

**PENERAPAN CHOREOGRAPHY MESSAGE BROKER  
UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS  
RESTFUL**

Skripsi



oleh:

**DESENDO IMANUEL  
71180344**

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

2024

**PENERAPAN CHOREOGRAPHY MESSAGE BROKER  
UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS  
RESTFUL**

Skripsi



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

Disusun oleh

**DESENDO IMANUEL**

**71180344**

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA

2024

## PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **PENERAPAN CHOREOGRAPHY MESSAGE BROKER UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS RESTFUL**

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 skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi mana pun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

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

Yogyakarta, 18 Januari 2024



**DESENDO IMANUEL**

**71180344**

## HALAMAN PERSETUJUAN

Judul Skripsi : PENERAPAN CHOREOGRAPHY MESSAGE BROKER  
UNTUK TRANSAKSI DATA BERBASIS  
ASYNCHRONOUS RESTFUL

Nama Mahasiswa : DESENDO IMANUEL

N I M : 71180344

Mata Kuliah : Skripsi (Tugas Akhir)

Kode : TI0366

Semester : Gasal

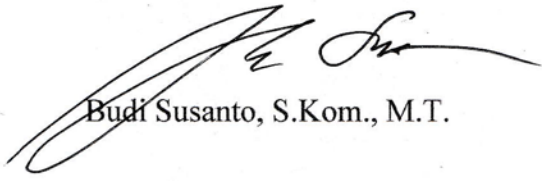
Tahun Akademik : 2023/2024

Telah diperiksa dan disetujui di  
Yogyakarta,  
Pada tanggal 18 Januari 2024

Dosen Pembimbing I

Dosen Pembimbing II

  
Maria Nila Anggia Rini, S.T., M.T.I.

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

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

---

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

Nama : Desendo Imanuel  
NIM : 71180344  
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 CHOREOGRAPHY MESSAGE BROKER UNTUK  
TRANSAKSI DATA BERBASIS ASYNCHRONOUS RESTFUL”**

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 : 24 Januari 2024

Yang menyatakan

(Desendo Imanuel)  
NIM.71180344

## HALAMAN PENGESAHAN

### PENERAPAN CHOREOGRAPHY MESSAGE BROKER UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS RESTFUL

Oleh: DESENDO IMANUEL / 71180344

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 20 Desember 2023

Yogyakarta, 18 Januari 2024  
Mengesahkan,

Dewan Penguji:

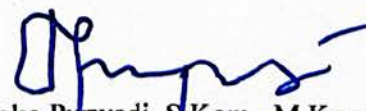
1. Maria Nila Anggia Rini, S.T., M.T.I.
2. Budi Susanto, S.Kom., M.T.
3. Aditya Wikan Mahastama, S.Kom.,  
M.Cs.
4. Antonius Rachmat C., S.Kom., M.Cs.

  
-----  
  
-----  
  
-----  
  
-----

Dekan

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

Ketua Program Studi

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

## **HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

---

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

Nama : Desendo Imanuel  
NIM : 71180344  
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 CHOREOGRAPHY MESSAGE BROKER UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS RESTFUL”**

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 : 24 Januari 2024

Yang menyatakan



Digitally signed  
by Desendo  
Immanuel  
Date: 2024.01.24  
10:26:30 +07'00'

(Desendo Imanuel)

NIM.71180344

## KATA PENGANTAR

Segala puji dan syukur kepada Tuhan yang maha kasih, karena atas segala rahmat, bimbingan, dan bantuan-Nya maka akhirnya Skripsi dengan judul “PENERAPAN CHOREOGRAPHY MESSAGE BROKER UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS RESTFUL” ini telah selesai disusun.

Penulis memperoleh banyak bantuan dari kerja sama baik secara moral maupun spiritual dalam penulisan Skripsi ini, untuk itu tak lupa Penulis ucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan yang maha kasih,
2. Orang tua—Ibu—yang selama ini telah sabar membimbing dan mendoakan Penulis tanpa kenal untuk selama-lamanya,
3. Keluarga tercinta atas dukungan, semangat, dan kasih sayang,
4. Restyandito, S.Kom, MSIS., Ph.D. selaku Dekan FTI,
5. Joko Purwadi, S.Kom., M.Kom. selaku Kaprodi Informatika,
6. Maria Nila Anggia Rini, S.T., M.T.I selaku Dosen Pembimbing 1, yang telah memberikan ilmunya dan dengan penuh kesabaran membimbing Penulis,
7. Budi Susanto, S.Kom., M.T. selaku Dosen Pembimbing 2 yang telah memberikan ilmu dan kesabaran dalam membimbing Penulis,
8. Willy Sudiarto Raharjo, S.Kom., M.Cs selaku Dosen, yang mendukung dalam hal penyediaan sumber daya penelitian,
9. Gani Indriyanto, Ir, MT. Selaku Dosen Wali, yang telah menjadi wali studi,
10. Rekan satu tim pengerjaan penelitian skripsi, Stefany Pischa Eva Angelifta selaku anggota tim *Frontend*, Yabes Qinen Yehdeya selaku anggota tim *Backend*,
11. Rekan-rekan seperjuangan yang tidak dapat Penulis sebutkan satu per satu,

