

**PENERAPAN ARSITEKTUR KOREOGRAFI PADA
LAYANAN BERBASIS RESTFUL UNTUK APLIKASI
KONVERSI DOKUMEN**

Skripsi



oleh:

IMMANUEL JOY PERKASA

71200544

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA**

2024

**PENERAPAN ARSITEKTUR KOREOGRAFI PADA
LAYANAN BERBASIS RESTFUL UNTUK APLIKASI
KONVERSI DOKUMEN**

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

IMMANUEL JOY PERKASA

71200544

DUTA WACANA

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

2024

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul:

PENERAPAN ARSITEKTUR KOREOGRAFI PADA LAYANAN BERBASIS RESTFUL UNTUK APLIKASI KONVERSI DOKUMEN

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari tugas akhir keserjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil tugas akhir ini adalah hasil plagiasi atau tiruan dari tugas akhir lain, saya bersedia dikenai sanksi yakni pencabutan gelar keserjanaan saya.

Yogyakarta, 2 Oktober 2024



IMMANUEL JOY PERKASA

71200544

HALAMAN PERSETUJUAN

Judul Tugas Akhir : PENERAPAN ARSITEKTUR KOREOGRAFI
PADA LAYANAN BERBASIS RESTFUL
UNTUK APLIKASI KONVERSI DOKUMEN

Nama Mahasiswa : IMMANUEL JOY PERKASA

NIM : 71200544

Mata Kuliah : Tugas Akhir

Kode : TI0366

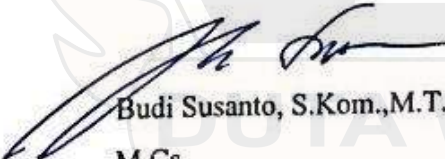
Semester : Ganjil

Tahun Akademik : 2024/2025

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 1 Oktober 2024

Dosen Pembimbing I

Dosen Pembimbing II


Budi Susanto, S.Kom., M.T.,
M.Cs


Willy Sudiarto Raharjo, S.Kom.,

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI/TESIS/DISERTASI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Kristen Duta Wacana, saya yang bertanda tangan di bawah ini:

Nama : Immanuel Joy Perkasa
NIM : 71200544
Program studi : Informatika
Fakultas : Teknologi Informasi
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Kristen Duta Wacana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

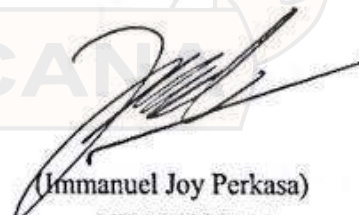
“PENERAPAN ARSITEKTUR KOREOGRAFI PADA LAYANAN BERBASIS RESTFUL UNTUK APLIKASI KONVERSI DOKUMEN”

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Kristen Duta Wacana berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : 6 November 2024

Yang menyatakan


(Immanuel Joy Perkasa)
NIM.71200544

HALAMAN PENGESAHAN

PENERAPAN ARSITEKTUR KOREOGRAFI PADA LAYANAN BERBASIS RESTFUL UNTUK APLIKASI KONVERSI DOKUMEN

Oleh: IMMANUEL JOY PERKASA / 71200544

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 23 Oktober 2024

Yogyakarta, 1 November 2024
Mengesahkan,

Dewan Penguji:

1. Budi Susanto, SKom.,M.T.
2. Willy Sudiarto Raharjo, S.Kom.,M.Cs.
3. Rosa Delima, Dr. S.Kom., M.Kom.
4. Gani Indriyanta, Ir. M.T.

