

**PENERAPAN CHOREOGRAPHY-BASED EVENT-  
DRIVEN MICROSERVICE DENGAN MESSAGE BROKER  
SERVICE UNTUK PORTAL OBJEK BUDAYA**

**SKRIPSI**



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

**2023**

**PENERAPAN CHOREOGRAPHY-BASED EVENT-  
DRIVEN MICROSERVICE DENGAN MESSAGE BROKER  
SERVICE UNTUK PORTAL OBJEK BUDAYA**

Skripsi



Diajukan kepada Fakultas Teknologi Informasi Program Studi Informatika

Universitas Kristen Duta Wacana

Sebagai salah satu syarat dalam memperoleh gelar Sarjana Komputer

Diajukan oleh:

**MICHAEL PANDU NURSETO**

**71180260**

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA

2023

## PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **PENERAPAN CHOREOGRAPHY-BASED EVENT-DRIVEN MICROSERVICE DENGAN MESSAGE BROKER SERVICE UNTUK PORTAL OBJEK BUDAYA**

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 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 skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar keserjanaan saya.

Yogyakarta, 7 Februari 2023



MICHAEL PANDU NURSETO  
71180260

## HALAMAN PERSETUJUAN

Judul Skripsi : PENERAPAN CHOREOGRAPHY-BASED EVENT-  
DRIVEN MICROSERVICE DENGAN MESSAGE  
BROKER SERVICE UNTUK PORTAL OBJEK  
BUDAYA

Nama Mahasiswa : MICHAEL PANDU NURSETO

N I M : 71180260

Matakuliah : Skripsi (Tugas Akhir)

Kode : TI0366

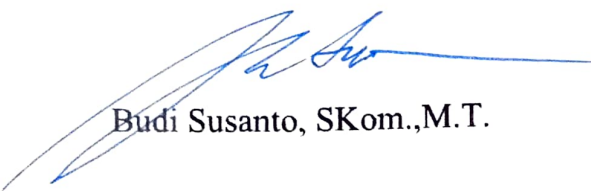
Semester : Gasal

Tahun Akademik : 2022/2023

Telah diperiksa dan disetujui di  
Yogyakarta,  
Pada tanggal 7 Februari 2023

Dosen Pembimbing I

Dosen Pembimbing II

  
Budi Susanto, SKom.,M.T.

  
Gloria Virginia, S.Kom., MAI, Ph.D.

## HALAMAN PENGESAHAN

### PENERAPAN CHOREOGRAPHY-BASED EVENT-DRIVEN MICROSERVICE DENGAN MESSAGE BROKER SERVICE UNTUK PORTAL OBJEK BUDAYA

Oleh: MICHAEL PANDU NURSETO / 71180260

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 11 Januari 2023


Yogyakarta, 7 Februari 2023  
Mengesahkan,

Dewan Penguji:

1. Budi Susanto, SKom., M.T.
2. Gloria Virginia, S.Kom., MAI, Ph.D.
3. Maria Nila Anggia Rini, S.T, M.T.I
4. I Kadek Dendy S., S.T., M.Eng.



Dekan



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

Ketua Program Studi



(Gloria Virginia, Ph.D.)



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

**UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA**

Saya yang bertanda tangan di bawah ini:

NIM : 71180260  
Nama : Michael Pandu Nurseto  
Prodi / Fakultas : Informatika / Teknologi Informasi  
Judul Tugas Akhir : PENERAPAN CHOREOGRAPHY-BASED  
EVENT-DRIVEN MICROSERVICE DENGAN  
MESSAGE BROKER SERVICE UNTUK  
PORTAL OBJEK BUDAYA

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan **Hak Bebas Royalti Non Eksklusif** (*Non-exclusive Royalty-free Right*) serta bersedia Tugas Akhirnya dipublikasikan secara online dan dapat diakses secara lengkap (*full access*).

Dengan Hak Bebas Royalti Noneklusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk *database*, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 9 February 2023

Yang menyatakan,

  
P.P.

**71180260 – MICHAEL PANDU NURSETO**



Karya sederhana ini dipersembahkan

kepada Tuhan, Keluarga Tercinta,

dan Kedua Orang Tua



*Segala sesuatu indah pada waktu-Nya*

Anonim

*Perjalanan ribuan mil dimulai dari langkah satu mil*

(Pepatah Kuno)



## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala berkat, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan pengerjaan dan penulisan skripsi berjudul **“PENERAPAN CHOREOGRAPHY-BASED EVENT-DRIVEN MICROSERVICE DENGAN MESSAGE BROKER SERVICE UNTUK PORTAL OBJEK BUDAYA”** ini dengan baik.

Skripsi ini disusun dan dikerjakan oleh penulis dengan tujuan memenuhi mata kuliah Skripsi dan sebagai salah satu syarat perolehan gelar Sarjana Program Studi Informatika pada Fakultas Teknologi Informasi Universitas Kristen Duta Wacana.

Dalam penulisan dan pengerjaan skripsi ini penulis telah mendapatkan bimbingan, doa, bantuan, saran dan dukungan dari berbagai pihak. Maka dari itu, peneliti tidak lupa mengucapkan banyak terima kasih kepada:

1. Tuhan Yesus Kristus yang telah mencurahkan berkat, rahmat, dan karunia-Nya yang melimpah.
2. Bapak Restyandito, S.Kom., MSIS., Ph.D., selaku Dekan Fakultas Teknologi Informasi Universitas Kristen Duta Wacana Yogyakarta.
3. Ibu Gloria Virginia, S.Kom., MAI., Ph.D., selaku Ketua Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana Yogyakarta dan dosen pembimbing 2 yang telah membimbing dan membantu penulis dalam penyusunan dan penulisan skripsi.
4. Bapak Budi Susanto, S.Kom., MT., selaku dosen pembimbing 1 yang telah membimbing dan mengarahkan penulis dalam pengerjaan dan penulisan skripsi.
5. Ibu, Tante dan Adik laki-laki penulis, “Novaliana Anastasia Hanggi, Angela Hanggi dan Marcellino Khrisna Nursetyo” yang selalu memberikan semangat, kasih sayang, dan dukungan pada penulis.
6. Alexander Ryu Siedharta, Petronella Ayu Ragil Kusuma, dan Joshua Christian Susabda selaku teman dekat penulis yang telah memberikan

semangat dan dukungan dalam proses pengerjaan dan penulisan skripsi.

