

**PENERAPAN SISTEM TERDISTRIBUSI PADA APLIKASI  
BERBASIS WEBSITE "SIMPLIFYPOS" UNTUK MENJAGA  
KETERSEDIAAN, SKALABILITAS DAN KEANDALAN**

**SKRIPSI**

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana  
Program Sistem Informasi



**Disusun Oleh**

**MAHFUDZ AJI WICAKSONO M. NUR**

**21.22.2455**

Kepada

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2023**

**PENERAPAN SISTEM TERDISTRIBUSI PADA APLIKASI  
BERBASIS WEBSITE "SIMPLIFYPOS" UNTUK MENJAGA  
KETERSEDIAAN, SKALABILITAS DAN KEANDALAN**

**SKRIPSI**

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana  
Program Sistem Informasi



**Disusun Oleh**

**MAHFUDZ AJI WICAKSONO M. NUR**

**21.22.2455**

Kepada

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2023**

**HALAMAN PERSETUJUAN**

**SKRIPSI**

**PENERAPAN SISTEM TERDISTRIBUSI PADA APLIKASI BERBASIS  
WEBSITE "SIMPLIFYPOS" UNTUK MENJAGA KETERSEDIAAN,  
SKALABILITAS DAN KEANDALAN**

yang disusun dan diajukan oleh

**Mahfudz Aji Wicaksono M. Nur**

**21.22.2455**

telah disetujui oleh Dosen Pembimbing Skripsi

pada tanggal 25 Juli 2023

**Dosen Pembimbing,**

  
**Anggit Dwi Hartanto, M.Kom.**

**NIK. 190302163**

**HALAMAN PENGESAHAN**

**SKRIPSI**

**PENERAPAN SISTEM TERDISTRIBUSI PADA APLIKASI BERBASIS  
WEBSITE "SIMPLIFYPOS" UNTUK MENJAGA KETERSEDIAAN,  
SKALABILITAS DAN KEANDALAN**

yang disusun dan diajukan oleh  
**Mahfudz Aji Wicaksono M. Nur**

**21.22.2455**

telah dipertahankan di depan Dewan Penguji  
pada tanggal 25 Juli 2023

**Susunan Dewan Penguji**

**Nama Penguji**

**Tanda Tangan**

**Atik Nurmasani, S.Kom., M.Kom**  
NIK. 190302354

**Muhammad Kopravi, S.Kom., M.Eng**  
NIK. 190302454

**Anggit Dwi Hartanto, M.Kom**  
NIK. 190302163

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Komputer  
tanggal 25 Juli 2023

**DEKAN FAKULTAS ILMU KOMPUTER**



**Hanif Al Fatta, S. Kom., M.Kom.**  
NIK. 190302096

## HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan dibawah ini,

Nama mahasiswa : Mahfudz Aji Wicaksono M. Nur

NIM : 21.22.2455

Menyatakan bahwa Skripsi dengan judul berikut:

**PENERAPAN SISTEM TERDISTRIBUSI PADA APLIKASI BERBASIS WEBSITE "SIMPLIFYPOS" UNTUK MENJAGA KETERSEDIAAN, SKALABILITAS DAN KEANDALAN**

Dosen pembimbing : Anggit Dwi Hartanto, M.Kom.

1. Karya tulis ini adalah benar-benar **ASLI** dan **BELUM PERNAH** diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian **SAYA** sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab **SAYA**, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini **SAYA** buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka **SAYA** bersedia menerima **SANKSI AKADEMIK** dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 25 Juli 2023

Yang menyatakan,



Mahfudz Aji Wicaksono M. Nur

## HALAMAN PERSEMBAHAN

Karya ini saya persembahkan untuk Bapak (alm), Ibu, adik dan keluarga saya atas doa, semangat dan dukungan yang diberikan selama ini demi keberhasilan saya.

### MOTTO

“Harus berprasangka baik terhadap keputusan Allah SWT”

“Terus mengejar impian dan tidak akan menyerah”



## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa Allah SWT, atas ijin, karunia dan perkenannya, penulis dapat menyelesaikan skripsi dengan judul “Penerapan Sistem Terdistribusi Pada Aplikasi Berbasis Website Simplifypos Untuk Menjaga Ketersediaan, Skalabilitas dan Keandalan” ini dengan sebaik baiknya.

