

**PEMANFAATAN API GATEWAY PADA ARSITEKTUR
LAYANAN MIKRO**

Skripsi



oleh

Nathaniel Alvin Pratama

71170157

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS KRISTEN DUTA WACANA

2021

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 : Nathaniel Alvin Pratama
NIM : 71170157
Program studi : Informatika
Fakultas : Teknologi Informasi
Jenis Karya : Skripsi

demikian 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:

**“PEMANFAATAN API GATEWAY PADA ARSITEKTUR LAYANAN
MIKRO”**

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 November 2021

Yang menyatakan



(Nathaniel Alvin Pratama)

NIM 71170157

**PEMANFAATAN API GATEWAY PADA ARSITEKTUR
LAYANAN MIKRO**

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi

Universitas Kristen Duta Wacana

Sebagai Salah Satu Syarat dalam memperoleh Gelar

Sarjana Komputer

Disusun oleh:

Nathaniel Alvin Pratama

71170157

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS KRISTEN DUTA WACANA

2021

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul

PEMANFAATAN API GATEWAY PADA ARSITEKTUR LAYANAN MIKRO

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 manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

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

15 November
Yogyakarta 2021

vin Pratama

71170157

HALAMAN PERSETUJUAN

Judul Skripsi : PEMANFAATAN API GATEWAY PADA
ARSITEKTUR LAYANAN MIKRO

Nama Mahasiswa : Nathaniel Alvin Pratama

NIM : 71170157

Mata Kuliah : Skripsi (Tugas Akhir)

Kode : TI0366

Semester : Gasal

Tahun Akademik : 2021/2022


Telah diperiksa dan disetujui di Yogyakarta

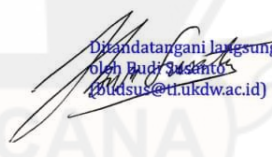
Pada Tanggal

5 Oktober 2021

Dosen Pembimbing I

Dosen Pembimbing II


Danny Sebastian, S.Kom., M.M., M.T


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

cn=Budi Susanto,
o=Universitas Kristen Duta
Wacana, ou=Fakultas
Teknologi Informasi,
email=budsus@ti.ukdw.ac.i
d, c=ID

2021.10.07 10:09:33 +07'00'

HALAMAN PENGESAHAN

PEMANFAATAN API GATEWAY PADA ARSITEKTUR LAYANAN MIKRO

Oleh: NATHANIEL ALVIN PRATAMA / 71170157

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 19 Oktober 2021

Yogyakarta, 5 November 2021
Mengesahkan,

Dewan Penguji:

1. Danny Sebastian, S.Kom., M.M., M.T.
2. Budi Susanto, SKom.,M.T.
3. Lukas Chrisantyo, S.Kom., M.Eng.
4. Willy Sudiarto Raharjo, S.Kom.,M.Cs.



Handwritten signatures of the exam board members, including a stamp with the number 110722401.



Dekan

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

Ketua Program Studi

(Gloria Virginia, Ph.D.)

UCAPAN TERIMAKASIH

Selama proses penulisan laporan skripsi ini, penulis mendapat banyak bantuan dari berbagai pihak. Bantuan yang penulis dapatkan berupa ilmu, fisik, non-fisik maupun bantuan dalam bentuk lainnya. Di dalam bagian ini, penulis ingin berterima kasih kepada pihak-pihak di bawah ini:

1. Tuhan Yesus Kristus, atas berkat dan Anugrah yang Ia limpahkan selama kehidupan penulis dan dari awal perkuliahan hingga mengerjakan skripsi ini. Tanpa berkatnya, penulis tidak mampu menyelesaikan laporan skripsi.
2. Kepada Ayah dan Ibu penulis yang telah membimbing penulis selama masa hidup penulis dan memberikan kesempatan kepada penulis untuk memiliki jenjang edukasi yang tinggi.
3. Kepada dosen pembimbing penulis, Bapak Danny Sebastian, S.Kom., M.M., M.T dan Bapak Budi Susanto, S.Kom.,M.T., karena telah membimbing dan mengajarkan banyak ilmu kepada penulis selama masa pengerjaan laporan skripsi ini.
4. Kepada Bapak Antonius Rachmat, S.Kom., M.Cs., selaku koordinator skripsi, yang telah membantu dan memberi saran kepada penulis selama masa pengerjaan laporan akhir penulis.
5. Kepada semua dosen Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, karena telah berbagi ilmu kepada penulis selama masa perkuliahan yang penulis lalui.
6. Kepada teman sekelompok penulis dalam mengembangkan sistem, Michael William Alexander, yang telah membagi ilmunya dalam pengembangan layanan dan mengarahkan penulis selama membangun sistem.
7. Kepada sahabat penulis semasa kuliah, Nicholas Christianto, Ananda Kusumawardana, Michael William Alexander, Cynthia Kumalasari, yang selalu berbagi tawa dan sedih dengan penulis semasa kuliah.
8. Kepada teman-teman program studi informatika 2017, yang sering membantu, menemani dan berbagi tawa dengan penulis selama perkuliahan.

9. Kepada keluarga besar yang selalu mendukung dan mendoakan penulis semasa hidup penulis.

Atas semua dukungan yang penulis dapat, penulis hanya bisa berterima kasih kepada pihak-pihak di atas. Penulis berharap skripsi ini dapat berguna baik untuk penulis maupun pihak lain yang membaca laporan ini.