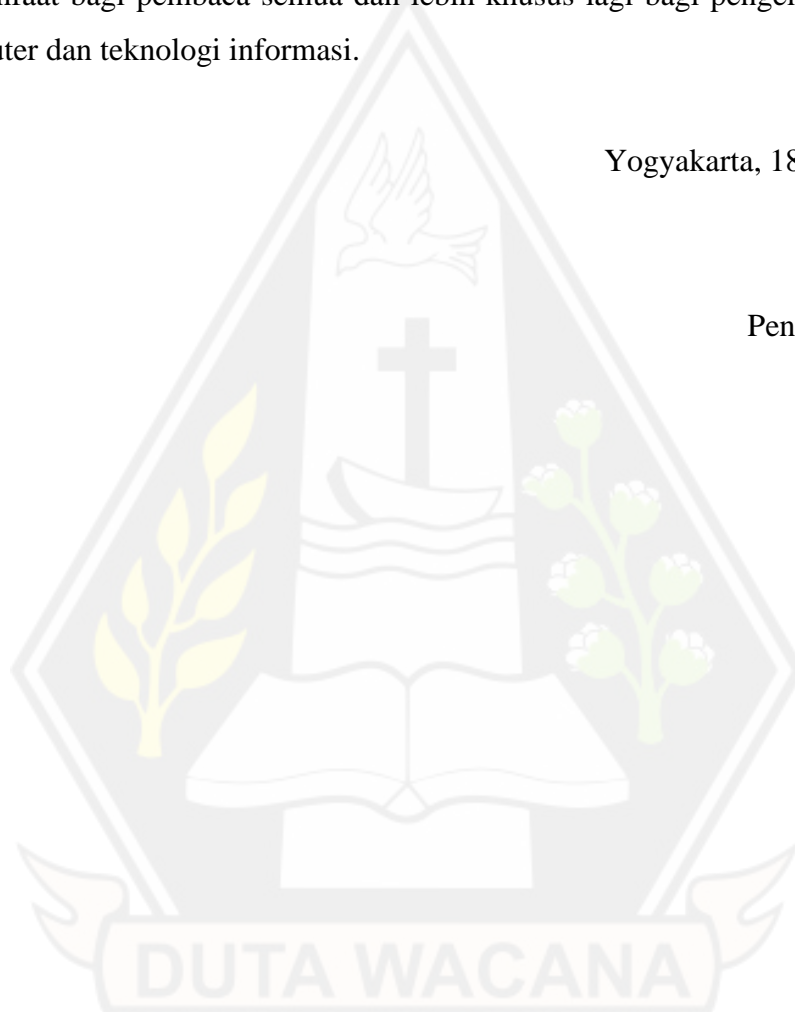


12. Lain-lain yang telah mendukung moral, spiritual, dan dana untuk belajar selama ini.

Laporan proposal/skripsi ini tentunya tidak lepas dari segala kekurangan dan kelemahan, untuk itu segala kritikan dan saran yang bersifat membangun guna kesempurnaan skripsi ini sangat diharapkan. Semoga proposal/skripsi ini dapat bermanfaat bagi pembaca semua dan lebih khusus lagi bagi pengembangan ilmu komputer dan teknologi informasi.

Yogyakarta, 18 Januari 2023

Penulis



## DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xiv
INTISARI.....	xviii
ABSTRACT.....	xx
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian .....	4
1.7 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Tinjauan Pustaka .....	7
2.2 Landasan Teori .....	10
2.2.1 Sistem Informasi Rumah Sakit.....	10
2.2.2 <i>Microservice</i> .....	10
2.2.3 <i>Choreography</i> .....	12
2.2.4 <i>Message Broker</i> .....	13
2.2.5 Saga.....	14
2.2.6 Aggregate .....	15
2.2.7 Domain Driven Design (DDD) .....	17

2.2.8	Command-Query Responsibility Segregation (CQRS)	17
2.2.9	Event Sourcing	21
2.2.10	Axon Framework	23
2.2.11	Docker	23
2.2.12	Spring Boot	25
2.2.13	Keycloak Server	26
BAB III METODOLOGI PENELITIAN		27
3.1	Alur Penelitian	27
3.2	Analisis Kebutuhan Sistem	29
3.2.1	Kebutuhan Fungsional	29
3.2.2	Kebutuhan Non-Fungsional	30
3.3	Use Case Diagram	30
3.3.1	[UC-1] Pengunjung	31
3.3.2	[UC-2] Pengguna	34
3.4	Perancangan Basis Data	36
3.5	Perancangan REST API	38
3.5.1	<i>Register New User</i>	39
3.5.2	<i>Register New Patient</i>	45
3.5.3	<i>Request OTP Code Sent via SMS by Phone Number</i>	47
3.5.4	<i>Verify OTP Code Sent via SMS by Phone Number</i>	48
3.5.5	<i>User Login by Phone Number</i>	50
3.5.6	<i>Get User Account Profile</i>	54
3.6	Diagram Proses Bisnis (BPMN)	57
3.6.1	<i>Data Query (Get/Fetch) Standard Model</i>	57
3.6.2	<i>Protected Data Query (Get/Fetch) Standard Model</i>	58
3.6.3	[UC-1-3] <i>OTP (One Time Password) Request</i>	60
3.6.4	[UC-1-4] <i>OTP (One Time Password) Verification</i>	61
3.6.5	[UC-1-2] <i>User Registration</i>	63
3.6.6	[UC-1-5] <i>User Login by Phone Number</i>	65
3.6.7	[UC-2-1] <i>Patient Registration</i>	66
3.7	Arsitektur Sistem	67