7. Teman-teman dalam tim pengembang Portal Objek Budaya Indonesia, Angger Herlambang Amandegani dan Kristofan Feriadi yang memberikan dukungan dan bantuan dari awal hingga akhir masa perkuliahan.
8. Ananda Apriliansyah Cahya Utama, Brian Bastian, Dimas Aji Setiawan, Fransiskus Deli Sunarso, Jovan Roderick Reinaldo, Jonathan El-Roi Setiawan, Rusel Alexander, Timotius Johan Wijaya, Yohanes Tennary Rinto Pradana, dan seluruh teman-teman penulis yang tidak dapat disebutkan satu persatu atas segala dukungan dan bantuan yang diberikan kepada penulis selama masa perkuliahan dan penulisan skripsi.

Penulis memiliki harapan semoga skripsi yang telah disusun ini dapat memberikan manfaat dan inspirasi sekaligus motivasi bagi pembaca. Dalam penulisan skripsi ini tentunya penulis sadar bahwa masih terdapat banyak kekurangan baik dalam penyusunan kata dan kalimat sehingga kritik dan saran dari pembaca sangat diharapkan demi kesempurnaan penyusunan skripsi ini.

Yogyakarta, 5 Desember 2022

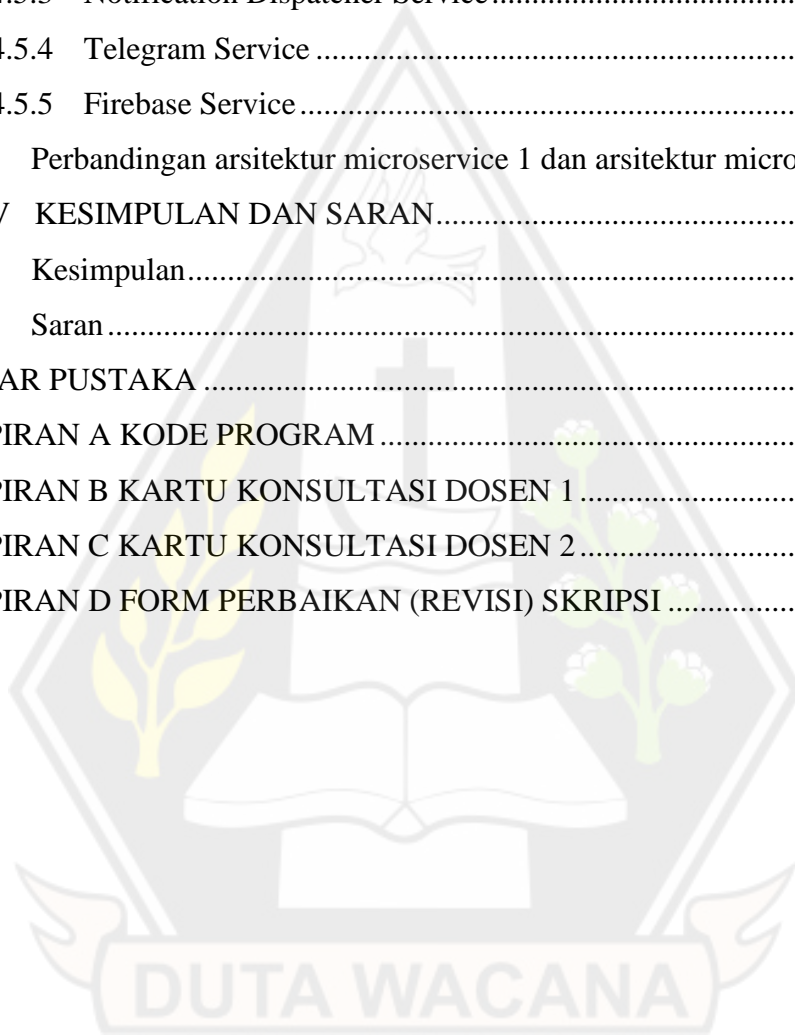
Michael Pandu Nurseto

## DAFTAR ISI

PENERAPAN CHOREOGRAPHY-BASED EVENT-DRIVEN MICROSERVICE DENGAN MESSAGE BROKER SERVICE UNTUK PORTAL OBJEK BUDAYA.....	i
PENERAPAN CHOREOGRAPHY-BASED EVENT-DRIVEN MICROSERVICE DENGAN MESSAGE BROKER SERVICE UNTUK PORTAL OBJEK BUDAYA.....	ii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI .....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE.....	vi
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA .....	vi
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR .....	xv
DAFTAR SCRIPT .....	xxii
INTISARI.....	xxiv
ABSTRACT.....	xxv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian .....	3
1.6.1 Metode Penelitian .....	3
1.7 Sistematika Penulisan.....	4

<b>BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI</b> .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Landasan Teori .....	7
2.2.1 Portal Objek Budaya .....	7
2.2.2 Arsitektur Microservice .....	8
2.2.3 Arsitektur Event-Driven .....	8
2.2.4 REST API / Representational State Transfer Application Programming Interface .....	9
2.2.5 Middleware .....	9
2.2.6 Message Broker Service .....	10
<b>BAB III METODOLOGI PENELITIAN</b> .....	13
3.1 Blok Diagram Sistem .....	13
3.2 Kebutuhan Sistem.....	15
3.2.1 Kebutuhan Perangkat Lunak.....	15
3.2.2 Kebutuhan Perangkat Keras.....	15
3.3 Metode Penelitian.....	16
3.3.1 Analisis Kebutuhan Service.....	17
3.3.2 Desain Service .....	23
3.3.3 Implementasi Kode Service .....	28
3.3.4 Evaluasi Service.....	28
<b>BAB IV IMPLEMENTASI DAN HASIL PENELITIAN</b> .....	30
4.1 Message Broker Service .....	30
4.2 File environment.....	35
4.3 Arsitektur 1 .....	36
4.3.1 Data Tracker Service .....	36
4.3.2 Notification Tracker Service.....	56
4.3.3 Notification Dispatcher Service.....	64
4.3.4 Telegram Service .....	72
4.3.5 Firebase Service .....	91
4.4 Arsitektur 2.....	105
4.4.1 Data Tracker Service .....	105