INTISARI

PEMANFAATAN API GATEWAY PADA ARSITEKTUR LAYANAN MIKRO

Dalam proses pengembangan sebuah sistem, akan terdapat banyak masalah ketika terjadi perubahan. Masalah yang muncul seringkali terjadi pada arsitektur monolitik, dimana kode program memiliki ketergantungan satu dengan lainnya. Dengan arsitektur tersebut, semakin besar sistem yang dibuat, akan membuat semakin banyak masalah dalam proses perubahan. Hal ini dapat membuat sistem yang dibuat menjadi penuh dengan *bug*.

Untuk mengatasi masalah tersebut, sistem dapat menerapkan arsitektur layanan mikro. Penerapan arsitektur ini dapat membantu sistem mengurangi *coupling* yang ketat. Dalam studi kasus penggunaan arsitektur ini, arsitektur ini digunakan dalam aplikasi toko buku berbasis seluler. Untuk membantu penggunaan arsitektur ini, *API gateway* juga digunakan untuk mengamankan layanan dan mengatur *request* kepada layanan. Untuk menentukan layanan sesuai dengan *requirement* dari aplikasi, layanan akan dites dengan *unit testing* yang dibuat sebelum pengembangan sistem.

Berdasarkan analisis sistem yang telah dibuat, penerapan arsitektur layanan mikro sesuai dengan karakteristik layanan mikro. Layanan yang dibuat bersifat modular, dengan basis data yang tidak terpusat, dan dirancang dengan fokus pada produk. *API gateway* membantu menjadi *entrypoint* dari tiap layanan, dengan menerapkan *API key* untuk keamanan layanan dan *usage plan* untuk membatasi *request*. Dengan penerapan tersebut, layanan menjadi aman dari *client* yang tidak memiliki *key* dan terhindar dari *request overload*. Untuk pengembangan sistem serupa, disarankan penggunaan *lambda function* dengan OAuth untuk menjadi *authorizer* dari *API gateway*, sehingga keamanan semakin lebih baik.

Kata kunci: Layanan Mikro, Monolitik, API Gateway.

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PENGESAHAN.....	v
UCAPAN TERIMAKASIH.....	vi
INTISARI	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN	xvii
BAB 1 Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Penelitian	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Metode Penelitian	3
1.7. Sistematika Penulisan	6
BAB 2 Tinjauan Pustaka.....	8
2.1. Tinjauan Pustaka	8
2.2. Landasan Teori	9
2.2.1. <i>Monolithic Architecture</i>	9
2.2.2. <i>Microservices Architecture</i>	11
2.2.3. <i>Refactoring Arsitektur Monolitik menuju Layanan Mikro</i>	13
2.2.4. <i>API Gateway</i>	15
BAB 3 Metodologi Penelitian	16
3.1. Alur Penelitian	16
3.2. Blok Diagram Sistem	18
3.3. <i>Requirement Aplikasi Toko Buku</i>	18
3.4. <i>Spesifikasi Aplikasi Toko Buku</i>	19
3.5. <i>Business Process Model</i>	19
3.6. <i>Domain Driven Design</i>	33