3.8	Poin Penting Dalam Pengembangan Sistem .....	69
3.8.1	Prototipe Sistem .....	69
3.8.2	Axon Server Sebagai <i>Message Broker</i> .....	69
3.8.3	Basis Data .....	70
3.8.4	Infrastruktur .....	70
3.9	Standarisasi.....	71
3.9.1	Struktur Folder .....	71
3.9.2	HTTP <i>Response Template</i> .....	72
3.9.3	<i>Library</i> .....	72
3.10	Perancangan Pengujian Sistem.....	73
BAB IV IMPLEMENTASI DAN ANALISIS SISTEM .....		75
4.1	Infrastruktur dan Konfigurasi Aplikasi .....	75
4.2	Penulisan Program.....	75
4.2.1	<i>Aggregate</i> .....	75
4.2.2	<i>Command</i> .....	76
4.2.3	<i>Event</i> .....	78
4.2.4	<i>Query</i> .....	79
4.2.5	Komunikasi Data.....	81
4.3	Pengujian dan Analisis .....	84
4.3.1	Tes Skenario: <i>Roll Back</i> atau <i>Compensating Transaction</i> .....	84
4.3.2	Tes Skenario: <i>Choreography</i> .....	92
4.3.3	Tes Skenario: Mengonsumsi <i>Event</i> .....	103
4.4	Pengujian: Unit Test, Component Test, dan Integration Test .....	109
4.4.1	<i>User Service</i> .....	110
4.4.2	<i>Auth Service</i> .....	114
BAB V KESIMPULAN DAN SARAN.....		116
5.1	Kesimpulan.....	116
5.2	Saran.....	117
DAFTAR PUSTAKA .....		118
LAMPIRAN A.....		120
KODE SUMBER PROGRAM .....		120

LAMPIRAN B .....	184
KARTU KONSULTASI DOSEN 1.....	184
LAMPIRAN C .....	186
KARTU KONSULTASI DOSEN 2.....	186
LAMPIRAN D.....	188
LAMPIRAN LAIN-LAIN .....	188



## DAFTAR TABEL

Tabel 3.1: [UC-1-2] Registrasi Pengguna.....	31
Tabel 3.2: [UC-1-3] Meminta Kode OTP via SMS Menggunakan Nomor Telepon .....	31
Tabel 3.3: [UC-1-4] Memverifikasi Kode OTP via SMS Menggunakan Nomor Telepon.....	32
Tabel 3.4: [UC-1-5] Memverifikasi Login Menggunakan Nomor Telepon .....	33
Tabel 3.5: [UC-2-1] Registrasi Pasien Baru .....	34
Tabel 3.6: [UC-2-3] Melihat Data Akun.....	35
Tabel 3.7: Daftar API.....	39



## DAFTAR GAMBAR

Gambar 2.1: Contoh Skenario Pemesanan menggunakan notasi BPM Sumber (Weske, 2019) .....	13
Gambar 2.2: Contoh Interaksi <i>Request-Response</i> Menggunakan <i>Message Queues</i> Sumber (Johansson & Dossot, 2020) .....	14
Gambar 2.3: Visualisasi Penerapan <i>Roll Back</i> atau <i>Compensating Transaction</i> Sumber (Richardson, 2019) .....	15
Gambar 2.4: Contoh Desain <i>Aggregate</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	16
Gambar 2.5: Contoh <i>Command</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	18
Gambar 2.6: Contoh Pengiriman <i>Command</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	19
Gambar 2.7: Contoh <i>Command Handler</i> Pada <i>Aggregate</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	20
Gambar 2.8: Contoh <i>Query Parameter</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	20
Gambar 2.9: Contoh <i>Query Request</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	20
Gambar 2.10: Contoh <i>Query Handler</i> Menggunakan Axon Framework Sumber (AxonIQ, 2023) .....	21
Gambar 2.11: Contoh Event Sourcing Menggunakan Axon Framework .....	22
Gambar 2.12: Contoh Penggunaan Docker dan Docker Compose Sumber (Docker, 2024) .....	25
Gambar 3.1: Diagram Alur Penelitian .....	27
Gambar 3.2: <i>Use Case Diagram</i> .....	30
Gambar 3.3: Desain Basis Data <i>Monolith</i> .....	37
Gambar 3.4: Desain <i>Aggregate Microservice Choreography</i> .....	38
Gambar 3.5: <i>Data Query (Get/Fetch) Standard Model</i> .....	58
Gambar 3.6: <i>Protected Data Query (Get/Fetch) Standard Model</i> .....	59

Gambar 3.7: Validasi Token .....	59
Gambar 3.8: [UC-1-3] OTP ( <i>One Time Password</i> ) Request .....	61
Gambar 3.9: [UC-1-4] OTP ( <i>One Time Password</i> ) Verification .....	62
Gambar 3.10: [UC-1-2] <i>User Registration</i> .....	64
Gambar 3.11: [UC-1-5] <i>User Login by Phone Number</i> .....	65
Gambar 3.12: [UC-2-1] <i>Patient Registration</i> .....	67
Gambar 3.13: Arsitektur Sistem.....	68
Gambar 4.1: <i>User Aggregate</i> .....	76
Gambar 4.2: CreateUserCommand.java .....	77
Gambar 4.3: Fungsi registerUser() Pada File UsersCommandServiceImpl.java .	78
Gambar 4.4: UserCreatedEvent.java.....	79
Gambar 4.5: UserDetailsQuery.java .....	80
Gambar 4.6: Fungsi getPatientProfiles() Pada File UsersQueryServiceImpl.java	80
Gambar 4.7: Fungsi fetchUser() Pada File UsersQueryHandler.java.....	80
Gambar 4.8: Konfigurasi <i>Tracking Processor</i> Pada Konfigurasi Kredensial <i>User Service</i> .....	83
Gambar 4.9: Konfigurasi <i>Tracking Processor</i> Pada Konfigurasi Kredensial <i>Auth Service</i> .....	83
Gambar 4.10: Alur Proses Tes Skenario: <i>Roll Back</i> atau <i>Compensating Transaction</i> .....	84
Gambar 4.11: Pemanggilan REST Meminta Kode OTP .....	85
Gambar 4.12: Entitas Kode OTP Pada <i>Projection Database</i> .....	85
Gambar 4.13: Pemanggilan REST Memverifikasi Nomor Telepon .....	86
Gambar 4.14: Entitas Kode OTP Pada <i>Projection Database</i> .....	86
Gambar 4.15: Toleransi Batas Atas <i>Timeout</i> Pemanggilan REST.....	87
Gambar 4.16: <i>Debugger Point</i> IdentityCreatedEvent.....	88
Gambar 4.17: Fungsi rejectUserRegistration() Terpicu.....	89
Gambar 4.18: <i>Roll Back</i> atau <i>Compensating Transaction</i> RejectUserRegistrationCommand.....	89
Gambar 4.19: <i>Roll Back</i> atau <i>Compensating Transaction</i> RejectUserRegistrationCommand.....	90