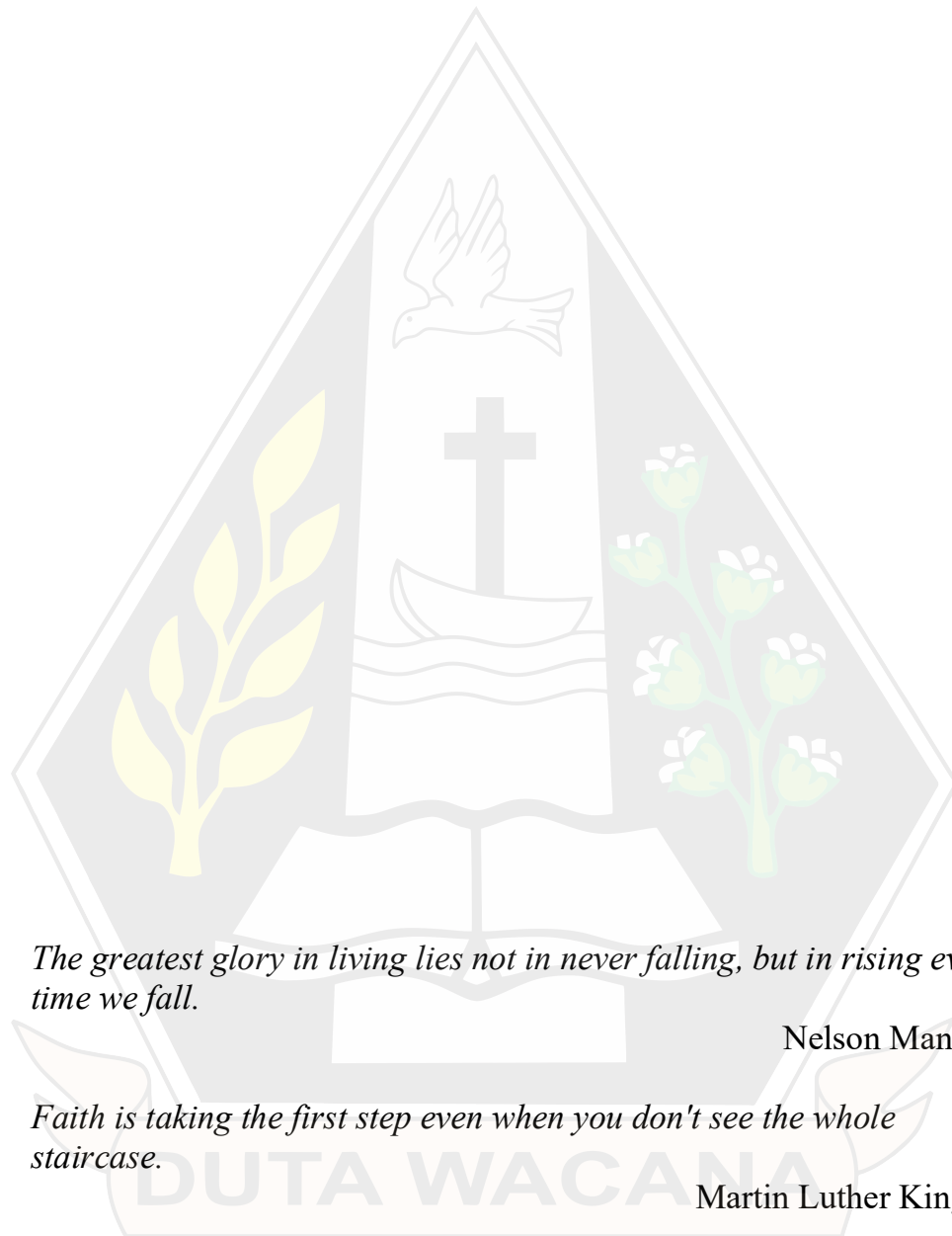


Dekan

(Restyandnu, S.Kom.,MSIS.,Ph.D.)

Ketua Program Studi

(Joko Purwadi, S.Kom., M.Kom.)



The greatest glory in living lies not in never falling, but in rising every time we fall.

Nelson Mandela

Faith is taking the first step even when you don't see the whole staircase.

Martin Luther King Jr.

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa, karena atas berkat dan kasih karunia-Nya, penulis dapat menyelesaikan Skripsi dengan judul **PENERAPAN ARSITEKTUR KOREOGRAFI PADA LAYANAN BERBASIS RESTFUL UNTUK APLIKASI KONVERSI DOKUMEN** ini dengan baik.

Dalam proses penyusunan Skripsi ini, penulis menerima banyak bantuan, dukungan, dan kerja sama, baik secara moral maupun spiritual. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, atas berkat dan kasih-Nya yang selalu menyertai penulis.
2. Orang tua yang senantiasa memberikan doa, kasih sayang, dan dukungan tanpa henti.
3. Bapak Restyandito, S.Kom., MSIS., Ph.D., selaku Dekan Fakultas Teknologi Informasi.
4. Bapak Joko Purwadi, S.Kom, M.Kom. selaku Kaprodi Informatika.
5. Bapak Budi Susanto, S.Kom., M.T., selaku Dosen Pembimbing 1, yang dengan sabar meluangkan waktu untuk memberikan arahan dan masukan dalam penyusunan Skripsi ini.
6. Bapak Willy Sudiarto Raharjo, S.Kom., M.Cs., selaku Dosen Pembimbing 2, yang telah memberikan kritik dan saran yang sangat berharga untuk penyempurnaan Skripsi ini.
7. Kakak-kakak tercinta yang selalu memberikan dukungan penuh secara moral dan spiritual, serta keponakan tersayang, Kanya, yang menjadi sumber semangat dalam menyelesaikan studi ini.
8. Grup teman-teman IDNC yang selalu memberikan semangat dan dukungan yang tak ternilai selama masa penelitian.
9. Grup MATIUS 6:34 yang telah membantu dan mendukung penulis selama masa perkuliahan, baik secara moral maupun spiritual.

10. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu penulis dalam berbagai aspek selama proses penyusunan Skripsi ini.