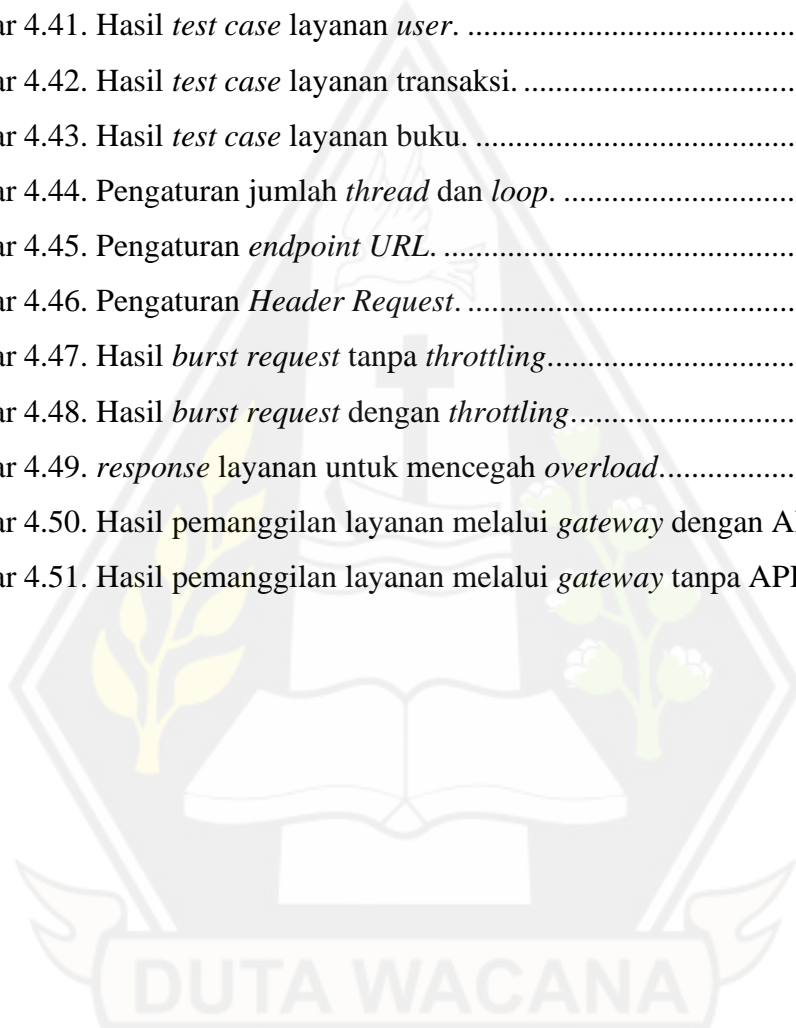
3.7.	Desain Basis Data	33
3.8.	Metode Pengujian.....	36
3.9.	<i>Sequence Diagram</i>	36
3.10.	<i>Request dan Response Layanan</i>	37
3.10.1.	<i>Buku Service</i>	38
3.10.2.	<i>Review Service</i>	48
3.10.3.	<i>Wishlist Service</i>	51
3.10.4.	<i>User Service</i>	53
3.10.5.	<i>Transaksi Service</i>	55
BAB 4	Implementasi dan Analisis Sistem	64
4.1.	Implementasi Sistem.....	64
4.1.1.	Layanan Mikro	64
4.1.2.	<i>Service deployment</i>	74
4.1.3.	<i>API Gateway</i>	78
4.1.4.	Aplikasi Android.....	83
4.2.	Analisis Sistem	87
4.2.1.	Pengujian Sistem dengan <i>Functional Testing</i>	87
4.2.2.	Pengujian <i>Usage Plan</i> dengan JMeter	90
4.2.3.	Pengujian <i>API Gateway</i> dengan <i>Postman</i>	92
4.2.4.	Penguraian Arsitektur Sistem	94
BAB 5	Kesimpulan dan Saran	95
5.1.	Kesimpulan.....	95
5.2.	Saran	95
Daftar Pustaka	96
Lampiran	98
<i>Source Code Service Buku</i>	98	
<i>Source Code Service Review</i>	124	
<i>Source Code Service Wishlist</i>	129	
<i>Source Code Service User</i>	134	
<i>Source Code Service Transaksi</i>	138	
Skenario Test layanan	162	

DAFTAR GAMBAR

Gambar 2.1. Contoh pattern dari Monolithic Application	10
Gambar 2.2. Contoh alur aplikasi dengan arsitektur Layanan Mikro	12
Gambar 2.3. Perbandingan hasil sebelum dan sesudah refactoring	15
Gambar 3.1. Flowchart proses penelitian.....	16
Gambar 3.2. Blok diagram sistem yang dibangun.	18
Gambar 3.3. proses bisnis penambahan buku.	20
Gambar 3.4. Proses bisnis menghapus data buku oleh <i>administrator</i>	21
Gambar 3.5. Proses bisnis mengubah data buku.....	22
Gambar 3.6. Proses bisnis <i>request</i> data detail buku pada saat memilih buku.....	23
Gambar 3.7. proses bisnis <i>request list</i> buku yang ada	23
Gambar 3.8. proses bisnis <i>user</i> menambahkan buku ke dalam keranjang.....	24
Gambar 3.9. Proses bisnis ketika <i>user</i> mengurangi barang pada keranjang.	25
Gambar 3.10. proses bisnis pembelian buku.....	26
Gambar 3.11. proses bisnis ketika <i>user</i> menambahkan buku ke dalam <i>wishlist</i>	27
Gambar 3.12. proses bisnis ketika <i>user</i> menghapus buku dari <i>wishlist</i>	28
Gambar 3.13. proses bisnis pencarian buku dalam aplikasi.....	29
Gambar 3.14. proses bisnis <i>login</i> dan pengecekan akun.....	30
Gambar 3.15. proses bisnis ketika <i>user</i> ingin melihat profilnya.	31
Gambar 3.16. proses bisnis mengubah profil <i>user</i>	31
Gambar 3.17. proses bisnis penggunaan aplikasi secara keseluruhan	32
Gambar 3.18. <i>Bounded context</i> aplikasi toko buku.....	33
Gambar 3.19. Desain basis data layanan buku.....	34
Gambar 3.20. Desain basis data layanan <i>User</i>	34
Gambar 3.21. Desain basis data layanan <i>wishlist</i>	34
Gambar 3.22. Desain basis data layanan <i>review</i>	35
Gambar 3.23. Desain basis data layanan transaksi	35
Gambar 3.24. <i>Sequence diagram</i> alur proses aplikasi.	37
Gambar 4.1. Query pembuatan tabel buku.....	65
Gambar 4.2. <i>Query</i> pembuatan tabel genre.....	65