4.4.2	Telegram Service .....	107
4.4.3	Firebase Service .....	110
4.5	Load testing .....	113
4.5.1	Data Tracker Service .....	113
4.5.2	Notification Tracker Service .....	122
4.5.3	Notification Dispatcher Service .....	126
4.5.4	Telegram Service .....	130
4.5.5	Firebase Service .....	136
4.6	Perbandingan arsitektur microservice 1 dan arsitektur microservice 2 .....	142
BAB V KESIMPULAN DAN SARAN .....		144
5.1	Kesimpulan .....	144
5.2	Saran .....	144
DAFTAR PUSTAKA .....		145
LAMPIRAN A KODE PROGRAM .....		147
LAMPIRAN B KARTU KONSULTASI DOSEN 1 .....		168
LAMPIRAN C KARTU KONSULTASI DOSEN 2 .....		170
LAMPIRAN D FORM PERBAIKAN (REVISI) SKRIPSI .....		171



## DAFTAR TABEL

Tabel 4-1 fungsi-fungsi dalam Data Tracker service.....	37
Tabel 4-2 route yang tersedia pada Data Tracker service .....	37
Tabel 4-3 Checklist kebutuhan fungsional Data Tracker Service.....	56
Tabel 4-4 Route Notification Tracker .....	56
Tabel 4-5 Checklist kebutuhan fungsional Notification Tracker Service .....	64
Tabel 4-6 route yang dipakai pada service Notification Dispatcher .....	65
Tabel 4-7 Checklist kebutuhan fungsional Notification Dispatcher Service .....	72
Tabel 4-8 Routes yang dipakai pada Telegram service .....	73
Tabel 4-9 Struktur tabel member .....	75
Tabel 4-10 Checklist kebutuhan fungsional Telegram Service .....	91
Tabel 4-11 routes pada Firebase Service .....	92
Tabel 4-12 Checklist kebutuhan fungsional Firebase Service .....	104
Tabel 4-13 Checklist kebutuhan fungsional Firebase service arsitektur 2.....	107
Tabel 4-14 Checklist kebutuhan fungsional Telegram service arsitektur 2.....	110
Tabel 4-15 Checklist kebutuhan fungsional Firebase Service arsitektur 2.....	113
Tabel 4-16 Perbandingan waktu pengiriman notifikasi Telegram.....	142
Tabel 4-17 Perbandingan waktu pengiriman notifikasi email .....	143



## DAFTAR GAMBAR

Gambar 2.1 contoh implementasi message broker service diantara beberapa service/API.....	11
Gambar 2.2 contoh implementasi message broker service pada bidang usaha jual beli.....	11
Gambar 3.1 Arsitektur microservice pertama .....	14
Gambar 3.2 Arsitektur microservice kedua .....	14
Gambar 3.3 Diagram metode penelitian .....	16
Gambar 3.4 Desain Data Tracker service pada Portal Objek Budaya .....	23
Gambar 3.5 Desain message broker service pada arsitektur pertama Portal Objek Budaya.....	23
Gambar 3.6 Desain message broker service pada arsitektur kedua Portal Objek Budaya.....	24
Gambar 3.7 Desain service Notification Tracker pada arsitektur pertama Portal Objek Budaya.....	25
Gambar 3.8 Desain service Notification Dispatcher pada arsitektur pertama Portal Objek Budaya.....	25
Gambar 3.9 Desain service Telegram pada arsitektur pertama Portal Objek Budaya.....	26
Gambar 3.10 Desain service Telegram pada arsitektur kedua Portal Objek Budaya .....	26
Gambar 3.11 Desain Firebase service pada arsitektur pertama Portal Objek Budaya.....	27
Gambar 3.12 Desain Firebase service pada arsitektur kedua Portal Objek Budaya .....	27
Gambar 4.1 Halaman login RabbitMQ Management .....	31
Gambar 4.2 RabbitMQ Management Dashboard.....	31
Gambar 4.3 Pembuatan queue portal .....	32
Gambar 4.4 Hasil pembuatan queue portal .....	33

Gambar 4.5 Pembuatan queue telegram untuk arsitektur kedua.....	33
Gambar 4.6 Pembuatan queue firebase untuk arsitektur kedua .....	34
Gambar 4.7 Pembuatan exchange dengan tipe fanout .....	34
Gambar 4.8 Hasil binding exchange dengan kedua queue .....	35
Gambar 4.9 Triggers pada basis data MySQL .....	39
Gambar 4.10 Stored procedure yang diimplementasikan pada basis data .....	40
Gambar 4.11 Execution plan pada query tabel “test” .....	43
Gambar 4.12 Execution plan pada query tabel “kategori”.....	43
Gambar 4.13 Execution plan pada query tabel “subkategori” .....	43
Gambar 4.14 Execution plan pada query tabel “objek” .....	43
Gambar 4.15 Execution plan pada query tabel “docs” .....	43
Gambar 4.16 Execution plan pada query tabel “sections”.....	43
Gambar 4.17 Execution plan pada query tabel “sections_docs” .....	44
Gambar 4.18 Hasil REST Test Case GET pada route ‘/’ Data Tracker service ...	51
Gambar 4.19 Hasil REST Test Case POST pada route ‘/newDataInserted’ untuk event SPARQL.....	52
Gambar 4.20 Pesan update data SPARQL pada tabel kategori terkirim ke antrian ‘portal’.....	52
Gambar 4.21 Hasil REST Test Case POST pada route ‘/newDataInserted’ untuk event MySQL.....	53
Gambar 4.22 Pesan update data MySQL pada tabel kategori terkirim ke antrian ‘portal’.....	53
Gambar 4.23 Tambah data pada tabel test pada basis data Portal .....	54
Gambar 4.24 Pesan terkirim ke antrian ‘portal’ pada message broker service.....	54
Gambar 4.25 Hasil REST Test Case POST pada route ‘/newDataInserted’ untuk Failed Case 1 .....	55
Gambar 4.26 Hasil REST Test Case POST pada route ‘/newDataInserted’ untuk Failed Case 2 .....	55