Dalam melaksanakan skripsi tersebut, penulis mendapatkan pengalaman, pengetahuan dan ilmu yang nantinya diharapkan mampu diterapkan dan bermanfaat, sehingga penulis mengucapkan terimakasih ke semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini.

Ucapan terimakasih ini penulis tujukan kepada:

1. Anggit Dwi Hartanto, M.Kom., selaku dosen pembimbing yang telah membimbing penulis dalam menyusun skripsi ini.
2. Ibu saya, Dra. Sri E. dan saudara saya, Mukhlis Agung Laksono yang telah memberi doa, dukungan dan semangat.
3. Seluruh dosen dan karyawan Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta.
4. Astri, yang telah meminjamkan saya server yang saya gunakan sebagai server pemantau. Titis, yang telah meminjamkan saya server yang saya gunakan sebagai server *worker2*. Kartika, yang telah meminjamkan saya server yang saya gunakan sebagai server *load balancer*. Aga, yang telah meminjamkan saya server yang saya gunakan sebagai server *backup*. Kalian sangat luar biasa.
5. Seluruh teman, saudara dan pihak yang telah memberikan dukungan selama pelaksanaan skripsi ini.

Dengan ini, penulis menyampaikan bahwa penulis terbuka atas setiap saran yang membangun baik dari sisi laporan ataupun dari sisi sistem yang dibangun agar kedepannya penulis dapat menyempurnakan menjadi bentuk yang lebih baik.

Yogyakarta, 25 Juli 2023

Penulis

## DAFTAR ISI

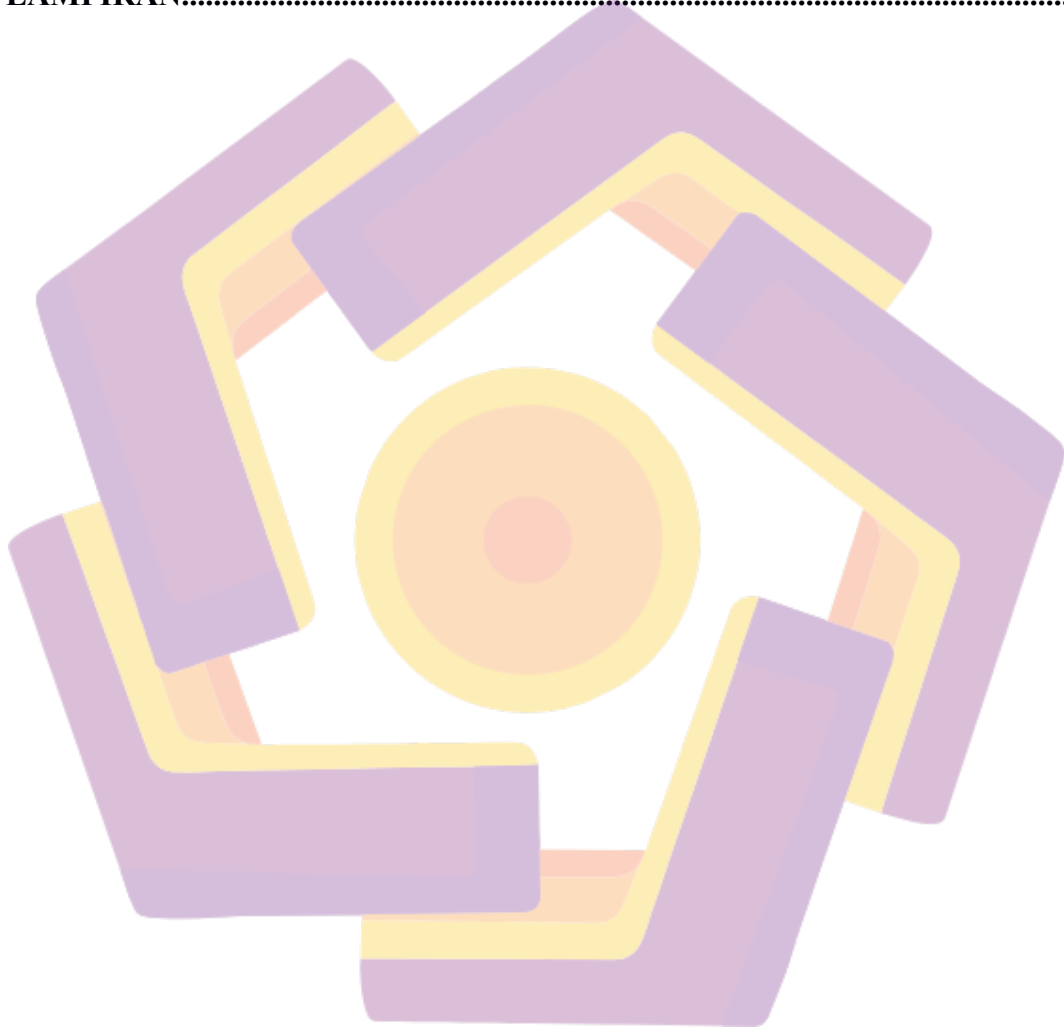
<b>HALAMAN JUDUL.....</b>	<b>i</b>
<b>HALAMAN PERSETUJUAN.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....</b>	<b>iv</b>
<b>HALAMAN PERSEMBAHAN.....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>xv</b>
<b>INTISARI.....</b>	<b>xvi</b>
<b>ABSTRACT.....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 Studi Litaratur.....	5
2.2 Dasar Teori.....	10
a. Sistem terdistribusi.....	10
b. Ketersediaan, Skalabilitas dan Keandalan Sistem .....	10
c. Kontainer dan docker.....	11