Gambar 4.20: <code>UserRegistrationRejectedEvent.java</code> .....	90
Gambar 4.21: <code>@EndSaga</code> dan <code>SagaLifeCycle.end()</code> .....	91
Gambar 4.22: Daftar <i>Events</i> Pada <i>Event Store</i> Axon Server Untuk Tes Skenario: <i>Roll Back</i> atau <i>Compensating Transaction</i> .....	91
Gambar 4.23: Alur Proses Tes Skenario: <i>Choreography</i> .....	92
Gambar 4.24 (a): Toleransi Batas Atas <i>Timeout</i> Pemanggilan REST.....	93
Gambar 4.25 (b): Toleransi Batas Atas <i>Timeout</i> Pemanggilan REST.....	93
Gambar 4.26: Pemanggilan REST Pendaftaran Pengguna Hasil Dengan <i>Status</i> <i>Code 500 Internal Server Error</i> .....	94
Gambar 4.27: Pemanggilan REST Pendaftaran Pengguna Hasil Dengan <i>Status</i> <i>Code 201 Created</i> .....	95
Gambar 4.28: Alur Proses Pendaftaran Pengguna.....	96
Gambar 4.29: Pengiriman <code>RegisterUserCommand</code> Dari Fungsi <code>registerUser()</code> Pada <i>File</i> <code>UsersCommandServiceImpl.java</code> .....	96
Gambar 4.30: Inisiasi Proses <i>Lifecycle Saga</i> Oleh <code>UserRegistrationEvent</code> Pada <i>File</i> <code>UserRegistrationSaga.java</code> .....	97
Gambar 4.31: <i>Saga</i> <code>UserRegistrationRejectedEvent</code> .....	97
Gambar 4.32: <i>Saga</i> <code>UserOtpVerifiedEvent</code> .....	98
Gambar 4.33: Konsumsi <code>IdentityCreatedEvent</code> Pada <i>File</i> <code>IdentityEventHandler.java</code> .....	99
Gambar 4.34: Konsumsi <code>ProjectionPersistedEvent</code> Pada <i>File</i> <code>UserRegistrationSaga.java</code> .....	99
Gambar 4.35: <i>Saga End Lifecycle</i> .....	100
Gambar 4.36: Fungsi <code>createUserKeycloak()</code> Pada <i>File</i> <code>OtpEventHandler.java</code> ..	100
Gambar 4.37: Fungsi <code>createUserKeycloak()</code> Pada <i>File</i> <code>KeycloakServiceImpl.java</code> .....	101
Gambar 4.38: Akun Pengguna Keycloak Server.....	101
Gambar 4.39: Daftar <i>Events</i> Pada <i>Event Store</i> Axon Server Untuk Tes Skenario: <i>Choreography</i> .....	102
Gambar 4.40: <i>Bug</i> <code>NullPointerException</code> Pada Proses <i>Saga</i> Pendaftaran Pengguna .....	103

Gambar 4.41: Alur Proses Tes Skenario: Mengonsumsi <i>Event</i> .....	103
Gambar 4.42: Pemanggilan REST Login Menggunakan Nomor Telepon .....	104
Gambar 4.43: Pemanggilan REST Registrasi Rumah Sakit .....	105
Gambar 4.44: Fungsi registerPatient() Pada <i>File</i> UsersCommandServiceImpl.java .....	105
Gambar 4.45: Konsumsi PatientRegisteredEvent Untuk Menyimpan Entitas Pada <i>Projection Database</i> .....	106
Gambar 4.46: Konsumsi PatientRegisteredEvent Pada PatientEventHandler di <i>Auth Service</i> .....	106
Gambar 4.47: Fungsi attachRealmRoleToUserKeycloak() Pada <i>File</i> KeycloakServiceImpl.java .....	107
Gambar 4.48: Fungsi attachRealmRolePatientToUserKeycloak() Pada <i>File</i> KeycloakServiceImpl.java .....	108
Gambar 4.49: Hasil Pemanggilan REST Registrasi Pasien .....	108
Gambar 4.50: Daftar <i>Events</i> Pada <i>Event Store</i> Axon Server Untuk Tes Skenario: Mengonsumsi <i>Event</i> .....	108
Gambar 4.51: Konfigurasi Tes Aplikasi <i>User Service</i> .....	109
Gambar 4.52: Konfigurasi Tes Aplikasi <i>Auth Service</i> .....	110
Gambar 4.53: Pengujian <i>User Service</i> - ServiceExceptionHandlerTest .....	110
Gambar 4.54: Pengujian <i>User Service</i> – <i>Aggregate</i> .....	111
Gambar 4.55: Pengujian <i>User Service</i> – <i>Data Model</i> .....	112
Gambar 4.56: Pengujian <i>User Service</i> – <i>Command, Event, dan Query</i> .....	113
Gambar 4.57: Pengujian <i>User Service</i> – <i>Saga</i> .....	113
Gambar 4.58: Pengujian <i>Auth Service</i> - AWS SNS .....	114
Gambar 4.59: Pengujian <i>Auth Service</i> – <i>Aggregate</i> .....	114
Gambar 4.60: Pengujian <i>Auth Service</i> - <i>Data Model</i> .....	115
Gambar 4.61: Pengujian <i>Auth Service</i> - <i>Event Handler</i> dan <i>Keycloak Service</i> ..	115