Gambar 4.27 Hasil REST Test Cases GET pada route ‘/’ service Notification Tracker .....	60
Gambar 4.28 Response GET pada route ‘/’ service Notification Tracker .....	60
Gambar 4.29 Hasil REST Test Cases POST route ‘/callDispatcher’ service Notification Tracker MySQL.....	61
Gambar 4.30 Response POST pada route ‘/callDispathet’ service Notification Tracker MySQL .....	61
Gambar 4.31 Hasil REST Test Cases POST route ‘/callDispatcher’ service Notification Tracker SPARQL.....	62
Gambar 4.32 Response POST pada route ‘/callDispathet’ service Notification Tracker SPARQL .....	62
Gambar 4.33 Data ditambahkan pada tabel test.....	63
Gambar 4.34 Penerimaan message dari antrian message broker service.....	63
Gambar 4.35 Hasil failed REST Test Case “/callDispatcher” .....	64
Gambar 4.36 Response failed case route “/call dispatcher” .....	64
Gambar 4.37 Hasil REST Test Case Route “/” service Notification Dispatcher..	68
Gambar 4.38 Response route “/” service Notification Dispatcher.....	68
Gambar 4.39 Hasil REST Test Cases POST route ‘/dispatch’ service Notification Tracker data MySQL.....	69
Gambar 4.40 Hasil REST Test Cases POST route ‘/dispatch’ service Notification Tracker data SPARQL .....	70
Gambar 4.41 Response POST pada route ‘/dispatch’ service Notification Tracker data MySQL.....	70
Gambar 4.42 Response POST pada route ‘/dispatch’ service Notification Tracker data SPARQL.....	70
Gambar 4.43 Penambahan records pada tabel “test” .....	71
Gambar 4.44 pemanggilan newsletter services .....	71
Gambar 4.45 Hasil Failed REST Test Case route “/dispatch” .....	71
Gambar 4.46 Response failed case route “/dispatch” .....	72

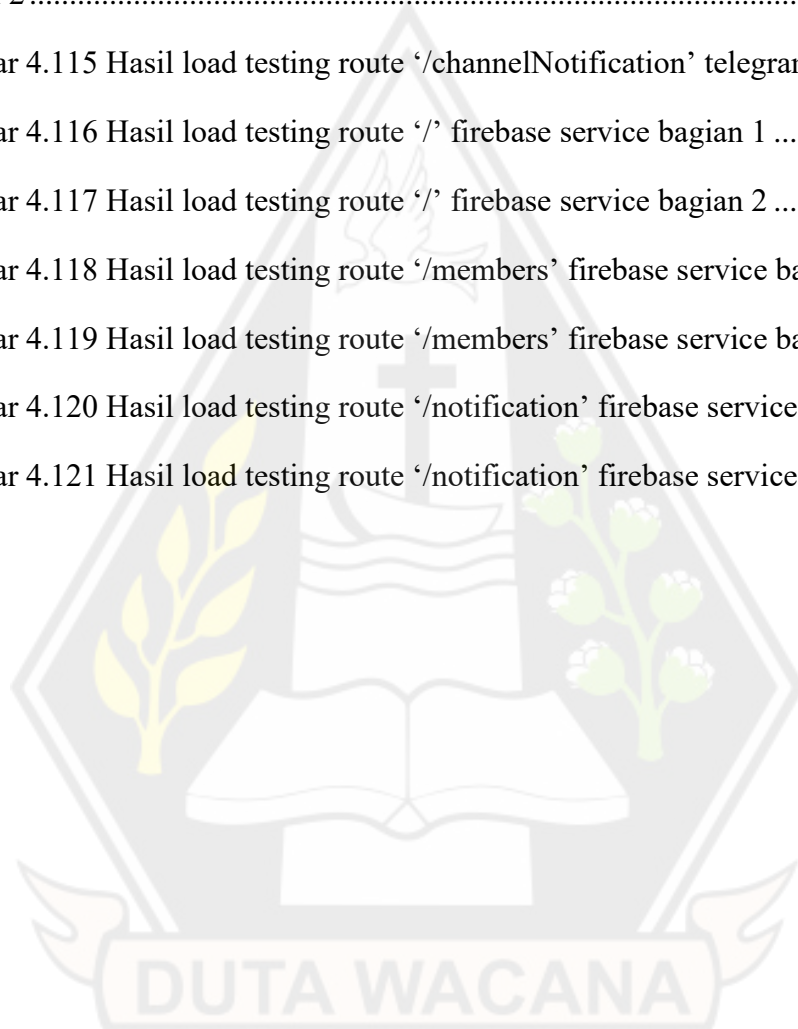
Gambar 4.47 Pembuatan struktur tabel pada halaman apache phpMyAdmin.....	76
Gambar 4.48 Struktur tabel members pada apache phpmuadmin.....	76
Gambar 4.49 Pembuatan Telegram Bot .....	79
Gambar 4.50 Hasil Rest Test Cases route “/” Telegram Service .....	82
Gambar 4.51 Response route “/” Telegram Service .....	83
Gambar 4.52 Hasil REST Test Case pada route “/members” .....	83
Gambar 4.53 Response yang dikirimkan route “/members” .....	84
Gambar 4.54 Hasil REST Test Cases POST route ‘/memberNotification’ service Telegram data MySQL.....	85
Gambar 4.55 Response POST pada route ‘/memberNotification’ service Telegram data MySQL .....	85
Gambar 4.56 Hasil REST Test Cases POST route ‘/memberNotification’ service Telegram data SPARQL .....	85
Gambar 4.57 Response POST pada route ‘/memberNotification’ service Telegram data SPARQL.....	86
Gambar 4.58 Notifikasi perubahan jumlah data yang diterima oleh member .....	86
Gambar 4.59 Hasil REST Test Cases POST route ‘/channelNotification’ service Telegram data SPARQL .....	87
Gambar 4.60 Hasil REST Test Cases POST route ‘/channelNotification’ service Telegram data mysql.....	87
Gambar 4.61 Response route ‘/channelNotification’ service Telegram .....	88
Gambar 4.62 Notifikasi perubahan jumlah data yang dikirimkan oleh Bot ke dalam channel.....	88
Gambar 4.63 Penambahan data pada tabel ‘test’ database ‘portal’ .....	88
Gambar 4.64 Notifikasi yang akan diterima oleh member .....	89
Gambar 4.65 Notifikasi yang akan dikirimkan ke channel.....	89
Gambar 4.66 Hasil Failed REST Test Case route “/channelNotification” .....	90
Gambar 4.67 Hasil Failed REST Test Case route “/memberNotification” .....	91