Gambar 4.3. <i>Query</i> pembuatan tabel penerbit	65
Gambar 4.4. <i>Query</i> pembuatan tabel penulis	65
Gambar 4.5. <i>Query</i> pembuatan tabel list_genre.....	65
Gambar 4.6. <i>Query</i> pembuatan tabel list_penulis	66
Gambar 4.7. <i>Model class</i> entitas buku	67
Gambar 4.8. <i>Interface</i> DAO.....	68
Gambar 4.9. <i>Interface</i> BukuDao.....	68
Gambar 4.10. <i>Repository Class</i> BukuRepository.....	68
Gambar 4.11. <i>Class</i> BukuService.....	70
Gambar 4.12. Contoh <i>class response</i> dalam layanan Buku.....	71
Gambar 4.13. <i>class</i> <i>BadRequestException</i>	72
Gambar 4.14. <i>class</i> <i>NotFoundException</i>	72
Gambar 4.15. <i>Controller class</i> untuk entitas buku.....	73
Gambar 4.16. atribut <i>packaging</i> berisi war.....	74
Gambar 4.17. <i>command</i> untuk membuat <i>package</i> dari <i>project</i>	74
Gambar 4.18. halaman pembuatan aplikasi baru pada <i>AWS elastic beanstalk</i>	75
Gambar 4.19. konfigurasi aplikasi baru.....	76
Gambar 4.20. proses unggah <i>package</i>	77
Gambar 4.21. <i>Environment variable</i> untuk layanan buku.....	77
Gambar 4.22. <i>Application Health</i> OK.....	78
Gambar 4.23. jenis-jenis API yang ditawarkan AWS.....	78
Gambar 4.24. konfigurasi awal <i>API Gateway</i>	79
Gambar 4.25. <i>List</i> API yang terbuat.....	80
Gambar 4.26. Proses pembuatan <i>resource</i> baru.....	80
Gambar 4.27. Proses pembuatan <i>method</i> baru.....	80
Gambar 4.28. hasil pembuatan <i>method</i> baru pada <i>resource</i> buku.....	80
Gambar 4.29. menambahkan <i>request parameter</i> dalam <i>endpoint</i> /buku.....	81
Gambar 4.30. menambahkan <i>path variable</i> dalam <i>endpoint</i> /buku.....	81
Gambar 4.31. proses konfigurasi <i>path</i>	82
Gambar 4.32. proses pembuatan <i>API key</i>	83
Gambar 4.33. konfigurasi <i>usage plan</i>	83

Gambar 4.34. pemasangan <i>usage plan</i> dengan <i>API key</i>	83
Gambar 4.35. <i>abstract class</i> <i>WishlistHandlers</i>	84
Gambar 4.36. <i>command</i> untuk melakukan <i>build</i>	85
Gambar 4.37. hasil <i>class</i> dari <i>RetroFit</i>	85
Gambar 4.38. Contoh pemanggilan <i>API</i>	86
Gambar 4.39. Hasil <i>test case</i> layanan <i>Review</i>	87
Gambar 4.40. Hasil <i>test case</i> layanan <i>wishlist</i>	88
Gambar 4.41. Hasil <i>test case</i> layanan <i>user</i>	89
Gambar 4.42. Hasil <i>test case</i> layanan transaksi.	89
Gambar 4.43. Hasil <i>test case</i> layanan buku.	90
Gambar 4.44. Pengaturan jumlah <i>thread</i> dan <i>loop</i>	91
Gambar 4.45. Pengaturan <i>endpoint URL</i>	91
Gambar 4.46. Pengaturan <i>Header Request</i>	91
Gambar 4.47. Hasil <i>burst request</i> tanpa <i>throttling</i>	91
Gambar 4.48. Hasil <i>burst request</i> dengan <i>throttling</i>	92
Gambar 4.49. <i>response</i> layanan untuk mencegah <i>overload</i>	92
Gambar 4.50. Hasil pemanggilan layanan melalui <i>gateway</i> dengan <i>API key</i>	93
Gambar 4.51. Hasil pemanggilan layanan melalui <i>gateway</i> tanpa <i>API key</i>	93



DAFTAR TABEL

Tabel 3.1. <i>request</i> dan <i>response body</i> mengambil semua data buku.	38
Tabel 3.2. <i>request</i> dan <i>response body</i> mengambil data detail buku.	38
Tabel 3.3. <i>request</i> dan <i>response body</i> mengambil data buku berdasarkan genre.	39
Tabel 3.4. <i>request</i> dan <i>response body</i> mengambil data buku lain dengan genre sama.....	40
Tabel 3.5. <i>request</i> dan <i>response body</i> menyimpan data buku baru	41
Tabel 3.6. <i>request</i> dan <i>response body</i> mengubah data buku.....	41
Tabel 3.7. <i>request parameter</i> dan <i>response body</i> menghapus data buku.....	42
Tabel 3.8. <i>request</i> dan <i>response body</i> mengambil semua genre.....	43
Tabel 3.9. <i>request</i> dan <i>response body</i> menambah genre baru.....	43
Tabel 3.10. <i>request</i> dan <i>response body</i> mengubah genre.....	43
Tabel 3.11. <i>request parameter</i> dan <i>response body</i> menghapus genre.	44
Tabel 3.12. <i>request</i> dan <i>response body</i> mengambil data penerbit.	44
Tabel 3.13. <i>request</i> dan <i>response body</i> menambah data penerbit.....	45
Tabel 3.14. <i>request</i> dan <i>response body</i> mengubah data penerbit.....	45
Tabel 3.15. <i>request parameter</i> dan <i>response body</i> menghapus data penerbit.	46
Tabel 3.16. <i>request</i> dan <i>response body</i> mengambil semua data penulis.....	46
Tabel 3.17. <i>request</i> dan <i>response body</i> mengambil data buku dengan penulis tertentu.....	47
Tabel 3.18. <i>request</i> dan <i>response body</i> menambah data penulis baru	47
Tabel 3.19. <i>request</i> dan <i>response body</i> mengubah data penulis.	48
Tabel 3.20. <i>request parameter</i> dan <i>response body</i> menghapus data penulis.	48
Tabel 3.21. <i>request</i> dan <i>response body</i> mengambil detail <i>review</i> dengan ID.....	49
Tabel 3.22. <i>request</i> dan <i>response body</i> mengambil data <i>review</i> dari sebuah buku.	49
Tabel 3.23. <i>request</i> dan <i>response body</i> mengambil data <i>review</i> dari <i>user</i> tertentu.	50
Tabel 3.24. <i>request</i> dan <i>response body</i> membuat <i>review</i> baru.	51
Tabel 3.25. <i>request</i> dan <i>response body</i> mengambil data <i>wishlist</i> dari <i>user</i>	52

