	xvi
BAB 1 PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	2
1.3    Ruang Lingkup.....	2
1.4    Tujuan Penelitian .....	2
1.5    Manfaat Penelitian .....	2
1.6    Sistematika Penulisan.....	3
BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI .....	5
2. 1    Tinjauan Pustaka .....	5
2. 2    Dasar Teori.....	7
2.2.1.    Sistem Monitoring.....	7
2.2.2.    Prometheus .....	7
2.2.3.    Exporter.....	8
2.2.4.    Grafana.....	8
2.2.5.    Bot Telegram.....	8
2.2.6.    Stress-ng .....	9
2.2.7.    Linux htop .....	9
BAB 3 METODE PENELITIAN.....	10
3.1.    Bahan / Data .....	10
3.2.    Peralatan .....	10
3.2.1.    Kebutuhan Perangkat Keras .....	10
3.2.2.    Kebutuhan perangkat Lunak .....	10



3.3.	Pemodelan yang digunakan.....	11
3.4.	Rencana pengujian .....	12
BAB 4 IMPLEMENTASI DAN PEMBAHASAN.....		13
4.1.	Implementasi .....	13
4.1.1.	Prometheus node-exporter .....	13
4.1.2.	Prometheus cadvisor-exporter.....	15
4.1.3.	Proxysql metrics exporter .....	16
4.1.4.	Nginx vts exporter.....	16
4.1.5.	Prometheus server .....	20
4.1.6.	Grafana .....	23
4.1.7.	Alerting .....	50
4.2.	Pengujian.....	58
4.2.1.	Install stress-ng.....	58
4.2.2.	Pengujian Data Grafana .....	59
4.2.3.	Pengujian Notifikasi.....	60
BAB 5 PENUTUP.....		62
5.1.	Kesimpulan .....	62
5.2.	Saran.....	62
DAFTAR PUSTAKA .....		64

## DAFTAR GAMBAR

Gambar 3.1.	Pemodelan Sistem .....	11
Gambar 4.1.	Docker-compose.yml node exporter .....	14
Gambar 4.2.	Docker-compose.yml .....	16
Gambar 4.3.	Proxysql.cnf .....	16
Gambar 4.4.	Openssl tools .....	17
Gambar 4.5.	Unduh nginx-module-vts .....	17
Gambar 4.6.	Re-compile nginx .....	18
Gambar 4.7.	Make install nginx.....	18
Gambar 4.8.	Nginx.cnf.....	19
Gambar 4.9.	Nginx-vts Dasbor .....	19
Gambar 4.10.	Perintah install vts-exporter .....	19
Gambar 4.11.	Docker-compose.yml .....	20
Gambar 4.12.	Prometheus.yml.....	21
Gambar 4.13.	Deploy docker-compose.....	21
Gambar 4.14.	Docker service ls .....	22
Gambar 4.15.	Prometheus Dasbor .....	22
Gambar 4.16.	Prometheus target status.....	23
Gambar 4.17.	Docker-compose.yml .....	24
Gambar 4.18.	Perintah deploy Grafana.....	24
Gambar 4.19.	Membuat data sources Grafana .....	25
Gambar 4.20.	Grafana Dasbor list .....	26
Gambar 4.21.	Variabel nginx Dasbor .....	27
Gambar 4.22.	Metric visualization Server Connections .....	27
Gambar 4.23.	Metrics visualization Server Cache.....	28
Gambar 4.24.	Metrics visualization Server Requests .....	29
Gambar 4.25.	Metrics visualization Server Traffic.....	29
Gambar 4.26.	Metrics visualization Upstream Request.....	30
Gambar 4.27.	Metrics visualization Upstream Traffic .....	30
Gambar 4.28.	Dasbor monitoring Nginx .....	31
Gambar 4.29.	Variabels node status.....	31
Gambar 4.30.	Metrics query panel node resource overview.....	33
Gambar 4.31.	Node resource table mapping.....	33
Gambar 4.32.	Metrics query panel uptime.....	33

Gambar 4.33.	Metrics visualization Disk Usage.....	34
Gambar 4.34.	Disk usage table mapping .....	35
Gambar 4.35.	Metrics visualization Memory .....	36
Gambar 4.36.	Metrics query panel cpu basic.....	36
Gambar 4.37.	Metrics visualization Network Packet .....	37
Gambar 4.38.	Metrics visualization Network Traffic .....	38
Gambar 4.39.	Dasbor monitoring Node Status .....	38
Gambar 4.40.	Variabel docker Dasbor.....	39
Gambar 4.41.	Metrics visualization CPU Usage .....	40
Gambar 4.42.	Panel visualisasi Memory usage .....	40
Gambar 4.43.	Panel visualisasi traffic upload.....	41
Gambar 4.44.	Panel visualisasi traffic download.....	41
Gambar 4.45.	Panel visualisasi Container Info .....	42
Gambar 4.46.	Dasbor docker status .....	42
Gambar 4.47.	Variabels Dasbor proxysql.....	43
Gambar 4.48.	Metrics query backend connection.....	43
Gambar 4.49.	Metrics visualization <i>Connection Backend Status</i> .....	44
Gambar 4.50.	Metrics visualization Connection Statistics .....	45
Gambar 4.51.	Metrics visualization active transaction .....	45
Gambar 4.52.	Panel visualisasi uptime proxysql .....	45
Gambar 4.53.	Panel visualisasi connection overview .....	46
Gambar 4.54.	Panel visualisasi database status .....	47
Gambar 4.55.	Panel visualisasi latency.....	48
Gambar 4.56.	Panel visualisasi Throughput Network .....	48
Gambar 4.57.	Panel visualisasi Connection Backend Throughput.....	49
Gambar 4.58.	Proxysql Status Dasbor .....	50
Gambar 4.59.	Membuat bot telegram. ....	51
Gambar 4.60.	Menambahkan Bot ke dalam group .....	51
Gambar 4.61.	Mencari chat id grup telegram .....	52
Gambar 4.62.	Telegram chat id.....	52
Gambar 4.63.	Grafana contact point .....	53
Gambar 4.64.	Template myalert .....	53
Gambar 4.65.	Template mymessage.....	54
Gambar 4.66.	Custom notification message .....	54