d. Docker compose.....	13
e. Docker registry.....	14
f. Docker image.....	14
g. Sistem Pemantauan ( <i>Monitoring System</i> ).....	14
h. Alertmanager.....	17
i. Exporter.....	17
j. Grafana.....	18
k. CI/CD.....	18
l. Github action.....	19
m. Percona XtraDB Cluster.....	20
n. Percona ProxySQL.....	21
o. Percona XtraBackup.....	21
p. <i>Infrastructure as Code (IaC)</i> .....	22
q. Ansible.....	23
r. NGINX sebagai HTTP <i>load balancer</i> .....	23
r. Arsitektur monolitik.....	24
<b>BAB III METODE PENELITIAN.....</b>	<b>25</b>
3.1 Objek Penelitian.....	25
3.2 Alur Penelitian.....	27
a. Tentukan tujuan.....	28
b. Studi literatur .....	28
c. Perencanaan dan perancangan strategi untuk mencapai tujuan.....	28
d. Pengembangan ( <i>development</i> ).....	28
e. Pengujian awal.....	28
f. Implementasi.....	29
g. Pengujian sistem dan pengambilan data.....	29

h. Peninjauan dan analisa.....	30
3.3 Alat dan Bahan Penelitian.....	30
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>31</b>
4.1 Perencanaan.....	31
a. Server penyeimbang beban ( <i>load balancer</i> ).....	32
b. Server pekerja ( <i>workers</i> ) .....	32
c. Server pencadangan ( <i>backup</i> ).....	33
d. Server pemantauan ( <i>monitoring</i> ).....	34
e. Sistem CI/CD ( <i>Continuous Integration/Continuous Deployment</i> ).....	35
4.2 Pengembangan ( <i>development</i> ).....	36
a. Server penyeimbang beban ( <i>load balancer</i> ).....	37
b. Server pekerja ( <i>workers</i> ) .....	37
c. Server pencadangan ( <i>backup</i> ).....	39
d. Server pemantauan ( <i>monitoring</i> ).....	41
e. Sistem CI/CD ( <i>Continuous Integration/Continuous Deployment</i> ).....	43
4.3 Pengujian awal.....	47
a. Server pemantauan ( <i>monitoring</i> ).....	47
b. Server pencadangan ( <i>backup</i> ).....	49
c. Server penyeimbang beban ( <i>load balancer</i> ).....	51
d. Server pekerja ( <i>workers</i> ).....	52
e. Sistem CI/CD ( <i>Continuous Integration/Continuous Deployment</i> ).....	55
4.4 Implementasi pada server ( <i>deployment</i> ).....	56
4.5 Pengujian sistem dan pengambilan data.....	57
a. Skenario Pertama.....	57
b. Skenario Kedua .....	60
c. Hasil Akhir.....	63