## INTISARI

### **PENERAPAN CHOREOGRAPHY MESSAGE BROKER UNTUK TRANSAKSI DATA BERBASIS ASYNCHRONOUS RESTFUL**

Oleh

DESENDO IMANUEL

71180344

Sebuah lembaga sosial gerejawi yang bergerak di bidang kesehatan, Yayasan Kristen untuk Kesehatan Umum (YAKKUM), saat ini memiliki rumah sakit yang tersebar di beberapa tempat seperti: Yogyakarta, Semarang, Surakarta (Solo), Purwokerto, Purworejo, dan Parakan. Saat ini, rumah sakit yang berada di bawah yayasan YAKKUM menjalankan sistem informasi secara independen yang berarti masing-masing rumah sakit membangun dan mengelola sistem informasinya secara mandiri tanpa adanya keterlibatan secara sistematis dengan rumah sakit lainnya. Hal tersebut memicu munculnya sifat heterogen pada data sehingga diperlukannya integrasi data.

Pada aplikasi skala *enterprise*, masalah integrasi antar komponen sistem aplikasi akan bertambah seiring dengan meningkatnya kebutuhan dan kompleksitas aplikasi tersebut. Masalah tersebut akan menciptakan masalah  $N \times N$  atau meningkatnya sifat ketergantungan antar *service* di mana  $N$  merupakan jumlah komponen aplikasi yang saling terintegrasi. Salah satu cara mengatasi hal tersebut adalah mengimplementasikan *middleware* dengan pendekatan *choreography* sebagai media penengah antar *service*.

Metode pengujian dilakukan dengan *unit test*, *component test*, *integration test*, menguji dan menginspeksi keandalan jalur komunikasi data antar *service*. Pada implementasi, sistem menggunakan Axon Server sebagai *message broker* yang dikonfigurasi dengan *event handler* dengan mode *tracking processor*. Setiap

*service* akan memiliki *database* masing-masing yang didukung dengan penerapan infrastruktur menggunakan Docker. Sistem akan diuji berdasarkan kemampuannya melakukan *roll back* atau *compensating transaction*, *choreography*, dan *event consumption*. Hasil pengujian menyatakan semua kebutuhan fungsional terpenuhi bahkan ketika *service* melakukan *reboot* di tengah-tengah transaksi. Tetapi di samping itu, hasil pengujian juga menunjukkan adanya kejangalan atau *bug* pada beberapa alur program.

**Kata-kata kunci** : *message broker, microservice, choreography, event-driven, container, docker, axon, keycloak, rest, api*



## **ABSTRACT**

### **IMPLEMENTATION OF CHOREOGRAPHY MESSAGE BROKER FOR ASYNCHRONOUS RESTFUL DATA TRANSACTIONS**

By

DESENDO IMANUEL

71180344

A social-religious institution engaged in the field of health, the Christian Foundation for Public Health (Yayasan Kristen untuk Kesehatan Umum - YAKKUM), currently operates hospitals in various locations such as Yogyakarta, Semarang, Surakarta (Solo), Purwokerto, Purworejo, and Parakan. Presently, the hospitals under the YAKKUM Foundation run independent information systems, meaning each hospital builds and manages its information system independently without systematic involvement with other hospitals. This leads to the emergence of heterogeneous data, necessitating data integration.

In enterprise-scale applications, integration issues among application system components will increase with the growing needs and complexity of the application. These issues will create  $N \times N$  problems or an increase in the interdependence nature among services, where  $N$  represents the number of integrated application components. One way to address this is by implementing middleware with a choreography approach as an intermediary between services.

Testing methods are conducted with unit tests, component tests, integration tests, and inspecting the reliability of data communication paths between services. In the implementation, the system uses Axon Server as a message broker configured with an event handler in tracking processor mode. Each service will have its own database supported by the infrastructure implementation using Docker. The system will be tested based on its ability to

perform roll back or compensating transactions, choreography, and event consumption. Test results indicate that all functional requirements are met, even when services undergo a reboot in the middle of a transaction. However, alongside this, the test results also reveal anomalies or bugs in some program flows.

**Keywords** : *message broker, microservice, choreography, event-driven, container, docker, axon, keycloak, rest, api*



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Sistem informasi rumah sakit merupakan salah satu hal yang menjadi faktor penentu efisiensi penggunaan sumber daya bagi rumah sakit dalam mencapai tujuan strategis atau visi serta memberikan layanan atau jasa kepada masyarakat secara maksimal. Sebuah lembaga sosial gerejawi yang bergerak di bidang kesehatan, Yayasan Kristen untuk Kesehatan Umum (YAKKUM), saat ini memiliki rumah sakit yang tersebar di beberapa tempat seperti: Yogyakarta, Semarang, Surakarta (Solo), Purwokerto, Purworejo, dan Parakan (Yayasan Kristen untuk Kesehatan Umum, 2022). Saat ini, rumah sakit yang berada di bawah yayasan YAKKUM menjalankan sistem informasi secara independen yang berarti masing-masing rumah sakit membangun dan mengelola sistem informasinya secara mandiri tanpa adanya keterlibatan secara sistematis dengan rumah sakit lainnya. Dengan begitu, perubahan data pada suatu sistem informasi tidak teraplikasikan pada sistem informasi lainnya. Hal tersebut memicu munculnya sifat heterogen atau keheterogenan pada data yang digunakan pada sistem informasi terdistribusi dan diperlukan melakukan integrasi data untuk mengatasi hal tersebut (Weske, 2019).