Gambar 4.67.	alert rule node cpu.....	55
Gambar 4.68.	alert rule memory usage.....	56
Gambar 4.69.	alert rule disk node.....	57
Gambar 4.70.	alert rule container cpu.....	57
Gambar 4.71.	Alert rule status .....	58
Gambar 4.72.	Install stress-ng.....	58
Gambar 4.73.	Perintah Stress-ng CPU Test.....	59
Gambar 4.74.	Perintah Stress-ng Memory Test.....	59
Gambar 4.75.	Notifikasi telegram.....	61

## DAFTAR TABEL

Tabel 1.1.	Tinjauan Pustaka .....	6
Tabel 1.2.	Tabel pengujian Grafana .....	60
Tabel 1.3.	Tabel pengujian htop.....	60
Tabel 1.4.	Tabel Pengujian notifikasi.....	61

## INTISARI

Pada era revolusi industri 4.0 ini, Sistem informasi mempunyai peranan yang sangat penting bagi semua perusahaan. Sehingga ini perlu dilakukan pengawasan dan pengecekan berkala agar sistem informasi tersebut selalu prima agar dapat meminimalisir down time yang bisa jadi menghambat proses bisnis. Agar pengawasan lebih efektif, maka diperlukan sebuah sistem monitoring yang didukung dengan alerting sehingga memungkinkan untuk memantau kinerja sistem pada sistem menggunakan smartphone(telegram).

Pada penelitian ini akan dibangun sistem monitoring dan alerting menggunakan Grafana, Prometheus dan Telegram. Dalam melakukan scrape data prometheus dibantu oleh Prometheus Exporter. Exporter yang digunakan yaitu Node exporter untuk mengambil data dari node atau hardware, Docker cadvisor exporter untuk mengambil data dari docker swarm dan nginx VTS exporter untuk mengambil data dari nginx. Data tersebut kemudian akan disimpan ke dalam database Prometheus kemudian ditampilkan ke prometheus web-gui hingga nantinya grafana akan me-request metrics menggunakan query yang dibuat lalu data ditampilkan pada dashboard Grafana dalam bentuk visualisasi. Untuk mempermudah dalam membaca visualisasi grafana, dasbor dibuat menjadi 4 sesuai dengan kategori objek yang di monitor. Setelah membuat dasbor visualisasi, kemudian dibuat rule alerting menggunakan query dan kondisi tertentu. Jika rule ini terpenruhi, maka grafana akan mengirimkan notifikasi ke telegram.

Dari hasil penelitian ini dapat disimpulkan bahwa sistem monitoring dan alerting berhasil dibangun pada dashboard Grafana yang terpusat dengan data yang cukup akurat, karena data yang ditampilkan pada dashboard grafana mendekati dengan data yang ada pada tools “htop” (task manager ubuntu). Untuk sistem alerting juga dapat berjalan dengan baik, dimana sistem ini dapat mengirimkan notifikasi dengan rata rata waktu 25,5 detik ke telegram.

**Kata kunci :** Grafana, Prometheus, Sistem monitoring, Sistem Alerting.

## ABSTRACT

In the era of the industrial revolution 4.0, information systems have a very important role for all companies. This needs to be monitored and checked periodically so that the information system is always primed so that it can minimize down time that could hinder business processes. For supervision to be more effective, a monitoring system is needed which is supported by alerting so that it is possible to monitor system performance on systems that use smartphones (telegrams).

In this research a monitoring and alerting system will be built using Grafana, Prometheus and Telegram. In scraping data, Prometheus is assisted by Prometheus Exporter. The exporters used are the Node exporter to retrieve data from nodes or hardware, the Docker cadvisor exporter to retrieve data from docker swarm and the nginx VTS exporter to retrieve data from nginx. The data will then be stored in the Prometheus database and then displayed on the Prometheus web-gui so that later Grafana will request metrics using the query made and then the data will be displayed on the Grafana dashboard in the form of a visualization. To make it easier to read the grafana visualization, the dashboard is made into 4 according to the category of objects being monitored. After creating the dashboard visualization, an alerting rule is created using certain queries and conditions. If this rule is fulfilled, grafana will send a notification to Telegram.

From the results of this research it can be concluded that the monitoring and alerting system was successfully built on the centralized Grafana dashboard with fairly accurate data, because the data displayed on the grafana dashboard is close to the data in the "htop" tool (ubuntu task manager). can also run well, where this system can send notifications with an average time of 25.5 seconds to telegram.

Keywords: Grafana, Prometheus, monitoring system, alerting system