Gambar 4.68 Response untuk failed case .....	91
Gambar 4.69 Pembuatan Collection Firebase Firestore.....	94
Gambar 4.70 Penambahan data pertama pada Collection.....	95
Gambar 4.71 Hasil pembuatan collection dan penambahan data pertama collection .....	95
Gambar 4.72 Hasil Rest Test Cases route “/” Firebase Service.....	100
Gambar 4.73 Response route “/” Firebase Service .....	100
Gambar 4.74 Hasil Rest Test Cases route “/members” Firebase Service .....	101
Gambar 4.75 Response route “/members” Firebase Service .....	101
Gambar 4.76 Hasil REST Test Cases POST route “/notification” Firebase Service .....	102
Gambar 4.77 Response route “/notification” Firebase Service.....	102
Gambar 4.78 Email notifikasi yang diterima oleh member .....	102
Gambar 4.79 Penambahan data pada tabel “test” pada database “portal” .....	103
Gambar 4.80 Status pengiriman notifikasi email kepada member .....	103
Gambar 4.81 notifikasi email yang diterima member.....	103
Gambar 4.82 Hasil failed REST Test Case.....	104
Gambar 4.83 Response yang dikirimkan pada failed test case .....	104
Gambar 4.84 Penambahan data tabel test basis data MySQL.....	107
Gambar 4.85 Hasil pengiriman message ke dalam exchange .....	107
Gambar 4.86 Penambahan data tabel “test” basis data “portal” .....	109
Gambar 4.87 Penerimaan message dan pengiriman notifikasi Telegram .....	110
Gambar 4.88 Hasil notifikasi Telegram .....	110
Gambar 4.89 Penambahan data pada tabel “test” basis data “portal” MySQL... ..	112
Gambar 4.90 Penerimaan message pada firebase service .....	112
Gambar 4.91 Email yang diterima member .....	112

Gambar 4.92 Hasil load testing Data Tracker service arsitektur 1 bagian 1.....	114
Gambar 4.93 Hasil load testing Data Tracker service arsitektur 1 bagian 2.....	114
Gambar 4.94 Hasil load testing Data Tracker service arsitektur 1 bagian 3.....	115
Gambar 4.95 Hasil pengiriman 50 message ke dalam message broker service..	116
Gambar 4.96 Hasil load testing route ‘/’ data tracker service arsitektur 2 bagian 1 .....	117
Gambar 4.97 Hasil load testing route ‘/’ data tracker service arsitektur 2 bagian 2 .....	118
Gambar 4.98 Hasil load testing route ‘/newDataInserted’ data tracker service arsitektur 2 bagian 1 .....	119
Gambar 4.99 Hasil load testing route ‘/’ data tracker service arsitektur 2 bagian 2 .....	120
Gambar 4.100 Hasil pengiriman 50 message ke dalam exchange .....	121
Gambar 4.101 Hasil load testing route ‘/’ notification tracker service bagian 1	122
Gambar 4.102 Hasil load testing route ‘/’ notification tracker service bagian 2	123
Gambar 4.103 Hasil load testing route ‘/callDispatcher’ notification tracker service bagian 1 .....	124
Gambar 4.104 Hasil load testing route ‘/callDispatcher’ notification tracker service bagian 2.....	125
Gambar 4.105 Hasil load testing route ‘/’ notification dispatcher service bagian 1 .....	126
Gambar 4.106 Hasil load testing route ‘/’ notification dispatcher service bagian 2 .....	127
Gambar 4.107 Hasil load testing route ‘/dispatch’ notification dispatcher service bagian 1 .....	128
Gambar 4.108 Hasil load testing route ‘/dispatch’ notification dispatcher service bagian 2 .....	129
Gambar 4.109 Hasil load testing route ‘/’ telegram service bagian 1 .....	130
Gambar 4.110 Hasil load testing route ‘/’ telegram service bagian 2.....	131



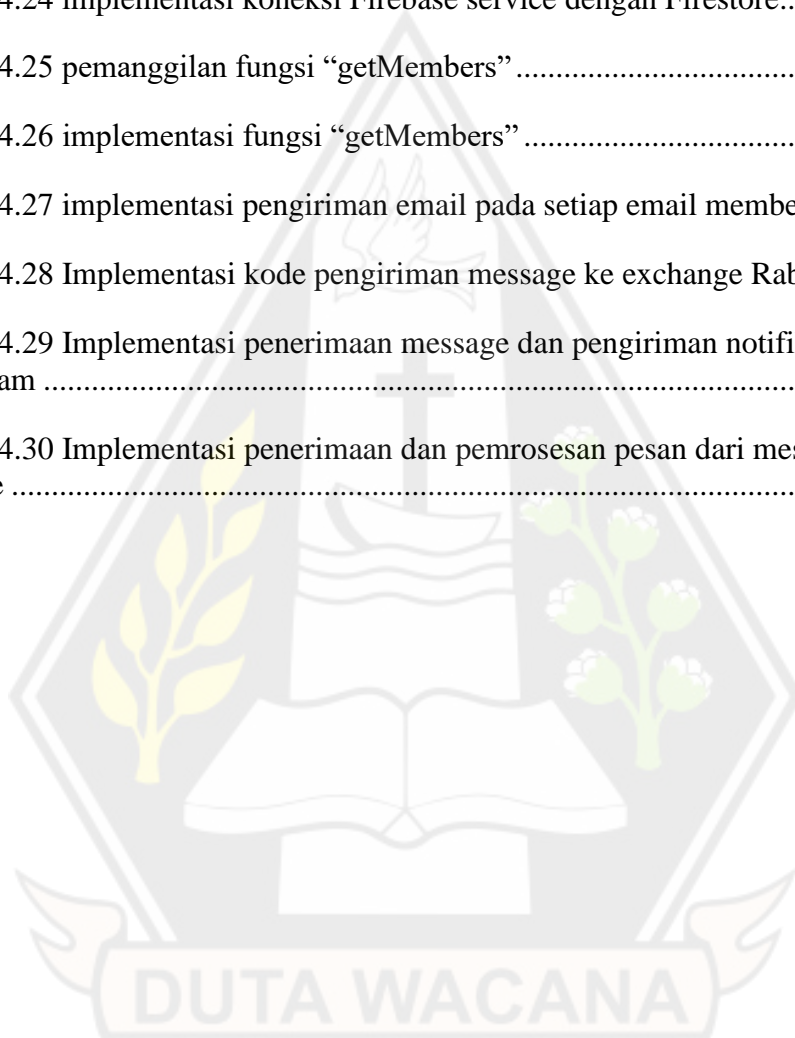
Gambar 4.111 Hasil load testing route ‘/members’ telegram service bagian 1 ..	132
Gambar 4.112 Hasil load testing route ‘/members’ telegram service bagian 2 ..	133
Gambar 4.113 Hasil load testing route ‘/memberNotification’ telegram service bagian 1 .....	134
Gambar 4.114 Hasil load testing route ‘/memberNotification’ telegram service bagian 2 .....	135
Gambar 4.115 Hasil load testing route ‘/channelNotification’ telegram service	136
Gambar 4.116 Hasil load testing route ‘/’ firebase service bagian 1 .....	136
Gambar 4.117 Hasil load testing route ‘/’ firebase service bagian 2 .....	137
Gambar 4.118 Hasil load testing route ‘/members’ firebase service bagian 1....	138
Gambar 4.119 Hasil load testing route ‘/members’ firebase service bagian 2....	139
Gambar 4.120 Hasil load testing route ‘/notification’ firebase service bagian 1	140
Gambar 4.121 Hasil load testing route ‘/notification’ firebase service bagian 2	141