Dalam pengembangan aplikasi skala *enterprise*, masalah integrasi antar komponen sistem aplikasi akan muncul dan bertambah seiring dengan meningkatnya kompleksitas aplikasi tersebut. Pada umumnya, masalah ini akan menciptakan masalah *N x N problem* atau meningkatnya sifat ketergantungan antar *service* berupa *data coupling* di mana *N* merupakan jumlah komponen aplikasi yang akan diintegrasikan. Masalah tersebut biasanya lebih sering muncul pada aplikasi *enterprise* yang dibangun dengan pendekatan *point-to-point integration* di mana masing-masing komponen sistem aplikasi dapat secara langsung berinteraksi dan dibangun dengan basis data secara independen (Weske,

2019). Salah satu cara mengatasi hal tersebut adalah menerapkan *middleware* atau komponen perangkat lunak dalam menjembatani komunikasi antar komponen sistem aplikasi atau *services*. Dalam implementasinya, digunakan sistem arsitektur *microservices* dengan memanfaatkan pendekatan *choreography* berbasis *event-driven*. Ide utama dari implementasi *choreography* adalah untuk mengurangi sifat *tight coupling* serta meningkatkan sifat *high cohesion* terhadap satu atau lebih *service* dengan memanfaatkan *message broker* sebagai perantara komunikasi antar *service*. Komunikasi tidak lagi dilakukan antar *service*, melainkan akan dilakukan secara *asynchronous* dengan melibatkan *message broker* sebagai penengah/perantara (*middleware*). *Service* akan memiliki peran sebagai *producer* atau *consumer*, sedangkan *message broker* akan memiliki peran untuk menyampaikan proses transaksi data yang bersifat *asynchronous* (Bellemare, 2020). Dengan begitu, sifat ketergantungan tidak lagi berada pada *service* melainkan akan berpusat pada *message broker*.

Dalam penelitian ini, Penulis akan menggunakan sampel data yang akan menyimulasikan data pasien riil. Proses pendaftaran akun baru akan berperan sangat penting pada proses pengumpulan data dan akan disebut sebagai *event* pendaftaran akun baru. Proses pendaftaran tersebut akan memanfaatkan peran *message broker* sebagai perantara dalam hal pertukaran data untuk menyampaikan *event* antar *service*. Proses pendaftaran akan dibagi menjadi dua tahap, yaitu pengisian data pada formulir registrasi akun baru dan verifikasi data kontak (nomor telepon) yang memiliki sub-proses masing-masing dan akan disebar ke beberapa *service* melalui *message broker* untuk dikerjakan lebih lanjut. Dalam hal pengaksesan sistem, Peneliti menerapkan API secara RESTful. Pada penelitian ini, Penulis hanya fokus pada implementasi transaksi data berbasis *asynchronous REST (Representational State Transfer)* pada sistem terdistribusi yang melibatkan lebih dari satu *service*.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah pada penelitian ini adalah keberhasilan penerapan *message broker* sebagai *middleware* dalam



menangani masalah integrasi data pada transaksi yang melibatkan lebih dari satu *service*.

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Penelitian dan pengembangan hanya mengembangkan *service* yang terlibat secara langsung dalam proses pendaftaran akun baru.
2. Penelitian dan pengembangan adalah untuk membuat aplikasi prototipe.
3. Data yang digunakan adalah data *dummy* yang menyerupai data pasien riil.
4. Proses bisnis pada sistem akan disesuaikan dengan prosedur yang diterapkan di rumah sakit pada umumnya.
5. Sistem dibangun dengan *framework* Spring Boot berbasis bahasa pemrograman Java.
6. Sistem dibangun dengan mendukung arsitektur pengembangan web REST (*Representational State Transfer*).
7. Mengimplementasikan Axon Server sebagai *message broker*.
8. Tidak menerapkan dan memperhatikan sisi keamanan (*security*) secara mendalam/mendetail.
9. Tidak menerapkan pengujian *unit*, *component*, dan *integration* secara keseluruhan ke semua *file* yang ada, melainkan kepada *file-file* yang mempunyai imbas ke penelitian saja. Sebagai pengecualian, *mocking* diterapkan pada fungsi pengiriman kode OTP via SMS.

### 1.4 Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah untuk menerapkan konsep *choreography* pada transaksi data berbasis *asynchronous* yang melibatkan lebih dari satu *service* menggunakan *message broker* sebagai *middleware* dalam hal komunikasi data.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian adalah sebagai berikut.

1. Memberikan alternatif pilihan bagi calon pasien dalam melakukan pendaftaran keanggotaan rumah sakit yang dapat dilakukan secara daring.
2. Penerapan REST API memberikan akses *multi-platform* yang fleksibel.
3. Membangun arsitektur sistem aplikasi yang mendukung pengembangan lanjutan dalam jangka panjang.
4. Penerapan *design pattern* dalam pengembangan aplikasi kiranya dapat mempermudah dan mempercepat pengembangan aplikasi lanjutan.

## 1.6 Metodologi Penelitian

Dalam penelitian ini terdapat beberapa langkah-langkah penelitian sebagai berikut.