4.5 Peninjauan dan analisa.....	64
<b>BAB V PENUTUP.....</b>	<b>68</b>
5.1 Kesimpulan.....	68
5.2 Saran.....	68
<b>DAFTAR PUSTAKA.....</b>	<b>69</b>
<b>LAMPIRAN.....</b>	<b>71</b>



## DAFTAR TABEL

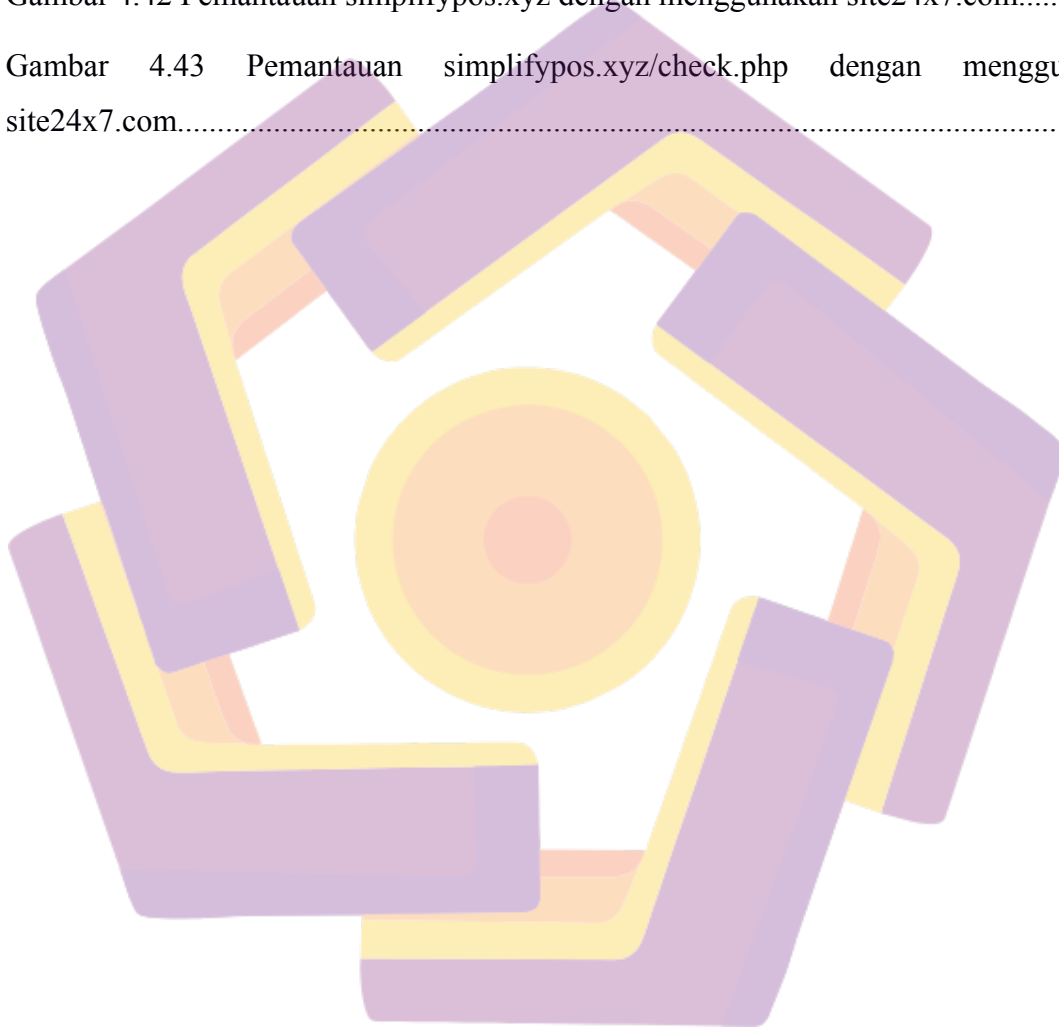
Tabel 2.1 Perbandingan Penelitian.....	8
Tabel 4.1 Langkah konfigurasi nginx.....	37
Tabel 4.2 Langkah konfigurasi Percona XtraDB Cluster.....	38
Tabel 4.3 Langkah konfigurasi proxysql.....	39
Tabel 4.4 Langkah konfigurasi backup.....	40
Tabel 4.5 Langkah konfigurasi prometheus.....	41
Tabel 4.6 Langkah konfigurasi node_exporter.....	41
Tabel 4.7 Langkah konfigurasi blackbox.....	42
Tabel 4.8 Langkah konfigurasi alertmanager.....	42
Tabel 4.9 Hasil percobaan dengan skenario 1.....	60
Tabel 4.10 Hasil percobaan dengan skenario 2.....	63
Tabel 4.11 Hasil akhir.....	63

