



UNIVERSITAS DIPONEGORO

**DESAIN DAN IMPLEMENTASI REDUNDANSI SISTEM *TRANSCODING*
TERSKALAKAN BERBASIS KUBERNETES**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Teknik**

Luqman Setyo Nugroho

21120117120008

FAKULTAS TEKNIK

DEPARTEMEN TEKNIK KOMPUTER

SEMARANG

September 2021

HALAMAN PENGESAHAN

Tugas Akhir diajukan oleh:

Nama : Luqman Setyo Nugroho
NIM : 21120117120008
Jurusan/Program Studi : Teknik Komputer
Judul Tugas Akhir : Desain dan Implementasi Redundansi Sistem
Transcoding Terskalakan Berbasis Kubernetes

Telah berhasil dipertahankan di hadapan Tim Penguji dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro.

TIM PENGUJI

Pembimbing I : Ike Pertiwi Windasari, S.T., M.T. ()
Pembimbing II : Dr. Adian Fatchur Rochim, S.T., M.T. ()
Ketua Penguji : Kumiawan Teguh Martono, S.T., M.T. ()
Anggota Penguji : Risma Septiana, S.T, M.Eng. ()

Semarang, 24 September 2021

Ketua Departemen Teknik Komputer




Dr. R. Rizal Isnanto, S.T., M.M., M.T.

NIP. 197007272000121001

HALAMAN PERNYATAAN ORISINILITAS

**Tugas Akhir ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk telah saya
nyatakan dengan benar.**

Nama : Luqman Setyo Nugorho
NIM : 21120117120008
Tanda Tangan : 
Tanggal : Semarang, 23 Agustus 2021

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Diponegoro, saya yang bertanda tangan di bawah ini :

Nama : LUQMAN SETYO NUGROHO

NIM : 21120117120008

Departemen : TEKNIK KOMPUTER

Fakultas : TEKNIK

Jenis Karya : TUGAS AKHIR

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya berjudul :

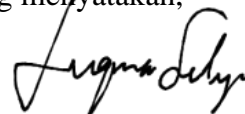
Desain dan Implementasi Redundansi Sistem *Transcoding* Terskalakan Berbasis Kubernetes beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Diponegoro berhak menyimpan, mengalihmedia / formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan memublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di : Semarang

Pada tanggal : 23 Agustus 2021

Yang menyatakan,



(Luqman Setyo Nugroho)

ABSTRAK

Dari penelitian yang dilakukan pada tahun 2017, didapatkan fakta bahwa 70% lalu lintas data yang ada di internet merupakan data multimedia. Terdapat tantangan untuk dapat mengakomodir perbedaan akses pada pengguna, seperti spesifikasi perangkat atau kondisi jaringan. Salah satu solusi dari hal tersebut adalah menggunakan metode Adaptive Bitrate Streaming (ABR). Proses ABR sendiri membutuhkan masukan berupa suatu video dengan berbagai jenis bitrate dan resolusi. Proses untuk mengkonversi suatu video ke dalam video dengan encoding, resolusi, atau bitrate lain dinamakan proses transcoding. Salah satu pendekatan yang dapat dilakukan untuk melakukan proses tersebut adalah menggunakan sistem transcoding terdistribusi. Pada tahun 2016, dilakukan penelitian untuk membuat sebuah sistem transcoding terdistribusi berbasis awan yang dapat terskalakan. Namun, sistem tersebut memiliki limitasi yaitu single point of failure (SPOF). Tujuan pada penelitian ini adalah untuk memperbaiki penelitian sebelumnya, namun dengan menghindari desain sistem dengan single point of failure.

Penelitian ini menggunakan platform Kubernetes untuk menyediakan redundansi pada sistem yang dibangun. Sistem redundansi ini berguna untuk menghilangkan single point of failure pada sistem. Selain itu, sistem yang dibangun juga dapat terskalakan dengan memanfaatkan kemampuan horizontal scaling pada Kubernetes. Kemudian, sistem yang dibangun juga menggunakan message queue & media penyimpanan yang terpisah untuk menghindari sistem yang “tightly coupled”.