1. Pengumpulan daftar kebutuhan

Penulis akan melakukan literasi dan mempelajari studi terkait penelitian yang serupa yang didapat dari studi literatur dalam bentuk buku, jurnal, dokumentasi perangkat lunak, atau situs daring lainnya yang mendukung penelitian. Berikutnya dilakukan pengumpulan daftar kebutuhan fungsional dan non-fungsional yang dibutuhkan dalam membangun sistem. Daftar kebutuhan tersebut diperoleh dengan mengadakan sesi tanya jawab secara daring dengan orang yang terlibat dalam pengembangan sistem. Pada tahap ini dilakukan perancangan sistem aplikasi dengan membuat model *use case* dan pemodelan proses bisnis. Tahap berikutnya akan dilanjutkan untuk mendesain basis data dan mendesain API *endpoint*.
2. Perancangan arsitektur sistem

Merancang desain arsitektur sistem aplikasi berdasarkan daftar kebutuhan yang diperoleh di tahap sebelumnya. Pada tahap ini dilakukan perancangan sistem aplikasi dengan membuat desain/rancangan diagram arsitektur sistem. Pada tahap ini Penulis akan mengimplementasikan juga desain arsitektur sistem.
3. Pengembangan sistem

Penulis akan mulai melakukan implementasi arsitektur sistem berdasarkan rancangan yang sudah dibuat pada tahap sebelumnya. Pada tahap ini, Penulis juga akan secara bertahap mempelajari kebutuhan teknis beserta teknologi yang digunakan dalam penelitian.

#### 4. *Testing*

Penulis akan melakukan pengujian terhadap sistem yang dibangun guna memastikan pengembangan sistem sesuai dengan daftar kebutuhan pada dokumen daftar kebutuhan.

#### 5. Analisis penerapan *message broker*

Dalam tahap ini, Penulis akan melakukan sekali lagi proses *testing* untuk melakukan evaluasi dan penilaian terhadap sistem aplikasi yang dibangun.

### 1.7 Sistematika Penulisan

Dalam penelitian ini,, sistematika penulisan akan dibagi menjadi 5 bab, yaitu sebagai berikut.

#### 1. Bab 1 – Pendahuluan

Bab ini mencakup pembahasan mengenai latar belakang, masalah yang ingin diteliti, dan alur penelitian.

#### 2. Bab 2 – Tinjauan Pustaka dan Dasar Teori

Bab ini berisikan pembahasan tentang dasar teori yang akan digunakan pada penelitian dan studi yang sebelumnya sudah pernah dilakukan oleh orang lain yang juga memiliki hubungan dengan penelitian ini.

#### 3. Bab 3 – Metodologi Penelitian

Pada bab ini Penulis akan membahas perincian penelitian serta metode yang digunakan untuk mengevaluasi sistem aplikasi yang ingin dibangun.

#### 4. Bab 4 – Implementasi dan Pembahasan

Bab ini berisikan penjabaran dan pembahasan mengenai sistem yang sudah diimplementasikan. Pada bab ini juga Penulis akan melakukan pengujian (*testing*) terhadap sistem yang sudah dibangun.

#### 5. Bab 5 – Kesimpulan

Pada bab ini Penulis akan melakukan evaluasi dan penarikan kesimpulan terhadap hasil pengujian sistem aplikasi yang sudah diimplementasikan.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

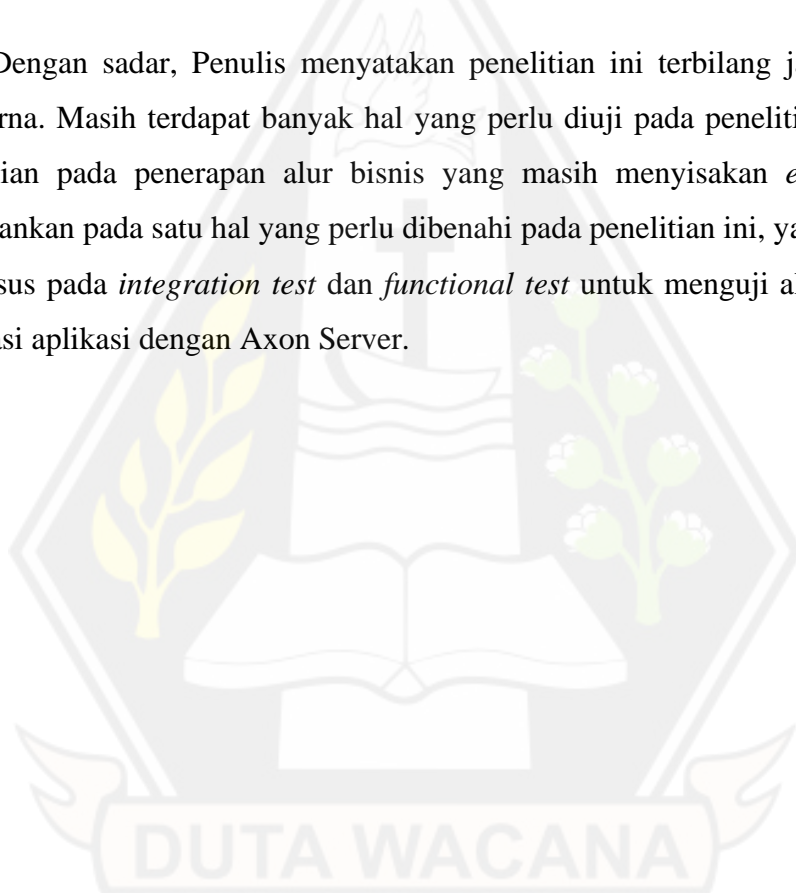
Kesimpulan dari penelitian ini adalah sebagai berikut.