Tabel 3.26. <i>request</i> dan <i>response body</i> menambah <i>wishlist</i>	52
Tabel 3.27. <i>request</i> dan <i>response body</i> menghapus buku dari <i>wishlist</i>	53
Tabel 3.28. <i>request</i> dan <i>response body</i> mengambil profil <i>user</i>	53
Tabel 3.29. <i>request</i> dan <i>response body</i> proses <i>sign up</i>	54
Tabel 3.30. <i>request</i> dan <i>response body</i> proses <i>sign in</i>	54
Tabel 3.31. <i>request</i> dan <i>response body</i> melakukan perubahan profil <i>user</i>	54
Tabel 3.32. <i>request</i> dan <i>response body</i> mengambil data isi keranjang dari <i>user</i>	55
Tabel 3.33. <i>request</i> dan <i>response body</i> menambah buku ke dalam keranjang	56
Tabel 3.34. <i>request</i> dan <i>response body</i> menghapus semua isi keranjang	56
Tabel 3.35. <i>request</i> dan <i>response body</i> menghapus sebuah buku dari keranjang..56	
Tabel 3.36. <i>request</i> dan <i>response body</i> menambah atau mengurangi jumlah buku	57
Tabel 3.37. <i>request</i> dan <i>response body</i> mengambil semua data order <i>user</i>	57
Tabel 3.38. <i>request</i> dan <i>response body</i> mengambil detail order.	58
Tabel 3.39. <i>request</i> dan <i>response body</i> mengambil semua data order dengan status <i>unfinished</i>	59
Tabel 3.40. <i>request</i> dan <i>response body</i> membuat order baru.	59
Tabel 3.41. <i>request</i> dan <i>response body</i> mengubah status order menjadi <i>finished</i> . 60	
Tabel 3.42. <i>request</i> dan <i>response body</i> mengambil alamat-alamat user.	60
Tabel 3.43. <i>request</i> dan <i>response body</i> mengambil detail alamat.	61
Tabel 3.44. <i>request</i> dan <i>response body</i> menambah alamat baru.	62
Tabel 3.45. <i>request</i> dan <i>response body</i> menghapus alamat.	62
Tabel 3.46. <i>request</i> dan <i>response body</i> mengubah alamat.	63

DAFTAR LAMPIRAN

- Lampiran 1. Buku model
- Lampiran 2. Genre model
- Lampiran 3. Penerbit model
- Lampiran 4. Penulis model
- Lampiran 5. ListPenulisBuku model
- Lampiran 6. ListGenreBuku model
- Lampiran 7. DAO *Interface*
- Lampiran 8. BukuDAO *Interface*
- Lampiran 9. GenreDAO *Interface*
- Lampiran 10. PenerbitDAO *Interface*
- Lampiran 11. PenulisDAO *Interface*
- Lampiran 12. ListPenulisDAO *Interface*
- Lampiran 13. ListGenreBukuDAO *Interface*
- Lampiran 14. Buku *Repository*
- Lampiran 15. Genre *Repository*
- Lampiran 16. Penerbit *Repository*
- Lampiran 17. Penulis *Repository*
- Lampiran 18. ListGenreBuku *Repository*
- Lampiran 19. ListPenulisBuku *Repository*
- Lampiran 20. Buku *Service*
- Lampiran 21. Genre *Service*
- Lampiran 22. Penulis *Service*
- Lampiran 23. Penerbit *Service*
- Lampiran 24. Buku *Controller*
- Lampiran 25. Genre *Controller*
- Lampiran 26. Penerbit *Controller*
- Lampiran 27. Penulis *Controller*
- Lampiran 28. *Review* model
- Lampiran 29. DAO *interface*
- Lampiran 30. ReviewDAO *interface*

Lampiran 31. *Review interface*
Lampiran 32. *Review Repository*
Lampiran 33. *Review Service*
Lampiran 34. *Review Controller*
Lampiran 35. *Wishlist model*
Lampiran 36. *DAO interface*
Lampiran 37. *WishlistDAO interface*
Lampiran 38. *Wishlist Repository*
Lampiran 39. *Wishlist Service*
Lampiran 40. *Wishlist Controller*
Lampiran 41. *User model*
Lampiran 42. *DAO interface*
Lampiran 43. *UserDAO interface*
Lampiran 44. *User Repository*
Lampiran 45. *User Service*
Lampiran 46. *User Controller*
Lampiran 47. *Alamat model*
Lampiran 48. *Buku model*
Lampiran 49. *DetailKeranjang model*
Lampiran 50. *ItemOrder model*
Lampiran 51. *Keranjang model*
Lampiran 52. *Order model*
Lampiran 53. *OrderBuku model*
Lampiran 54. *DAO interface*
Lampiran 55. *AlamatDAO interface*
Lampiran 56. *DetailKeranjangDAO interface*
Lampiran 57. *ItemOrderDAO interface*
Lampiran 58. *KeranjangDAO interface*
Lampiran 59. *OrderDAO interface*
Lampiran 60. *Alamat Repository*
Lampiran 61. *ItemOrder Repository*