Dari hasil penelitian ini, didapatkan bahwa sistem yang dibangun dapat menghindari single point of failure. Hasil pengujian yang dilakukan untuk menguji aspek SPOF menunjukkan success rate hingga 99,998% pada skenario Chaos Mesh yang dilakukann. Skenario Chaos Mesh digunakan untuk mensimulasikan hal yang terjadi ketika terdapat node yang mati pada sistem. Namun, sistem lebih lambat daripada sistem pada penelitian sebelumnya sekitar 39%-175% bergantung pada konfigurasi sistem. Hal tersebut terjadi karena konfigurasi bawaan Kubernetes tidak dioptimasi untuk jenis pekerjaan yang berati di CPU (CPU Bound).

Kata Kunci— Transcoding, Kubernetes, Single Point of Failure.

ABSTRACT

From research conducted in 2017, it was found that 70% of data traffic on the internet is multimedia data. There are challenges to be able to accommodate differences in access to users, such as device specifications or network conditions. One solution to this is to use the Adaptive Bitrate Streaming (ABR) method. The ABR process itself requires input in the form of a video with various types of bitrates and resolutions. The process of converting a video into a video with an encoding, resolution, or other bitrate is called the transcoding process. One approach that can be taken to carry out this process is to use a distributed transcoding system. In 2016, research was conducted to create a scalable, cloud-based distributed transcoding system. However, the system has a limitation, namely a single point of failure (SPOF). The purpose of this study is to improve on previous research, but by avoiding system designs with a single point of failure.

This study uses the Kubernetes platform to provide redundancy in the system being built. This redundancy system is useful for eliminating single point of failure in the system. In addition, the system built can also be scalable by utilizing the horizontal scaling capability of Kubernetes. Then, the system built also uses a separate message queue & storage media to avoid a "tightly coupled" system.

From the results of this study, it was found that the system built can avoid a single point of failure. The results of the tests carried out to test the SPOF aspect showed a success rate of up to 99.998% in the Chaos Mesh scenario that was carried out. The Chaos Mesh scenario is used to simulate what happens when there are dead nodes in the system. However, the system is slower than the system in the previous study by about 39%-175% depending on the system configuration. This happens because the default Kubernetes configuration is not optimized for the type of work that is heavy on the CPU (CPU Bound).

Keywords- Transcoding, Kubernetes, Single Point of Failure,

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT karena atas rahmat dan karunia-Nya, Penulis dapat menyelesaikan penyusunan tugas akhir dengan judul “**Desain dan Implementasi Redundansi Sistem *Transcoding* Terskalakan Berbasis Kubernetes**”. Dalam penyusunan Tugas Akhir ini Penulis banyak menerima dukungan, bimbingan, bantuan, doa serta arahan dari berbagai pihak. Oleh karena itu, melalui kesempatan ini Penulis bermaksud menyampaikan ucapan terima kasih kepada:

1. Dr. R. Rizal Isnanto, ST. MM. MT selaku Ketua Departemen Teknik Komputer Universitas Diponegoro yang telah memberikan lingkungan akademis yang baik sehingga Tugas Akhir ini dapat terlaksana.
2. Ike Pertiwi Windasari, S.T., M.T. dan Dr. Adian Fatchur Rochim., ST., MT selaku dosen pembimbing Tugas Akhir yang telah memberi masukan serta solusi permasalahan ketika pengerjaan Tugas Akhir berlangsung.
3. Seluruh dosen, mahasiswa, staf, dan pegawai Teknik Komputer Undip yang telah membantu kegiatan akademis dan administrasi sehingga Tugas Akhir dapat dikerjakan dengan lancar.
4. Kedua orang tua, saudara, serta keluarga besar tercinta atas doanya yang tidak pernah berhenti kepada Penulis.
5. Seluruh pihak yang tidak dapat Penulis sebutkan satu-persatu yang telah membantu hingga terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa segala kemampuan dan ilmu pengetahuan masih terbatas sehingga Tugas Akhir ini membutuhkan masukan, saran serta kritik yang membangun dari berbagai kalangan. Penulis berharap, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang memerlukan. Akhir kata Penulis ucapkan terima kasih.