Penulis menyadari bahwa Skripsi ini masih jauh dari sempurna, oleh karena itu segala kritik dan saran yang membangun sangat diharapkan demi kesempurnaan karya ini. Semoga Skripsi ini dapat memberikan manfaat bagi pembaca serta berkontribusi dalam perkembangan ilmu komputer dan teknologi informasi.

Yogyakarta, 2 Oktober 2024



Immanuel Joy Perkasa



DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA	vi
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR LISTING	xvi
INTISARI	xviii
ABSTRACT.....	xx
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Landasan Teori	13
2.2.1 Arsitektur Asynhronous	13
2.2.2 Microservices Koreografi	13
2.2.3 Kafka.....	17
2.2.4 Automated Testing.....	19
BAB III METODOLOGI PENELITIAN	22

3.1	Analisis Kebutuhan Sistem	22
3.1.1	Perangkat Keras	22
3.1.2	Perangkat Lunak	22
3.2	Objek Penelitian	23
3.3	Subjek Penelitian.....	23
3.4	Arsitektur Sistem.....	23
3.4.1	Konfigurasi 1: Langsung dari Aplikasi Web ke Layanan Konversi	24
3.4.2	Konfigurasi 2: Penggunaan Kafka sebagai Message Broker dengan Layanan Konversi dan Notificator Bersama	25
3.4.3	Konfigurasi 3: Penggunaan Kafka dengan Layanan Konversi Terpisah dan Notificator Terpisah dan Menggunakan dua Broker	27
3.5	Perancangan Antarmuka	30
3.6	Metode Penelitian.....	32
BAB IV IMPLEMENTASI DAN PEMBAHASAN		37
4.1	Implementasi Sistem	37
4.1.1	Implementasi Aplikasi Web.....	37
4.1.2	Implementasi Konfigurasi Sistem.....	38
4.1.3	Implementasi Konfigurasi Server	67
4.1.4	Implementasi Konfigurasi Kafka.....	68
4.1.5	Implementasi Frontend	71
4.1.6	Implementasi Pengujian.....	72
4.2	Hasil Rancangan Antarmuka Pengguna	75
4.3	Analisis.....	78
4.3.1	Fungsional dan Logging	78
4.3.2	Perbandingan Hasil DOCX dan PDF	83
4.4	Pembahasan.....	85
4.4.1	Pengujian dengan JMeter	85
4.4.2	Hasil Performa	99

4.4.3 Perbandingan Kinerja Sistem.....	106
BAB V KESIMPULAN DAN SARAN	117
5.1 Kesimpulan.....	117
5.2 Saran.....	118
DAFTAR PUSTAKA	119
LAMPIRAN A KODE SUMBER PROGRAM	121
LAMPIRAN B KARTU KONSULTASI DOSEN 1	143
LAMPIRAN C KARTU KONSULTASI DOSEN 2	144
LAMPIRAN D CONTOH DATA HASIL PENGUJIAN	145
LAMPIRAN E FORMULIR REVISI SKRIPSI.....	149



DAFTAR TABEL

Tabel 2.1. Perbandingan Monolith dan Microservices	6
Tabel 2.2. Rangkuman Tinjauan Pustaka.....	11
Tabel 3.1. Contoh tabel hasil evaluasi dengan metrik evaluasi yang sudah ditentukan.....	35
Tabel 4.1. Komponen Konfigurasi Sistem.....	38
Tabel 4.2. Tabel konfigurasi JMeter pada setiap konfigurasi	74
Tabel 4.3. Hasil Pengujian JMeter Konfigurasi 1 pada layanan docxpdf.....	87
Tabel 4.4. Hasil Pengujian JMeter Konfigurasi 1 pada layanan pptxpdf	88
Tabel 4.5. Hasil Pengujian JMeter Konfigurasi 1 pada layanan xlsxpdf.....	88
Tabel 4.6. Hasil Pengujian JMeter Konfigurasi 1 pada layanan splitpdf.....	89
Tabel 4.7. Hasil Pengujian JMeter Konfigurasi 1 pada layanan protectpdf	89
Tabel 4.8. Hasil Pengujian JMeter Konfigurasi 2 untuk layanan docxpdf	90
Tabel 4.9. Hasil Pengujian JMeter Konfigurasi 2 untuk layanan pptxpdf.....	91
Tabel 4.10. Hasil Pengujian JMeter Konfigurasi 2 untuk layanan xlsxpdf	92
Tabel 4.11. Hasil Pengujian JMeter Konfigurasi 2 untuk layanan splitpdf.....	93
Tabel 4.12. Hasil Pengujian JMeter Konfigurasi 2 untuk layanan protectpdf.....	94
Tabel 4.13. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan docxpdf (skalabilitas 2 broker)	96
Tabel 4.14. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan docxpdf (skalabilitas 3 broker)	96
Tabel 4.15. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan docxpdf (skalabilitas 4 broker)	97
Tabel 4.16. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan pptxpdf.....	97
Tabel 4.17. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan xlsxpdf	98
Tabel 4.18. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan splitpdf.....	98
Tabel 4.19. Hasil Pengujian JMeter Konfigurasi 3 untuk layanan protectpdf.....	99