## DAFTAR SCRIPT

Script 4.1 import dan pembacaan variabel environment.....	36
Script 4.2 file .env .....	36
Script 4.3 variabel environment yang digunakan pada Data Tracker service.....	36
Script 4.4 MySQL connection pada Data Tracker service.....	41
Script 4.5 pengambilan data dan pemanggilan fungsi pengecekan jumlah data MySQL.....	42
Script 4.6 fungsi pengecekan jumlah dan penyimpanan data MySQL Data Tracker service .....	45
Script 4.7 pengambilan data Fuseki dan pemanggilan fungsi setDataCount .....	48
Script 4.8 fungsi penyimpanan dan pengecekan jumlah data Fuseki pada Data Tracker service .....	49
Script 4.9 pengiriman pesan ke dalam antrian 'portal' pada message broker service .....	51
Script 4.10 file .env pada service Notification Tracker .....	57
Script 4.11 implementasi kode penerimaan dan pembacaan pesan dari antrian message broker service .....	59
Script 4.12 implementasi pemanggilan service Notification Dispatcher .....	59
Script 4.13 file .env pada Notification Dispatcher service.....	64
Script 4.14 pemanggilan fungsi callNotifications .....	67
Script 4.15 pemanggilan setiap newsletter service dan pengecekan status newsletter .....	68
Script 4.16 implementasi Routes yang digunakan pada Telegram Service dalam kode.....	74
Script 4.17 file .env yang digunakan pada service Telegram .....	74
Script 4.18 implementasi kode service dalam pembuatan koneksi dengan MySQL database .....	77
Script 4.19 pengambilan data member dari basis data dan pemanggilan fungsi setMembersData.....	78

Script 4.20 Penyimpanan hasil query ke variabel local pada service Telegram ...	79
Script 4.21 implementasi pengiriman notifikasi kepada personal chat member...	81
Script 4.22 implementasi pengiriman notifikasi pada telegram channel .....	82
Script 4.23 file .env pada Firebase service.....	94
Script 4.24 implementasi koneksi Firebase service dengan Firestore.....	96
Script 4.25 pemanggilan fungsi “getMembers” .....	97
Script 4.26 implementasi fungsi “getMembers” .....	98
Script 4.27 implementasi pengiriman email pada setiap email member.....	99
Script 4.28 Implementasi kode pengiriman message ke exchange RabbitMQ...	106
Script 4.29 Implementasi penerimaan message dan pengiriman notifikasi Telegram .....	109
Script 4.30 Implementasi penerimaan dan pemrosesan pesan dari message broker service .....	112



## INTISARI

Portal Objek Budaya merupakan sebuah *web* portal yang bertujuan sebagai tempat penyebaran informasi kebudayaan yang ada di Indonesia. Portal Objek Budaya akan menerapkan arsitektur *event-driven microservice* yang saat ini banyak diterapkan pada perusahaan-perusahaan di dunia. Penerapan arsitektur tersebut dapat dilakukan jika *service-service* yang ada dalam sistem Portal Objek Budaya dapat dipanggil atau dijalankan berdasarkan sebuah *trigger*. Untuk memecahkan masalah tersebut maka akan dilakukan penerapan sebuah design pattern yang melibatkan sebuah *Message Broker Service* dimana *Message Broker* tersebut digunakan sebagai koordinator dari pemanggilan fungsi-fungsi *service* yang ada pada Portal Objek Budaya.

Penerapan arsitektur *event-driven microservice* dengan konsep *choreography* pada Portal Objek Budaya dilakukan dengan pembuatan beberapa *service* kecil yang memiliki fungsi atau fokus tersendiri. Pembuatan setiap *service* tersebut akan dilakukan secara terpisah atau mandiri. *Message broker service* akan diterapkan di antara 2 *service* dimana fungsi dari *services* yang berperan sebagai konsumen akan dipanggil ketika ada pesan baru yang diterima dari dalam antrian yang ada pada *message broker service*. Pesan yang diterima oleh *consumer services* akan menjadi sebuah *trigger* dalam proses pengiriman notifikasi atau pemberitahuan adanya penambahan atau pengurangan data pada basis data yang dipakai kepada setiap member atau pengembang Portal Objek Budaya.