Lampiran 62. KeranjangDetail *Repository*

Lampiran 63. Keranjang *Repository*

Lampiran 64. Order *Repository*

Lampiran 65. Alamat *Service*

Lampiran 66. Keranjang *Service*

Lampiran 67. Order *Service*

Lampiran 68. Alamat *Controller*

Lampiran 69. Keranjang *Controller*

Lampiran 70. Order *Controller*

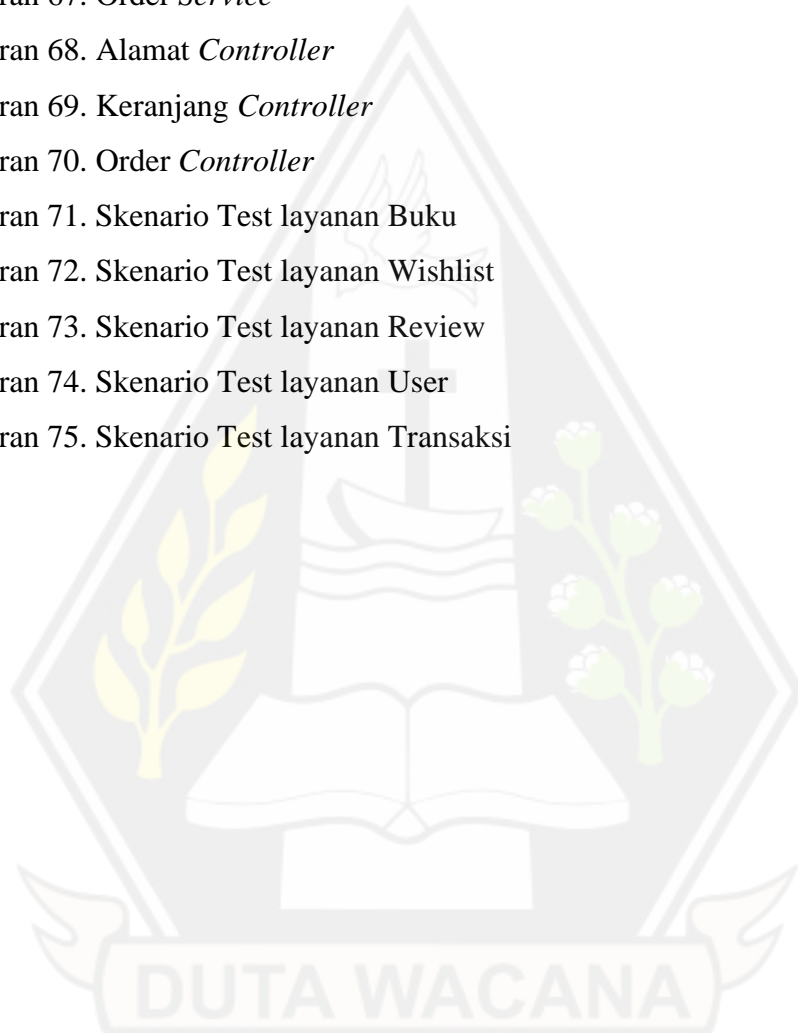
Lampiran 71. Skenario Test layanan Buku

Lampiran 72. Skenario Test layanan Wishlist

Lampiran 73. Skenario Test layanan Review

Lampiran 74. Skenario Test layanan User

Lampiran 75. Skenario Test layanan Transaksi



BAB 1

Pendahuluan

1.1. Latar Belakang

Dalam pengembangan sebuah sistem, proses pembaruan, pengoptimalan ataupun penambahan suatu fitur di sebuah sistem sering terjadi. Akan tetapi dengan adanya pengoptimalan suatu sistem terkadang menyebabkan kerusakan kode di dalam sistem yang lama. Hal tersebut lebih sering dijumpai di sebuah pengembangan sistem dengan menggunakan arsitektur monolitik. Arsitektur Monolitik sendiri merupakan salah satu arsitektur yang paling banyak digunakan untuk pengembangan aplikasi saat ini. Menurut Ghifari Munawar dan Ade Hodijah (2018) dengan menggunakan arsitektur Monolitik aplikasi akan dibungkus di dalam satu package besar, dan akan menyebabkan pengaruh terhadap satu bagian program dikarenakan perubahan di bagian lain (*tightly coupled*). Sehingga, semakin besar dan kompleks sebuah aplikasi dalam arsitektur monolitik, akan semakin kesusahan dalam membuat perubahan. Untuk mengatasi kekurangan arsitektur Monolitik, penerapan arsitektur Layanan Mikro (*Microservices*) menjadi solusi yang baik.