DAFTAR GAMBAR

Gambar 2.1. Performa Analisis Message Broker (Fu, Zhang, & Yu, 2020).....	8
Gambar 2.2 Contoh Arsitektur Koreografi	14
Gambar 2.3. Arsitektur KafkaAPI Producers dan Consumers	17
Gambar 3.1. Arsitektur sistem konfigurasi 1	24
Gambar 3.2. Arsitektur sistem konfigurasi 2	26
Gambar 3.3. Arsitektur sistem konfigurasi 3	28
Gambar 3.4. Wireframe Halaman Home	30
Gambar 3.5. Wireframe halaman service.....	30
Gambar 3.6. Wireframe halaman selesai submit file	31
Gambar 3.7. Diagram Alur Penelitian	32
Gambar 4.1. Isi Instances docxpdf pada konfigurasi 3	61
Gambar 4.2. Halaman Login.....	75
Gambar 4.3. Halaman Home.....	76
Gambar 4.4. Halaman Service	76
Gambar 4.5. Ketika permintaan sedang diproses.....	77
Gambar 4.6. Ketika permintaan sudah diproses	77
Gambar 4.7. Logging ketika Request berhasil diterima oleh backend (Konfigurasi 2)	79
Gambar 4.8. Hasil konversi berhasil dikirim (Konfigurasi 1)	79
Gambar 4.9. Kafka Broker ID 0 sudah dapat aktif	80
Gambar 4.10. Producer berhasil mengirim pesan ke Kafka	80
Gambar 4.11. Hasil konversi berhasil dikirim (Konfigurasi 2)	81
Gambar 4.12. Kafka Broker ID 3 diaktifkan untuk Konfigurasi 3	81
Gambar 4.13. Dua broker aktif di zookeeper yang sama	82
Gambar 4.14. Producer berhasil mengirim pesan (Konfigurasi 3)	82
Gambar 4.15. Consumer berhasil menerima pesan dari Kafka (Konfigurasi 3)....	82
Gambar 4.16. Notificator Producer dapat menerima permintaan	83
Gambar 4.17. Notificator consumer dapat menerima pesan dari Kafka	83
Gambar 4.18. Screenshot S3 ketika proses konversi berhasil.....	84
Gambar 4.19. Input file (.docx)	85

Gambar 4.20. Output file (.pdf)	85
Gambar 4.21. Line Chart Success Rate Konfigurasi 1	100
Gambar 4.22. Line Chart Latency Konfigurasi 2	101
Gambar 4.23. Line Chart Success Rate Konfigurasi 3	102
Gambar 4.24. Line Chart Latency Konfigurasi 3	103
Gambar 4.25. Line Chart Success Rate Konfigurasi 3	104
Gambar 4.26. Line Chart Latency Konfigurasi 3	105
Gambar 4.27. Line Chart Perbandingan Success Rate pada Setiap Konfigurasi (1 MB).....	106
Gambar 4.28. Line Chart Perbandingan Latency pada Setiap Konfigurasi (1 MB).....	108
Gambar 4.29. Line Chart Perbandingan Success Rate pada Setiap Konfigurasi (5 MB).....	110
Gambar 4.30. Line Chart Perbandingan Latency pada Setiap Konfigurasi (5 MB).....	111
Gambar 4.31. Line Chart Perbandingan Success Rate pada Setiap Konfigurasi (10 MB).....	113
Gambar 4.32. Line Chart Perbandingan Latency pada Setiap Konfigurasi (10 MB).....	115



DAFTAR LISTING

Listing 4.1. Kode penyimpanan dan pengiriman data layanan docx	39
Listing 4.2. Kode tambahan data action dan password pada protectpdf.....	40
Listing 4.3. Kode upload S3	40
Listing 4.4. untuk mengirimkan data ke endpoint API yang disediakan oleh backend	41
Listing 4.5. Kode menyimpan data dari request ke dalam variable.....	42
Listing 4.6. fungsi convert_docx_to_pdf()	43
Listing 4.7. Kode fungsi fungsi upload_file_to_s3(),	43
Listing 4.8. Kode Fungsi send_email().....	45
Listing 4.9. Kode Fungsi docxpdf()	45
Listing 4.10. Kode data yang dikirimkan frontend di Konfigurasi 2	46
Listing 4.11. kode terkait pengiriman permintaan pada layanan docxpdf.....	46
Listing 4.12. kode terkait pengiriman permintaan pada layanan protectpdf.....	47
Listing 4.13. Kode pada backend untuk menerima permintaan.....	48
Listing 4.14. Kode klien S3 mengambil URL dari data yang dikirimkan	49
Listing 4.15. Kafka Producer	50
Listing 4.16. producer.py menangani proses permintaan	52
Listing 4.17. Inisiasi Kafka dan topic	53
Listing 4.18. Proses consumer dalam menerima dan menguraikan pesan.....	54
Listing 4.19. Mengunduh file dari S3	55
Listing 4.20. Pemrosesan file yang berhasil diunduh	55
Listing 4.21. Kode fungsi run_script().....	56
Listing 4.22. Konfigurasi SMTP dan AWS	57
Listing 4.23. Proses Konversi oleh LibreOffice	58
Listing 4.24. Upload ke S3	59
Listing 4.25. Kode fungsi send_email().....	60
Listing 4.26. Inisiasi Kafka dan Topic	61
Listing 4.27. Proses consumer dalam menerima dan menguraikan pesan.....	61
Listing 4.28. Fungsi pengiriman pesan ke notifikator dari docxpdf.py	63
Listing 4.29. Notifikator Producer	64