Dengan melakukan implementasi *message broker service* diantara *Data Tracker service* dan *service Notification Tracker* maka pemanggilan fungsi *service* guna penyampaian notifikasi kepada setiap *member* atau anggota pengembang dari Portal Objek Budaya berdasarkan sebuah *trigger* yaitu perubahan jumlah data dapat terpecahkan dan sistem *microservice* berhasil diterapkan pada sistem Portal Objek Budaya.

Kata kunci: *microservice*, *choreography*, *message broker*, *trigger*, *event-driven*, Portal Objek Budaya, *microservice architecture*.

## **ABSTRACT**

# **IMPLEMENTATION OF CHOREOGRAPHY-BASED EVENT-DRIVEN MICROSERVICE WITH MESSAGE BROKER SERVICE FOR PORTAL OBJEK BUDAYA**

By

Michael Pandu Nurseto

71180260

Portal Objek Budaya is a portal on the web that has the goal of becoming a means or a place for disseminating or spreading information or knowledge of Indonesian culture. Portal Objek Budaya will implement an event-driven microservice architecture that has been implemented by multiple companies in the world into their backend system. This implementation can be done only if services that will be implemented can be called by a trigger or an event that occurs in the system. To solve this problem a design pattern involving message broker service will be used in the implementation phase. In this design pattern, the message broker will be used as a manager that coordinates all of the service calls in Portal Objek Budaya's backend system.

Implementation of event-driven microservice architecture with choreography working concept for Portal Objek Budaya started with creating microservices with individual focus or function. The development of every service will be done individually. Message broker service will also be developed individually and used between 2 services that act as a producer and a consumer of the message sent to its queue. Services that acted as a consumer will run its function when a new message is received from the broker's queue. That message will become a trigger to send a notification by email and Telegram that there are addition or reduction of the amount of data used by Portal Objek Budaya to all of the active developers in Portal Objek Budaya's development team.

The result of this design pattern implementation is a success. With implementing a message broker service between two services, notification was able to be sent to all members by a trigger which in this case is an addition or reduction of the amount of data in the database can be solved.

Keywords: microservice, choreography, message broker, trigger, event-driven, Portal Objek Budaya, microservice architecture.





# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Indonesia merupakan negara yang mempunyai keanekaragaman dalam hal budaya yang sangat banyak. Pada zaman sekarang ini banyak orang yang kurang mengenal keanekaragaman budaya yang dimiliki oleh Indonesia, maka dari itu dibuatlah aplikasi web bernama Portal Objek Budaya guna menyebarkan informasi-informasi mengenai keanekaragaman budaya yang dimiliki Indonesia dalam berbagai aspek yang ada. Portal Objek Budaya akan menerapkan arsitektur *microservice* yang saat ini banyak diterapkan pada beberapa perusahaan besar di dunia dikarenakan mudahnya dilakukan pengembangan pada setiap *service* yang ada dan penanganan *error* yang terisolasi pada setiap *service*. Penerapan arsitektur tersebut dapat dilakukan jika pengembang mengerti bagaimana cara melakukan pemanggilan *web service* berdasarkan sebuah *trigger*.

Dalam teknologi *microservice* bisa dilakukan 2 pendekatan yang memungkinkan penerapan arsitektur tersebut. Pendekatan pertama adalah pembuatan sebuah *service* yang digunakan sebagai *gateway* guna memanggil fungsi dari *services* lain yang terdapat dalam aplikasi web. Pembuatan *service* yang berperan sebagai *gateway* ini masih memiliki kelemahan dalam pemanggilan fungsi *service* dalam skala yang cukup besar. Pendekatan kedua merupakan penerapan sebuah design pattern yang melibatkan *message broker service* sebagai koordinator dari pemanggilan fungsi *services* yang ada dalam sistem Portal Objek Budaya.

Penelitian ini merupakan bagian dari Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) yang mendapat pendanaan dari RISTEK-BRIN dengan judul “Portal Objek Budaya Berbasis Semantic Web”. Dalam penelitian ini, peneliti akan melakukan penerapan arsitektur *event-driven microservice* pada aplikasi web Portal Objek budaya dengan menerapkan pendekatan kedua yang berfokus pada

peran *message broker service* sebagai koordinator pemanggilan fungsi *services* dengan menggunakan sebuah *event* yang akan di *publish* ke dalam *message broker service* sebagai *trigger* pemanggilan fungsi *service* dalam sistem Portal Objek Budaya.

## 1.2 Rumusan Masalah

Masalah yang dapat dirumuskan dalam penelitian ini adalah bagaimana cara pemenuhan kebutuhan sistem Portal Objek Budaya sehingga memungkinkan pemanggilan fungsi *services* pada sistem guna memberi sebuah pemberitahuan atau notifikasi berdasarkan sebuah *trigger* yang dihasilkan oleh sistem ketika terjadi sebuah perubahan jumlah data atau informasi pada basis data Portal Objek Budaya secara terkoordinasi. Dalam penelitian ini peneliti akan berusaha menyelesaikan masalah tersebut dengan menerapkan arsitektur *event-driven microservice* bersamaan dengan pemanfaatan *Message Broker Service* guna melakukan koordinasi dalam pemanggilan fungsi pada *services* yang ada ketika sebuah *event* dihasilkan oleh sistem.

## 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini yaitu:

1. Layanan *Message Broker* yang digunakan merupakan *Message Broker* yang memiliki layanan *free tier*.
2. Semua data yang digunakan merupakan data milik Portal Objek Budaya.
3. Data ontologi/Fuseki yang dipakai dalam penelitian ini adalah data Candi, Keris, dan Makanan.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini yakni menerapkan konsep koreografi dalam arsitektur *event-driven microservice* dengan penggunaan *message broker service* dalam pengoordinasian pemanggilan fungsi *services* yang digunakan dalam sistem Portal Objek Budaya.

## 1.5 Manfaat Penelitian

Manfaat yang bisa didapatkan dari penelitian ini merupakan peningkatan kemampuan sistem *backend* Portal Objek Budaya dalam aspek pelaksanaan fungsi-fungsi secara konsisten (*reliability*) dan ketersediaan service (*availability*). Selain kedua hal tersebut manfaat lain yang didapatkan dari penelitian ini yaitu mempermudah dilakukannya peningkatan atau pengembangan skala (*scalability*) pada backend Portal Objek Budaya oleh pengembang di kemudian hari.