Semarang, 23 Agustus 2021

Penulis

DAFTAR ISI

DESAIN DAN IMPLEMENTASI REDUNDANSI SISTEM <i>TRANSCODING</i> TERSKALAKAN BERBASIS KUBERNETES	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN ORISINILITAS	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	iv
ABSTRAK	v
ABSTRACT.....	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xiv
1. BAB I PENDAHULUAN	15
1.1. Latar Belakang.....	15
1.2. Rumusan Masalah.....	17
1.3. Batasan Masalah	17
1.4. Tujuan Penelitian	18
1.5. Manfaat Penelitian	18
1.6. Metodologi Penelitian.....	18
1.7. Sistematika Penulisan	19
2. BAB II LANDASAN TEORI	21
2.1. Penelitian Terdahulu	21
2.2. Dasar Teori	23
2.2.1. Skalabilitas.....	23
2.2.2. <i>Single Point of Failure</i>	24
2.2.3. Kubernetes	25
2.2.4. <i>Transcoding</i>	28
2.2.5. Virtualisasi Komputer.....	29

2.2.6.	RabbitMQ	29
2.2.7.	MongoDB	30
2.2.8.	Golang.....	30
2.2.9.	REST API	31
2.2.10.	<i>Chaos Mesh</i>	32
2.2.11.	<i>Golden Metrics</i>	32
3.	BAB III METODE PENELITIAN	33
3.1.	Identifikasi Kebutuhan.....	34
3.1.1.	Kebutuhan Fungsional	34
3.1.2.	Kebutuhan Non-Fungsional	35
3.1.3.	Kebutuhan Pengembangan Sistem.....	35
3.2.	Perancangan Sistem	37
3.2.1.	Perancangan Arsitektur Sistem.....	37
3.2.2.	Perancangan Struktur Data pada Basis Data.....	39
3.2.3.	Perancangan Diagram UML	44
4.	BAB IV PEMBAHASAN.....	56
4.1.	Implementasi Sistem.....	56
4.2.	Pengujian Implementasi.....	57
4.2.1.	Pengujian Performa Sistem Implementasi	57
4.2.2.	Pengujian Performa Sistem Morph.....	58
4.2.3.	Pengujian Sistem Implementasi Menggunakan Metode <i>Load Testing</i> 58	
4.3.	Pembahasan Hasil Pengujian	59
4.3.1.	Pembahasan Perbandingan Performa Implementasi dengan Morph 59	
4.3.2.	Pembahasan Hasil Pengujian Beban (<i>Load Testing</i>) Sistem ...	66
4.4.	Pengujian <i>Blackbox</i> Terhadap Sistem Implementasi.....	73
4.4.1.	Pengujian Fungsionalitas Mengunggah Video	73
4.4.2.	Pengujian Fungsionalitas Halaman <i>Progress</i>	74
4.4.3.	Pengujian Fungsionalitas Halaman Worker	75

4.4.4. Pengujian Fungsionalitas Halaman Pemutar Video.....	77
5. BAB V KESIMPULAN DAN SARAN.....	79
5.1. Kesimpulan	79
5.2. Saran	80
DAFTAR PUSTAKA	81

DAFTAR TABEL

Tabel 3.1 Kebutuhan perangkat keras sistem.....	36
Tabel 3.2 Daftar atribut dokumen video.	40
Tabel 3.3 Daftar atribut dokumen <i>worker</i>	40
Tabel 3.4 Daftar atribut dokumen <i>task</i>	41
Tabel 3.5. Daftar Atribut <i>embedded document</i> SplitTask.....	42
Tabel 3.6. Daftar Atribut <i>embedded document</i> TranscodeTask.....	43
Tabel 3.7. Daftar Atribut <i>embedded document</i> DashTask.	43
Tabel 3.8. Deskripsi diagram <i>use-case</i>	46
Tabel 4.1. Perbandingan durasi sistem implementasi, Morph, dan VM.....	60
Tabel 4.2. Perbandingan durasi <i>transcoding</i>	60
Tabel 4.3. Hasil persentase kenaikan performa kecepatan <i>transcoding</i>	61
Tabel 4.4. Durasi proses <i>transcoding</i> setiap potongan video.	65
Tabel 4.5. Komparasi persentase kesuksesan pada kondisi normal dan skenario <i>chaos</i>	67
Tabel 4.6. Komparasi jumlah permintaan per detik pada kondisi normal dan skenario <i>chaos</i>	68
Tabel 4.7. Hasil pengujian latensi permintaan (durasi dalam milisekon).....	68
Tabel 4.8. Tabel pengujian fungsionalitas mengunggah video.....	74
Tabel 4.9. Tabel pengujian fungsionalitas halaman <i>progress task</i>	75
Tabel 4.10. Tabel pengujian fungsionalitas halaman <i>worker</i>	76
Tabel 4.11. Tabel pengujian fungsionalitas halaman video <i>player</i>	78