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur docker.....	10
Gambar 2.2 Arsitektur docker.....	11
Gambar 2.3 Arsitektur prometheus.....	15
Gambar 2.4 Dashboard Grafana.....	16
Gambar 2.5 Github Action Workflow.....	18
Gambar 2.6 Percona XtraDB.....	18
Gambar 3.1 Halaman login simplifypos.....	25
Gambar 3.2 Halaman daftar <i>purchase order</i> .....	26
Gambar 3.3 Halaman detail <i>purchase order</i> .....	26
Gambar 3.4 Halaman daftar rekanan.....	26
Gambar 3.5 Halaman detail rekanan.....	27
Gambar 3.6 Halaman daftar produk.....	27
Gambar 3.7 Alur penelitian.....	28
Gambar 4.1 Desain sistem.....	31
Gambar 4.2 Desain server <i>worker</i> .....	33
Gambar 4.3 Pengiriman <i>backup</i> .....	34
Gambar 4.4 Desain pemantauan.....	35
Gambar 4.5 Gambaran sistem secara keseluruhan.....	36
Gambar 4.6 CI <i>workflow</i> .....	43
Gambar 4.7 <i>Job's steps</i> pada CI.....	44
Gambar 4.8 CD <i>workflow</i> .....	45
Gambar 4.9 <i>Job's steps</i> pada CD.....	46
Gambar 4.10 ansible-playbook.....	46
Gambar 4.11 <i>metrics</i> dalam bentuk text yang dihasilkan <i>node_exporter</i> .....	48
Gambar 4.12 Dashboard Grafana.....	48

Gambar 4.13 <i>blackbox exporter</i> .....	49
Gambar 4.14 Notifikasi telegram.....	50
Gambar 4.15 daftar <i>database logical backup</i> .....	50
Gambar 4.16 daftar <i>database physical backup</i> .....	51
Gambar 4.17 <i>database physical backup</i> .....	51
Gambar 4.18 <i>network traffic</i> dari <i>container</i> yang ada di <i>worker1</i> dan <i>worker2</i> .....	52
Gambar 4.19 Penggunaan CPU dari <i>container</i> yang ada di <i>worker1</i> dan <i>worker2</i> .....	52
Gambar 4.20 Penggunaan CPU dari <i>container</i> yang ada di <i>worker1</i> dan <i>worker2</i> .....	52
Gambar 4.21 Akses layanan langsung ke server <i>worker</i> .....	53
Gambar 4.22 Menambahkan <i>record</i> pada <i>database</i> pada server <i>worker1</i> .....	53
Gambar 4.23 Data pada <i>database</i> pada server <i>worker1</i> tereplikasi pada <i>database</i> di server <i>worker2</i> .....	54
Gambar 4.24 Menambahkan <i>record</i> pada <i>database</i> pada server <i>worker2</i> .....	54
Gambar 4.25 Data pada <i>database</i> pada server <i>worker2</i> tereplikasi pada <i>database</i> di server <i>worker1</i> .....	54
Gambar 4.26 <i>pull_request</i> dengan <i>tag</i> .....	55
Gambar 4.27 Memicu CI.....	55
Gambar 4.28 <i>pull_request</i> mengubah versi.....	56
Gambar 4.29 Memicu CD.....	56
Gambar 4.30 Skenario 1 - Percobaan pertama.....	58
Gambar 4.31 Skenario 1 - Percobaan kedua.....	58
Gambar 4.31 Skenario 1 - Percobaan ketiga.....	59
Gambar 4.33 Skenario 1 - Percobaan keempat.....	59
Gambar 4.34 Skenario 1 - Percobaan kelima.....	60
Gambar 4.35 Skenario 2 - Percobaan pertama.....	61
Gambar 4.36 Skenario 2 - Percobaan kedua.....	61