## 1.6 Metodologi Penelitian

### 1.6.1 Metode Penelitian

Metode penelitian yang akan digunakan dalam penelitian ini yaitu:

1. Studi Literatur

Studi pustaka dari berbagai sumber yang tersedia tersedia seperti buku, jurnal, artikel, dan sumber-sumber lain yang mengandung informasi yang berkaitan dengan penelitian ini.

2. Analisis Kebutuhan *Service*

Analisis mengenai kebutuhan *service* yang akan diimplementasikan, algoritma yang akan digunakan serta *libraries* yang mungkin dipakai dalam tahap implementasi kode *service*.

### 3. Desain *Service*

Pembuatan desain *service* yang mencakup bagaimana *service* dapat berkomunikasi dengan *service* lain yang terdapat dalam sistem Portal Objek Budaya.

### 4. Implementasi Kode *Service*

Implementasi desain *service* yang sudah dibuat ke dalam kode program sehingga *service* yang sudah didesain dapat terimplementasikan dengan baik.

### 5. Evaluasi *Service*

Evaluasi *services* yang telah diimplementasikan menggunakan REST Test Cases guna mendapatkan data mengenai koordinasi semua *service* beserta dengan *message broker service* yang telah diimplementasikan ke dalam sistem Portal Objek Budaya.

## 1.7 Sistematika Penulisan

Penelitian ini ditulis dalam lima bab besar yaitu:

Bab 1 Pendahuluan. Bab ini memuat latar belakang mengapa penelitian ini dilakukan, rumusan masalah yang ditimbulkan dari latar belakang yang ada, batasan peneliti mengenai masalah yang telah dirumuskan, tujuan dari penelitian ini, metodologi penelitian yang digunakan dalam pelaksanaan penelitian, serta sistematika penulisan dari penelitian ini.

Bab 2 Tinjauan Pustaka dan Landasan Teori. Bab ini memuat informasi-informasi dan teori-teori yang digunakan peneliti sebagai dasar penelitian dan literatur pendukung dalam pelaksanaan penelitian guna memecahkan masalah yang ada.

Bab 3 Metode Penelitian. Bab ini memuat penjelasan secara rinci mengenai semua metode yang digunakan dalam penelitian ini dari langkah awal hingga akhir.

Bab 4 Implementasi dan Hasil Penelitian. Bab ini memuat proses serta hasil dari implementasi arsitektur *event-driven microservice* dengan *message broker service* pada Portal Objek Budaya.

Bab 5 Kesimpulan dan Saran. Bab ini berisi kesimpulan yang diambil dari hasil implementasi arsitektur *event-driven microservice* dengan *message broker service* pada Portal Objek budaya serta saran peneliti mengenai hal-hal yang bisa diterapkan atau diimplementasikan pada penelitian yang akan datang berdasarkan hasil dari penelitian ini.



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan penelitian ini, dapat disimpulkan bahwa penerapan *choreography-based event-driven microservice* dengan bantuan *message broker service* berhasil diterapkan pada sistem Portal Objek Budaya. Perbandingan kedua arsitektur *microservice* yang dibuat juga memberikan bukti dari arsitektur mana yang lebih baik dalam hal efisiensi waktu. Dengan menempatkan *message broker service* di antara *Data Tracker service* dan *Notification Tracker service*, masalah pemanggilan fungsi-fungsi yang digunakan untuk penyampaian notifikasi kepada anggota Portal Objek Budaya ketika ditemukan sebuah data baru pada basis data yang digunakan oleh Portal Objek Budaya dapat terpecahkan.

#### **5.2 Saran**

Saran yang didapatkan dari penelitian ini untuk dilakukan pada penelitian-penelitian selanjutnya yaitu:

1. Pelaksanaan *load testing* lebih mendalam untuk mendapatkan informasi mengenai batas performa pada setiap *service* yang telah dibuat.
2. Menambahkan koneksi antara *Data Tracker service* dengan data-data ontologi selain ontologi Candi, Keris, dan Makanan yang dipakai oleh Portal Objek Budaya sehingga data yang digunakan sebagai *trigger* pengiriman notifikasi lebih banyak.
3. Menambahkan fitur *logging* sehingga setiap kali ada sebuah aktifitas yang berjalan baik *error* ataupun hasil *console.log* setiap *service* dapat dicatat sebagai *history* pengembangan setiap *service*.



## DAFTAR PUSTAKA

- Etzkorn, L. H. (2017). *Introduction to Middleware: Web Services, Object Components, and Cloud Computing (1st Ed)*. Chapman and Hall/CRC.
- Fauzi, A., Harli, E., & Haryanto, Y. (2021). Implementasi Arsitektur Event-Driven dan Microservices untuk Mesin Vending. *Jurnal Format Volume 10 Nomor 2*.
- Febriyanti, F., Pramukantoro, E. S., & Bakhtiar, F. A. (2019). Perbandingan Kinerja Redis, Mosquitto, dan MongoDB sebagai Message Broker pada IoT Middleware. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- IBM. (2020, Januari 23). *What are Message Brokers? | IBM*. Retrieved from IBM: <https://www.ibm.com/cloud/learn/message-brokers>
- Jin, B., Sahni, S., & Shevat, A. (2018). *Designing Web APIs: Building APIs That Developers Love*. O'Reilly Media.
- Korab, J. (2017). *Understanding Message Brokers: Learn the Mechanics of Messaging through ActiveMQ and Kafka*. O'Reilly Media.
- Macero, M. (2017). *Learn Microservices with Spring Boot: A Practical Approach to RESTful Services using RabbitMQ, Eureka, Ribbon, Zuul and Cucumber (1st Ed)*. Apress.
- Munawar, G., & Hodijah, A. (2018). Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK. *Jurnal & Penelitian Teknik Informatika*.
- Nadareishvili, I., Mitra, R., McLarty, M., & Amundsen, M. (2016). *Microservice Architecture: Aligning Principles, Practices, and Culture*. O'Reilly Media.

Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.

Sakti, C. S., & Hermawan, I. (2020). Implementasi Arsitektur Microservice Pada Back End Sistem Informasi Atlantas Berbasis Website. *Jurnal Teknologi Terpadu*.