Dalam arsitektur Layanan Mikro, tiap fungsi / *feature* dibagi menjadi modul yang independen yang berkomunikasi melalui API. Melalui arsitektur tersebut, aplikasi lebih mudah dipelihara, mempermudah pengujian, dan tiap fungsi dapat disebarkan secara independen. Keunggulan tersebut dapat mengatasi kekurangan yang ada pada arsitektur Monolitik, yang kesulitan beradaptasi dengan perubahan. Tidak hanya itu, tetapi arsitektur Layanan Mikro juga lebih mudah dipahami karena tiap fungsionalitas terbagi menjadi komponen yang kecil dan sederhana. Penerapan arsitektur Layanan Mikro dapat dilakukan dengan berbagai cara, salah satunya dengan menggunakan *Representational State Transfer (REST) Application Programming Interface (API)*. REST API merupakan arsitektur komunikasi yang menggunakan *Hypertext Transfer Protocol (HTTP)* sebagai protokol dalam pertukaran data. Dalam protokol HTTP, *JavaScript Object Notation (JSON)*

digunakan sebagai media pengiriman data / komunikasi antar komponen. Maka dari itu, REST API dapat membantu dalam penstrukturan penerapan routes antar API sehingga lebih mudah dibaca dan terstruktur dengan baik.

Dalam penerapan arsitektur Layanan Mikro ini, yang menjadi studi kasus adalah aplikasi toko Buku berbasis seluler (*Mobile*). Aplikasi toko buku digunakan untuk membantu *user* melihat deskripsi melihat *sample* buku, dan *user* dapat melakukan pembelian buku melalui transaksi online. Untuk membantu menangani keamanan, arsitektur Layanan Mikro akan menerapkan *API Gateway*. Hal ini juga membuat aplikasi dapat lebih mudah beradaptasi dengan perubahan. Dengan studi kasus ini, penulis akan menganalisis proses penerapan arsitektur Layanan Mikro dengan *API Gateway* dalam aplikasi toko buku.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat dijabarkan rumusan masalah seperti berikut:

1. Bagaimana *API Gateway* dalam arsitektur Layanan Mikro dapat membantu proses penanganan *request* dan pengamanan layanan?

1.3. Batasan Penelitian

Dalam penelitian ini, diterapkan batasan-batasan untuk permasalahan, batasan tersebut seperti berikut:

1. Penerapan arsitektur layanan mikro menggunakan RESTful API dengan *gateway*.
2. Pembuatan fungsi sistem *backend* akan didasarkan pada *testing* sebelumnya telah dibuat.
3. Pembuatan *API Gateway* akan menggunakan *Amazon Web Service API Gateway*.
4. *API Gateway* akan diterapkan sebagai *single entry point* untuk *client* memanggil setiap fungsi pada semua layanan.

5. Pembuatan aplikasi seluler hanya untuk pengujian pemanggilan layanan.
6. Penelitian hanya dilakukan pada sistem *backend* dari aplikasi toko buku yang dibangun.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, dapat dirumuskan tujuan dari penelitian ini, yaitu:

1. Membuat arsitektur layanan mikro dengan *API Gateway* untuk aplikasi toko buku.
2. Mempelajari cara kerja arsitektur Layanan Mikro yang menggunakan RESTful API.

1.5. Manfaat Penelitian

Penelitian ini bermanfaat untuk pembaca hasil penelitian, yaitu:

1. Menambah pengetahuan mengenai proses penerapan arsitektur Layanan Mikro dengan *API Gateway*.
2. Menambah pengetahuan serta gambaran dalam membangun aplikasi dengan arsitektur Layanan Mikro.
3. Dengan penelitian ini, diharapkan dapat membantu pembaca memahami proses penerapan dan keuntungan arsitektur Layanan Mikro dengan *API Gateway*.

1.6. Metode Penelitian

Dalam penelitian ini, penulis menggunakan metode penelitian yang dibagi menjadi 2 bagian, seperti berikut:

1. Tahap Persiapan

Dalam tahap ini, yang akan dilakukan adalah:

- a. Studi permasalahan

Dalam studi permasalahan, penulis akan memahami permasalahan umum yang biasa dialami akibat penerapan arsitektur Monolitik.

b. Tinjauan Pustaka

Dalam Tinjauan Pustaka, penulis akan mencari dan mengkaji berbagai sumber baik jurnal maupun buku, terkait arsitektur *Microservices*, arsitektur *Monolithic*, *Cloud Services*, dan RESTful API dalam *Microservices*.

c. *Requirement Gathering*

Penulis akan mengumpulkan informasi, kebutuhan dan proses bisnis dalam pengembangan aplikasi toko buku.

2. Tahap Perancangan

Dalam tahap Perancangan, akan dilakukan 4ndepen-langkah seperti berikut:

a. Desain Model Arsitektur *Microservices*

Penulis akan merancang desain dari backend aplikasi toko buku dengan arsitektur Layanan Mikro dengan menggunakan *API Gateway*. Untuk mendesain layanan mikro ini, akan mengacu pada pendekatan *Domain Driven Design*.

b. Desain *Request* dan *Response* tiap layanan

Dalam tahap ini, penulis akan merancang bentuk *request* dan *response* yang akan dimiliki tiap layanan. Bentuk tersebut akan mengacu pada desain layanan.