Gambar 4.37 Skenario 2 - Percobaan ketiga.....	62
Gambar 4.38 Skenario 2 - Percobaan keempat.....	62
Gambar 4.39 Skenario 2 - Percobaan kelima.....	63
Gambar 4.40 Grafik <i>error rate</i> antar percobaan antar skenario.....	64
Gambar 4.51 Grafik <i>response time</i> antar percobaan antar skenario.....	64
Gambar 4.42 Pemantauan simplifypos.xyz dengan menggunakan site24x7.com.....	65
Gambar 4.43 Pemantauan simplifypos.xyz/check.php dengan menggunakan site24x7.com.....	66



## DAFTAR LAMPIRAN

Lampiran 1. Docker compose untuk menjalankan <i>docker container</i> .....	71
Lampiran 2 <i>Script</i> untuk melakukan <i>backup database</i> secara <i>logical</i> .....	71
Lampiran 3 <i>Script</i> untuk melakukan <i>backup database</i> secara <i>physical</i> .....	72
Lampiran 4 Prometheus <i>config</i> dalam <i>template</i> jinja.....	72
Lampiran 5 Alertmanager <i>config</i> dalam <i>template</i> jinja.....	75
Lampiran 6 <i>Alert rules</i> secara <i>general</i> .....	76
Lampiran 7 Alur CD.....	77
Lampiran 8 Alur CI.....	78
Lampiran 9. Dockerfile dari simplifypos.....	79
Lampiran 10. Konfigurasi mysql dalam <i>template</i> jinja.....	79
Lampiran 11. Konfigurasi wsrep dalam <i>template</i> jinja.....	79
Lampiran 12. Ansible <i>task</i> yang digunakan untuk konfigurasi Percona XtraDB.....	80
Lampiran 13. Konfigurasi proxysql dalam <i>template</i> jinja.....	81
Lampiran 14. Ansible <i>task</i> untuk konfigurasi proxysql.....	82
Lampiran 15. Konfigurasi nginx dalam <i>template</i> jinja.....	83
Lampiran 16. <i>Service file</i> dalam <i>template</i> jinja.....	83
Lampiran 17. Konfigurasi blackbox.....	84



## INTISARI

Secara tradisional, aplikasi berbasis *web* dan semua layanan yang diperlukan seperti basis data, *web server*, *file server* dan lain sebagainya, yang memungkinkan aplikasi tersebut dijalankan, diletakkan pada satu server yang sama. Pada sistem tersebut tentu terdapat keunggulan dan kelemahan. Kelemahan yang paling terasa yaitu aplikasi tersebut akan sangat bergantung pada satu sumber daya dan kurangnya toleransi kesalahan yang apabila terjadi suatu insiden pada sumber daya tersebut misal server berhenti bekerja, maka otomatis aplikasi tersebut menjadi tidak dapat diakses. Hal ini tentu sangat berpengaruh bagi pengguna maupun penyedia layanan. Dari perspektif bisnis, ini juga tidak baik. Salah satu desain alternatif yang tujuannya untuk mengatasi permasalahan tersebut yaitu sistem terdistribusi.

Skripsi ini bertujuan untuk mengimplementasikan sistem terdistribusi yang menyediakan layanan *simplifypos*, sebuah aplikasi berbasis *website* yang digunakan untuk memudahkan manajemen permintaan mulai dari transaksi, dokumentasi hingga laporan. Layanan tersebut didukung oleh sistem yang didesain dengan memperhatikan prinsip ketersediaan, skalabilitas dan keandalan.