1. Berdasarkan hasil pengujian, penerapan Axon Server sebagai *message broker* pada penelitian ini terbilang berhasil dengan pengecualian sebagai berikut.
  - Pada pengujian *compensating transaction*, diambil kesimpulan bahwa integritas data pada saat melakukan transaksi data dapat dijaga yang dibuktikan dengan berhasilnya dalam menerapkan *roll back* atau *compensating transaction*. Meski begitu, program masih meninggalkan *intermittent error* pada alur program terkhusus pada beberapa skenario yang Penulis tidak dapat identifikasi lebih lanjut.
  - Pengujian *choreography* mengindikasikan bahwa penerapan transaksi data berhasil diterapkan pada lingkungan *microservice* dengan pendekatan *choreography* pada lebih dari satu *service* (*User Service* dan *Auth Service*). Hal tersebut dibuktikan dengan keberhasilan penerapan komponen *event handler* dalam memproyeksikan data ke *projection database* dan penerapan pengiriman kode OTP yang melibatkan dua *service*, yaitu *User Service* dan *Auth Service*. Namun begitu, program masih meninggalkan *intermittent error* yang diasumsikan pada implementasi *saga* atau *aggregate*.
  - Pengujian terhadap *event handler* pada *User Service* dan *Auth Service* berhasil mengonsumsi *event* dengan lancar tanpa kendala. Hal ini mengindikasikan bahwa berhasilnya penerapan integrasi data dari *User Service* ke *Auth Service*. Hal tersebut juga terimplementasi pada proses pendaftaran akun Pengguna yang mana data Pengguna pada *User Service* akan digunakan oleh *Auth Service* untuk meregistrasikan Pengguna pada Keycloak Server.

2. Daftar kebutuhan fungsional terpenuhi berdasarkan pengujian *unit*, *component*, dan *integration*.
3. Daftar kebutuhan non-fungsional terpenuhi berdasarkan implementasi perancangan sistem, yaitu Axon Server sebagai *message broker*; arsitektur *microservice* yang mengadopsi konsep *choreography*; basis data relasional menggunakan MySQL; Keycloak Server dalam mengimplementasikan layanan autentikasi pada mekanisme *login*.

## 5.2 Saran

Dengan sadar, Penulis menyatakan penelitian ini terbilang jauh dari kata sempurna. Masih terdapat banyak hal yang perlu diuji pada penelitian ini seperti pengujian pada penerapan alur bisnis yang masih menyisakan *error*. Penulis menekankan pada satu hal yang perlu dibenahi pada penelitian ini, yaitu pengujian terkhusus pada *integration test* dan *functional test* untuk menguji alur bisnis dan integrasi aplikasi dengan Axon Server.



## DAFTAR PUSTAKA

- AxonIQ. (2023, Desember 01). *Axon Reference Guide*. Diambil kembali dari Axon Reference Guide: <https://docs.axoniq.io/reference-guide/>
- Bellemare, A. (2020). *Building Event-Driven Microservices*. O'Reilly Media, Inc.
- Docker. (2024, Januari 17). *awesome-compose*. Diambil kembali dari GitHub, Inc.: <https://github.com/docker/awesome-compose/tree/master/sparkjava-mysql>
- Docker, Inc. (2023, Desember 01). *Docker Overview*. Retrieved from Docs Docker: <https://docs.docker.com/get-started/overview/>
- Haux, R., Ammenwerth, E., Winter, A., & Brigl, B. (2004). *Strategic Information Management in Hospitals: An Introduction to Hospital Information Systems*. Springer Science & Business Media New York.
- Hong, X. J., Yang, H. S., & Kim, Y. H. (2018). Performance Analysis of RESTful API and RabbitMQ for Microservice Web Application. *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 257-259.
- Johansson, L., & Dossot, D. (2020). *RabbitMQ Essentials: Build distributed and scalable applications with message queuing using RabbitMQ*. Packt Publishing.
- Keycloak. (2023, Desember 1). *Server Administration Guide Version 22.0.5*. Diambil kembali dari Server Administration Documentation: [https://www.keycloak.org/docs/22.0.5/server\\_admin/](https://www.keycloak.org/docs/22.0.5/server_admin/)
- Malik, B. H., Dar, Z. M., Kayani, S. M., Dar, M., Shafiq, M. H., Kabir, I., . . . Ali, A. (2018). University Notification Subscription System using Amazon Web Service. *International Journal of Advanced Computer Science and Applications*, 9(5), 349-354. doi:10.14569/IJACSA.2018.090545
- Marcos-Pablos, S., García-Holgado, A., & García-Peñalvo, F. J. (2019, Oktober). Modelling the business structure of a digital health ecosystem. *Proceedings of the Seventh International Conference on Technological*

*Ecosystems for Enhancing Multiculturalism*, 838-846.

doi:10.1145/3362789.3362949

Nadareishvili, I., Mitra, R., McLarty, M., & Amundsen, M. (2016). *Microservice Architecture: Aligning Principles, Practices, and Culture*. O'Reilly Media, Inc.

Richardson, C. (2019). *Microservices Patterns With Examples in Java*. Manning Publications Co.

Rudrabhatla, C. K. (2018). Comparison of Event Choreography and Orchestration Techniques in Microservice Architecture. *International Journal of Advanced Computer Science and Applications*, 9(8), 18-22.

doi:10.14569/IJACSA.2018.090804

Spring Projects. (2023, Desember 01). *Spring Boot*. Diambil kembali dari Spring: <https://spring.io/projects/spring-boot>

Spring Projects. (2023, Desember 01). *Spring Data JPA*. Diambil kembali dari Spring: <https://spring.io/projects/spring-data-jpa>

Weske, M. (2019). *Business Process Management*. Heidelberg: Springer Berlin.

Yayasan Kristen untuk Kesehatan Umum. (2022, Juni 16). *Unit Layanan*. Dipetik Juni 16, 2022, dari Yayasan Kristen untuk Kesehatan Umum: <https://yakkum.or.id/unit-layanan/>