Listing 4.30. Kode fungsi <code>download_file_from_s3()</code>	65
Listing 4.31. Kode Fungsi <code>send_email()</code>	66
Listing 4.32. Kode fungsi <code>process_message()</code>	67
Listing 4.33. Konfigurasi Zookeeper	69
Listing 4.34. Konfigurasi Broker pada Konfigurasi 2	70
Listing 4.35. Penambahan Broker pada Konfigurasi 3	71



INTISARI

PENERAPAN ARSITEKTUR KOREOGRAFI PADA LAYANAN BERBASIS RESTFUL UNTUK APLIKASI KONVERSI DOKUMEN

Oleh

IMMANUEL JOY PERKASA

71200544

Pada era digital saat ini, penggunaan dokumen digital telah menjadi kebutuhan penting di berbagai sektor, terutama di lingkungan kerja dan pendidikan. Permintaan untuk mengonversi dokumen dari satu format ke format lain terus meningkat seiring dengan kebutuhan untuk berbagi, mencetak, atau mengintegrasikan data dari berbagai sumber. Penelitian ini bertujuan untuk mengembangkan aplikasi web konversi dokumen berbasis arsitektur microservices dan pemrosesan asinkron, dengan menggunakan Apache Kafka sebagai message broker untuk mengatasi masalah antrian panjang dan lonjakan permintaan yang tinggi.

Sistem yang dirancang berfokus pada pemrosesan dokumen dalam jumlah besar secara cepat dan efisien. Aplikasi ini menggunakan arsitektur microservices dengan koreografi yang mengelola beberapa layanan seperti konversi dokumen, pengiriman notifikasi, dan manajemen antrian permintaan. Konfigurasi yang diterapkan mencakup pembagian beban melalui penambahan broker dan partisi, yang diuji dalam berbagai skenario untuk menilai kinerja Kafka dalam mengelola permintaan.

Hasil penelitian menunjukkan bahwa penerapan arsitektur asinkron dan penggunaan Kafka secara signifikan meningkatkan performa aplikasi dalam

menangani lonjakan permintaan konversi dokumen. Dengan throughput yang lebih tinggi dan latensi yang lebih rendah dibandingkan dengan sistem yang tidak menggunakan Kafka, sistem pada konfigurasi dengan 2 broker dan partisi mampu memproses dokumen berukuran 1 MB hingga 10 MB dengan lebih cepat dan efisien. Penelitian ini menunjukkan bahwa arsitektur asinkron berbasis microservices dan message broker merupakan solusi optimal dalam mengatasi tantangan pemrosesan dokumen dalam skala besar dan kecepatan tinggi.

Kata kunci: konversi dokumen, arsitektur asynchronous, microservices, Apache Kafka, message broker, koreografi.



ABSTRACT

APPLICATION OF CHOREOGRAPHY ARCHITECTURE IN RESTFUL SERVICES FOR DOCUMENT CONVERSION APPLICATIONS

By

IMMANUEL JOY PERKASA

71200544

In the current digital era, the use of digital documents has become a crucial need across various sectors, especially in work and education environments. The demand for converting documents from one format to another continues to rise, driven by the need to share, print, or integrate data from multiple sources. This study aims to develop a web-based document conversion application utilizing microservices architecture and asynchronous processing, with Apache Kafka as the message broker to address the challenges of long queues and high request spikes.

The designed system focuses on processing large volumes of documents quickly and efficiently. The application employs a microservices architecture with choreography that manages various services, such as document conversion, notification delivery, and request queue management. The configuration applied includes load balancing through the addition of brokers and partitions, tested in different scenarios to assess Kafka's performance in handling requests.

The results of the study show that implementing asynchronous architecture and utilizing Kafka significantly improves the application's performance in handling document conversion request spikes. With higher throughput and lower latency compared to systems without Kafka, the system, configured with 2 brokers and partitions, was able to process documents ranging from 1 MB to 10 MB more quickly and efficiently. This research demonstrates that an asynchronous,

microservices-based architecture using a message broker is an optimal solution for addressing the challenges of large-scale, high-speed document processing.