DAFTAR GAMBAR

Gambar 3.1 Fase metode <i>rapid application development</i>	33
Gambar 3.2 Arsitektur implementasi sistem.....	37
Gambar 3.3 Diagram kelas untuk basis data yang dirancang.	44
Gambar 3.4 Diagram <i>use-case</i> aplikasi.....	45
Gambar 3.5 Diagram <i>sequence</i> mengunggah video.....	47
Gambar 3.6 Diagram <i>sequence</i> mengambil <i>progress task</i>	48
Gambar 3.7 Diagram <i>sequence</i> mengambil <i>status worker</i>	49
Gambar 3.8 Diagram <i>sequence</i> mengatur jumlah <i>worker</i>	49
Gambar 3.9 Diagram <i>sequence streaming</i> video.	50
Gambar 3.10 Diagram aktivitas mengunggah video.....	51
Gambar 3.11 Diagram aktivitas mengambil <i>progress task</i>	52
Gambar 3.12 Diagram aktivitas mengambil <i>status worker</i>	53
Gambar 3.13 Diagram aktivitas mengatur jumlah <i>worker</i>	53
Gambar 3.14 Diagram aktivitas melakukan <i>streaming</i>	54
Gambar 3.15 <i>Deployment</i> diagram sistem.	55
Gambar 4.1 Komparasi durasi proses <i>transcoding</i> implementasi (An-SPf), Morph, dan VM	60
Gambar 4.2. Komparasi durasi proses <i>transcoding</i> antara implementasi (An-SPf) dan Morph.....	61
Gambar 4.3. Komparasi peningkatan kecepatan <i>transcoding</i> antara implementasi (An-SPf) dan Morph.	62
Gambar 4.4. Penggunaan CPU dan memori pada implementasi (An-SPf), Morph, dan VM.	63
Gambar 4.5. Penggunaan CPU dan memori pada sistem implementasi (An-SPf), Morph.....	64
Gambar 4.6. Data deret waktu penggunaan CPU pada <i>worker</i> sistem implementasi.	66
Gambar 4.7. Tingkat kesuksesan pada sistem implementasi dengan beberapa konfigurasi jumlah replika berbeda dalam kondisi pengujian <i>chaos</i>	67
Gambar 4.8. Grafik distribusi persentil untuk skenario tidak terdapat gangguan pada beberapa konfigurasi jumlah replika.	69

Gambar 4.9. Grafik persentil distribusi persentil untuk skenario <i>chaos</i> pada beberapa konfigurasi jumlah replika.	70
Gambar 4.10. Komparasi distribusi persentil permintaan untuk skenario <i>chaos</i> dan kondisi normal pada konfigurasi 1 replika.	71
Gambar 4.11. Komparasi distribusi persentil permintaan untuk skenario <i>chaos</i> dan kondisi normal pada konfigurasi 3 replika.	71
Gambar 4.12. Komparasi distribusi persentil permintaan untuk skenario <i>chaos</i> dan kondisi normal pada konfigurasi 5 replika.	72
Gambar 4.13. Komparasi distribusi persentil permintaan untuk skenario <i>chaos</i> dan kondisi normal pada konfigurasi 7 replika.	73
Gambar 4.14. Tampilan halaman web untuk mengunggah video.	74
Gambar 4.15. Tampilan halaman web ketika proses mengunggah video.	74
Gambar 4.16. Tampilan halaman web <i>progress task</i>	75
Gambar 4.17. Tampilan halaman web <i>worker</i>	77
Gambar 4.18. Tampilan halaman web <i>worker</i> ketika jumlah <i>worker</i> dinaikkan. .	77
Gambar 4.19. Tampilan halaman web <i>worker</i> ketika jumlah <i>worker</i> diturunkan. .	77
Gambar 4.20. Tampilan halaman web <i>player</i> ketika video diputar.	78

DAFTAR LAMPIRAN