Ketersediaan merupakan kemampuan sistem menyediakan layanan agar dapat diakses dengan baik sesuai standar atau *level* yang ditetapkan pada suatu waktu. Skalabilitas merupakan kemampuan sistem untuk berkembang menyesuaikan dengan kebutuhan layanan. Keandalan merupakan kemampuan sistem dalam menghadapi dan mengatasi beban seperti tinggi nya lalu lintas akses ke layanan, penggunaan sumber daya dan lain sebagainya.

Dengan menerapkan ketiga prinsip tersebut diharapkan layanan yang disediakan dapat diakses dengan baik oleh pengguna dan memudahkan penyedia layanan untuk mengembangkan, menjaga performa layanan tetap baik dan andal.

Demi mewujudkan hal tersebut, desain sistem untuk layanan *simplifypos* ini harus mengantisipasi kemungkinan terjadinya insiden seperti gangguan jaringan, server dan lain sebagainya. Selain itu sistem juga harus memiliki kemampuan untuk berkembang menyesuaikan dengan beban sistem, sehingga beban tersebut dapat diatasi dan tidak mengganggu layanan agar mutu dan target dari SLO (*Service-Level Objective*) tetap dapat terpenuhi.

Dengan penerapan sistem terdistribusi, ini memungkinkan layanan dapat disediakan dari beberapa sumber daya dan mengurangi ketergantungan. Sebagai gambaran, pada implementasi ini, penulis menggunakan dua server sebagai *web server* dan juga sebagai *database server*, satu server sebagai penyeimbang beban, satu server sebagai server pencadangan dan satu server sebagai server pemantau. Untuk instalasi dan konfigurasi sistem pun juga sudah terdapat otomatisasi menggunakan *github action* dan *ansible*.

Hasil dari penelitian ini memperlihatkan ketersediaan layanan yang lebih baik, hal tersebut terlihat ketika terdapat salah satu server yang mengalami insiden namun layanan tetap dapat berjalan, selain itu, bagi penyedia layanan juga terdapat kemudahan dalam pengembangan layanan dan sistem.

**Kata Kunci:** Sistem terdistribusi, website, ketersediaan, skalabilitas, keandalan

## ABSTRACT

*Traditionally, a web-based application and all the services needed like database, web server, file server and so on that makes the application runs are placed on the same server. This system has its own advantages and disadvantages. Some of the biggest disadvantages are that the application will really depend on one resource and will be less fault-tolerant which when an incident happens like a server being down or not working properly then the application inside will be affected and can't be accessed. Such things will really affect both the users and the service provider. From the business perspective, this is not good. One of the alternative designs that used to solve those problems is a distributed system.*

*This thesis's objective is to implement the distributed system that serve simplifypos, a web-based application that is used to make the order management start from transaction, documentation up to reporting. This service is supported by the system that is designed with due regard to the principle of availability, scalability and reliability.*

*Availability is the system ability to serve the service that is available and accessible in a point of time according to the standard or level that was previously set. Scalability is the system ability to grow adapting with the service needs. Reliability is the system ability to handle and manage the load like high access traffic, resource usage and so on. By implementing those principles, we hope that the service will be well accessible by the users and makes the service provider easier to develop and keep the service performance good and reliable.*

*To make it happen, the system design for this simplifypos service must anticipate the possibility of incidents like network issues, server problems and so on. Beside that, the system should also have the ability to grow adapting with its load, so it can be managed and not interrupt or disturb the service in order to reach the target of SLO (Service-Level Objective).*

*By implementing this distributed system, this will make it possible for the service to be served by multiple instances or resources and less dependent on one resource. Just to illustrate, in this implementation, the author use two servers as the web server and also as the database server, one server as the load balancer, one server as the backup server and one server as monitoring server. For installation and configuration purposes, I am also using github action and ansible to automate them.*

*The result shows that the service availability is better. It looks when there is one server facing an incident or downtime but the service is still running, beside that for the service provider, this implementation makes the system developing is more easier.*

**Keywords:** *Distributed system, the website, availability, scalability, reliability.*