Keywords: document conversion, asynchronous architecture, microservices, Apache Kafka, message broker, choreography.



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pada era digital saat ini penggunaan dokumen digital begitu marak karena telah menjadi kebutuhan penting bagi kebanyakan orang, terutama terlibat di lingkungan pekerjaan dan pendidikan. Dokumen digital sendiri terdiri dari banyak format dokumen seperti .docx, .pptx, .pdf, dsb. Dengan adanya banyak format dokumen tersebut, maka keperluan untuk mengubah format file semakin meningkat. Pengguna sering kali menemui situasi di mana mereka perlu mengonversi dokumen dari satu format ke format lain untuk berbagai keperluan, seperti berbagi informasi dengan pengguna lain, mencetak dokumen, atau mengintegrasikan data dari berbagai sumber. Dalam lingkungan kerja, konversi dokumen seringkali menjadi tugas rutin yang penting. Demikian pula, di dunia pendidikan, mahasiswa dan pengajar seringkali harus mengonversi dokumen agar sesuai dengan format yang diperlukan oleh institusi atau lembaga pendidikan mereka.

Munculnya aplikasi web konversi dokumen di tengah tren penggunaan dokumen digital ini adalah respons terhadap kebutuhan yang semakin besar. Dengan adanya aplikasi web konversi dokumen maka memiliki tujuan untuk mempermudah pengguna dalam mengubah konversi dokumen dengan jumlah file yang cukup banyak. Dengan beberapa klik, pengguna dapat mengubah format dokumen sesuai keinginan mereka. Oleh karena itu, aplikasi web konversi dokumen menjadi sangat populer karena memberikan kemudahan dan aksesibilitas yang tinggi bagi pengguna dari berbagai latar belakang. Namun, dengan bertambahnya pengguna yang mengakses layanan konversi dokumen secara bersamaan, muncul tantangan terkait pengelolaan permintaan dan efisiensi pemrosesan. Dengan adanya tantangan tersebut, maka dapat mengakibatkan masalah antrian permintaan/request yang memanjang. Hal tersebut dapat berpotensi membuat server down sehingga layanan gagal melakukan konversi dokumen, terutama dalam sistem dengan cara

kinerja monolith yang dimana rancangan arsitektur sebuah aplikasi yang menjalankan semua logika dalam satu server aplikasi (Daya et al., 2015). Tentu saja hal ini akan sangat merugikan pengguna yang sudah menunggu filenya diproses dengan tetap berada di halaman web namun tidak mendapatkan hasil akhir yang sesuai (Abdurohman, 2018). Oleh karena itu, diperlukan solusi yang efisien untuk mengatasi masalah ini.

Aspek arsitektur, terutama dalam konteks sisi asynchronous dan penggunaan arsitektur microservices, menjadi solusi efektif untuk mengatasi masalah lonjakan permintaan pada aplikasi web konversi dokumen. Arsitektur microservices memungkinkan pengembangan aplikasi sebagai layanan-layanan kecil yang independen, meningkatkan skalabilitas dan pengelolaan sumber daya, sementara komunikasi asynchronous meningkatkan efisiensi dengan memungkinkan pemrosesan tugas secara bersamaan (Ćatović et al., 2022). Adopsi arsitektur ini memungkinkan respons lebih cepat, menjaga efisiensi operasional, dan mengelola permintaan lebih baik, terutama saat terjadi lonjakan permintaan tinggi. Sebaliknya, arsitektur synchronous dalam sistem monolith memiliki kelemahan peningkatan kompleksitas dan investasi biaya yang tinggi untuk skala besar (Senduk et al., 2023). Penelitian ini bertujuan mengembangkan aplikasi web konversi dokumen dengan arsitektur asynchronous dan microservices menggunakan message broker, fokus pada efisiensi, responsivitas, dan pengalaman pengguna. Aplikasi ini mampu mengonversi berbagai format dokumen ke format .pdf, mengedit file .pdf, dan mengkonversi dokumen XML ke JSON. Arsitektur ini memungkinkan pemrosesan paralel, mengurangi penantian, dan mengelola lonjakan permintaan dengan bantuan message broker. Aplikasi terdiri dari dua layanan utama, konversi dokumen dan notifikasi, dengan model microservices dan arsitektur koreografi yang meningkatkan efisiensi dan ketahanan (Kalske et al., 2018; Senduk et al., 2023). Penelitian ini diharapkan meningkatkan kualitas aplikasi konversi dokumen, menjadikannya lebih kompetitif, dan memenuhi kebutuhan pengguna di masa depan.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan dapat dirumuskan masalah penelitian sebagai berikut:

1. Bagaimana penerapan arsitektur koreografi dalam menangani permasalahan antrian permintaan dalam konversi dokumen?
2. Bagaimana arsitektur koreografi dapat mengoptimalkan pengelolaan permintaan konversi dokumen dalam skenario penggunaan yang bersifat asinkron?
3. Bagaimana penggunaan message broker memengaruhi latensi dan performa aplikasi dalam proses konversi dokumen pada arsitektur microservices koreografi dibandingkan dengan sistem message broker?

1.3. Batasan Masalah

1. Dokumen yang dapat dikonversi hanya beberapa dokumen yang memiliki format open document di dalam strukturnya.
2. Sistem yang dibangun tidak membahas atau meneliti keamanan data dan desain tampilan web.
3. Penggunaan message broker hanya untuk layanan terkait membuat titik antrian dalam proses konversi dokumen saja.

1.4. Tujuan Penelitian

Penelitian yang akan dilakukan oleh penulis memiliki tujuan untuk mencari tahu tingkat efisiensi dari penggunaan arsitektur *asynchronous* pada layanan konversi dokumen berbasis web dalam mengatasi *request* dalam jumlah yang besar.

1.5. Manfaat Penelitian

Penelitian ini bermanfaat bagi para pengguna yang sering menggunakan dokumen digital dalam kehidupan sehari-hari, terutama pengguna yang

terlibat di lingkungan perkantoran dan pendidikan. Selain itu, juga dapat berguna bagi pembuatan program konversi web kedepannya agar tidak membutuhkan server dengan komputasi yang tinggi, sehingga dapat menghemat biaya yang akan dikeluarkan dalam pembuatan web.

1.6. Metodologi Penelitian

Proses meneliti selalu terdiri dari langkah-langkah melakukan penelitian. Langkah-langkah tersebut dijelaskan dalam metodologi penelitian.

1.7. Sistematika Penulisan

Laporan/Proposal skripsi ini disusun dengan sistematika bagian pertama, terdiri dari empat bab: Bab 1 yaitu Pendahuluan yang berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan pernyataan keaslian disertasi. Bab 2 yaitu Tinjauan Pustaka dan Landasan Teori yang berisi tinjauan pustaka tentang penelitian-penelitian terkait, dan berbagai tinjauan pustaka spesifik, yaitu tentang, Bab 3 yaitu Metodologi Penelitian, Bab 4 yaitu Implementasi dan Pembahasan, dan Bab 5 yaitu Kesimpulan dan Saran

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis terhadap kinerja ketiga konfigurasi yang diuji, ditemukan perbedaan signifikan dalam latensi dan tingkat keberhasilan (success rate) saat menangani paket data berukuran 1 MB, 5 MB, dan 10 MB dengan berbagai tingkat permintaan. Pada paket berukuran 1 MB, Kafka dengan satu broker (Konfigurasi 2) dan empat broker (Konfigurasi 3) mampu mempertahankan tingkat keberhasilan 100% hingga laju 100 permintaan per detik, sedangkan sistem tanpa Kafka (Konfigurasi 1) hanya berhasil mencapai tingkat keberhasilan 5% pada laju yang sama dan sepenuhnya gagal pada laju 1000 permintaan per detik. Untuk permintaan lebih tinggi, seperti pada paket 5 MB, Konfigurasi 1 juga tidak mampu mengatasi beban, sedangkan Konfigurasi 3 tetap menunjukkan stabilitas dengan latensi rata-rata sekitar 75839.83 ms pada 1000 permintaan per detik. Dengan adanya hasil tersebut, maka dapat disimpulkan bahwa sistem tanpa Kafka tidak dapat menangani beban permintaan tinggi secara efisien, sementara Kafka dengan konfigurasi broker memberikan keunggulan signifikan dalam menjaga keberhasilan sistem pada beban yang lebih berat.

Secara keseluruhan, konfigurasi Kafka dengan empat broker (Konfigurasi 3) menunjukkan performa paling unggul dalam menangani permintaan dengan volume data yang besar dan tingkat permintaan yang tinggi. Meskipun performa optimal ini membutuhkan tambahan memori pada Kafka dan server, peningkatan yang diperlukan tidak terlalu signifikan. Di sisi lain, kompleksitas konfigurasi bertambah dengan adanya elemen-elemen seperti replikasi, partisi, Zookeeper, Kafka cluster, offset, dan consumer group, yang memerlukan pengelolaan lebih lanjut untuk menjaga efektivitas dan stabilitas sistem dalam jangka panjang. Dengan demikian, meskipun konfigurasi yang lebih kompleks ini menawarkan efisiensi pemrosesan yang lebih tinggi, tantangan dalam pengelolaan dan

pemeliharaan sistem juga meningkat, sehingga keseimbangan antara performa dan kompleksitas perlu dipertimbangkan dalam implementasi Kafka.

5.2 Saran

Setelah menarik kesimpulan tersebut, ada beberapa saran yang dapat diajukan untuk pengembangan lebih lanjut dari sistem yang menggunakan Apache Kafka sebagai message broker dalam arsitektur microservices ini. Saran-saran tersebut adalah sebagai berikut:

- 1 Pengoptimalan Konfigurasi Kafka: Disarankan untuk melakukan optimasi lebih lanjut pada konfigurasi Kafka, terutama pada pengelolaan partisi dan replikasi, untuk meningkatkan performa sistem dalam menghadapi beban kerja yang lebih besar dan dinamis.
- 2 Integrasi Monitoring Real-Time: Mengintegrasikan alat monitoring seperti Prometheus dan Grafana untuk memantau performa Kafka dan layanan terkait secara real-time, sehingga potensi bottleneck atau kegagalan dapat terdeteksi dan ditangani lebih cepat.
- 3 Penambahan Dukungan Format Dokumen: Untuk memperluas cakupan layanan, disarankan menambahkan dukungan untuk lebih banyak format dokumen serta meningkatkan kecepatan pemrosesan, terutama untuk dokumen berukuran besar.
- 4 Load Balancing: Implementasi load balancing yang lebih baik, baik untuk Kafka maupun layanan lain, akan membantu memastikan kestabilan sistem saat terjadi lonjakan permintaan yang signifikan.
- 5 Pengujian Skala Infrastruktur Cloud: Melakukan pengujian lebih lanjut pada fitur auto-scaling di EC2 guna memastikan sistem dapat menyesuaikan dengan kebutuhan yang meningkat secara dinamis seiring pertumbuhan aplikasi.

Saran-saran tersebut diharapkan dapat menjadi pertimbangan yang baik dalam pengembangan sistem ini di masa mendatang.

DAFTAR PUSTAKA

- Abdurohman, A. (2018, November 28). Mengapa menggunakan message broker? Retrieved from Medium: <https://medium.com/@acep.abdurohman90/mengapa-menggunakan-message-broker-c17453cb225e>
- Bakshi, K. (2017). Microservices-based software architecture and approaches. IEEE Aerospace Conference Proceedings. <https://doi.org/10.1109/AERO.2017.7943959>
- Ćatović, A., Buzadžić, N., & Lemes, S. (2022). Microservice development using RabbitMQ message broker. Science, Engineering and Technology, 2(1), 30–37. <https://doi.org/10.54327/set2022/v2.i1.19>
- Daya, S., Van, N., Kameswara, D., Carlos, E., Ferreira, M., Glozic, D., Gucer, V., Gupta, M., Joshi, S., Lampkin, V., Martins, M., Narain, S., & Vennam, R. (2015). Redbooks Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach (Vol. 1).
- Fan, Q., & Wang, Q. (2016). Performance comparison of web servers with different architectures: A case study using high concurrency workload. Proceedings - 3rd Workshop on Hot Topics in Web Systems and Technologies, HotWeb 2015, 37–42. <https://doi.org/10.1109/HotWeb.2015.11>
- Fu, G., Zhang, Y., & Yu, G. (2021). A fair comparison of message queuing systems. IEEE Access, 9, 421–432. <https://doi.org/10.1109/ACCESS.2020.3046503>
- Hegde, R. G., & S, N. G. (2020). Low latency message brokers. International Research Journal of Engineering and Technology, May, 2731–2738.
- Hesse, G., Matthies, C., & Uflacker, M. (2021). How fast can we insert? An empirical performance evaluation of Apache Kafka

(Vol. 3).

John, V. (2017). A survey of distributed message broker queues.

Kalske, M., Mäkitalo, N., & Mikkonen, T. (2018). Challenges when moving from monolith to microservice architecture. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10544 LNCS, 32–47. https://doi.org/10.1007/978-3-319-74433-9_3

Nguyen, C. N., Lee, J., Hwang, S., & Kim, J. S. (2019). On the role of message broker middleware for many-task computing on a big-data platform. *Cluster Computing*, 22, 2527–2540. <https://doi.org/10.1007/s10586-018-2634-9>

Pongnumkul, S., Siripanpornchana, C., & Thajchayapong, S. (2017). Performance analysis of private blockchain platforms in varying workloads.

Riyanto, D. J., Safaat, N. H., & Affandes, M. (2022). Implementasi service choreography pattern arsitektur microservice classroom akademik menggunakan Docker. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7, 768–779.

Senduk, F. X., Najoran, X. B. N., & Sompie, S. R. U. A. (2023). Development of microservices architecture with RESTful API gateway using backend-for-frontend pattern in higher education academic portal. *Jurnal Teknik Informatika*, 18, 315–324.

Singh Gadwal, A., & Prasad, L. (2020). Comparative review of the literature of automated testing tools. <https://doi.org/10.13140/RG.2.2.36836.19848>

Wang, J., & Wu, J. (2019). Research on performance automation testing technology based on JMeter. *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*, 55–58. <https://doi.org/10.1109/ICRIS.2019.00023>