3. Tahap Implementasi

a. Pemrograman *backend* Arsitektur Layanan Mikro

Dalam tahap ini, Penulis akan melakukan proses pengembangan sistem backend dari aplikasi toko buku yang dikerjakan. Sistem backend akan menggunakan API *Gateway*.

b. Testing sistem Layanan Mikro

Tahap ini dilakukan untuk memastikan kode program dengan arsitektur Layanan Mikro berjalan dengan baik tanpa masalah dan *bug*.

c. *Deployment* layanan yang telah dibuat

Tahap ini merupakan tahap akhir dalam arsitektur Layanan Mikro, tahap ini bertujuan agar tiap layanan dapat digunakan dalam aplikasi seluler toko buku.

d. Penerapan API *Gateway* sebagai *single entrypoint* layanan

Dalam tahap ini, penulis akan memasang tiap layanan yang sebelumnya diunggah dengan sebuah API *Gateway*.

4. Tahap Evaluasi

a. Analisis kesesuaian *microservices* dengan *requirement*

Analisis akan dilakukan dengan menjalankan *test case* yang sebelumnya dibuat berdasarkan *requirement*. Hasil dari *testing* akan menjadi penentu kesesuaian layanan dengan *requirement*.

b. Analisis penerapan API *Gateway* dalam Arsitektur *Microservices*

Analisis akan dilakukan dengan melihat proses penerapan API Gateway dari segi keamanan, *change adaptability*.

1.7. Sistematika Penulisan

Untuk memperjelas pembahasan dalam penelitian ini, laporan penelitian ini dibagi menjadi 5 bab yaitu Bab 1 Pendahuluan, Bab 2 Tinjauan Pustaka, Bab 3 Metodologi Penelitian, Bab 4 Implementasi dan Analisis Sistem dan Bab 5 Kesimpulan dan Saran, yang akan diperjelas sebagai berikut:

BAB 1, bab ini berisi mengenai latar belakang, perumusan masalah, 6ndepen masalah, tujuan penelitian, manfaat penelitian, metode atau pendekatan penelitian dan sistematika penelitian. Dalam bab ini juga dijelaskan secara garis besar mengenai penelitian yang akan dijalankan oleh penulis serta manfaat bagi semua pihak yang terkait dengan penelitian ini.

BAB 2, bab ini berisi tentang tinjauan 6ndepen dan landasan teori. Pada tinjauan 6ndepen akan menjelaskan mengenai penelitian serupa yang telah dilakukan yang akan menjadi referensi penulis dalam melaksanakan penelitian. Pada landasan teori berisi kumpulan teori dan definisi mengenai tools yang akan digunakan maupun teori yang dikutip dari berbagai artikel ilmiah dan buku untuk membantu dalam penyelesaian penelitian ini.

BAB 3, bab ini berisi tentang perancangan sistem yang akan dibuat berupa arsitektur sistem, blok diagram sistem, data yang akan digunakan dan diimplementasikan ke dalam sistem.

BAB 4, bab ini berisi mengenai implementasi dan analisis sistem yang telah dibuat berdasarkan requirement yang dijelaskan sebelumnya.

BAB 5, bab ini berisi mengenai kesimpulan dari hasil sistem yang telah dijalankan. Lalu juga diberikan saran untuk penelitian selanjutnya dan diharapkan lebih berguna untuk menghasilkan penelitian yang lebih baik dari sebelumnya.



BAB 5

Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan analisis sistem sebelumnya, dapat disimpulkan bahwa semua layanan yang dibuat sesuai karakteristik layanan mikro. Seperti yang telah diuraikan dalam sub-bab 4.2.4, layanan yang dibuat bersifat modular, dengan basis data yang tidak terpusat, dan dirancang dengan fokus pada produk. Lalu dalam penerapan API *gateway*, pemanfaatan *feature* seperti:

1. Penerapan API *key* untuk mengakses layanan yang telah dibuat membantu mencegah *request* dari sumber yang tidak dikenal berhasil menolak *request* dari sumber tanpa *key*.
2. Penerapan *usage plan* untuk mengatur *request limit* berhasil membatasi *request*, dan dapat membantu mencegah *request overload* pada *server* layanan.

5.2. Saran

1. Saran untuk pengembangan layanan dengan bantuan API *gateway*, untuk membuat *authorizer* dengan *bearer* saat menggunakan *access token*. Dengan adanya *bearer* pada *access token*, API *key* dapat dimasukkan ke dalam *bearer* dan dapat dikirim bersama dengan *access token*.
2. Disarankan membuat *authorizer* dengan *identity provider 3rd party* seperti OAuth. Dengan adanya *authorizer* tersebut, keamanan API akan semakin baik.

Daftar Pustaka

- Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (2010). Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1), 341-377. doi:10.1007/s11042-010-0660-6
- Mazlami, G., Cito, J., & Leitner, P. (2017). Extraction of Microservices from Monolithic Software Architectures. *2017 IEEE International Conference on Web Services (ICWS)*, 524-531. doi:10.1109/icws.2017.61
- Munawar, G., & Hodijah, A. (october 2018). Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK. *Publikasi Jurnal & Penelitian Teknik Informatika*, 3(1), 232-239.
- Rattanaungrot, S., & White, M. (october 2016). Development of service oriented mobile AR applications for museum learning activities. *2016 22nd International Conference on Virtual Systems and MultiMedia*, 1-8. doi:10.1109/VSMM.2016.7863202
- Richardson, C. (2019). *Microservices patterns: With examples in Java*. Shelter Island, NY: Manning Publications.
- Balalaie, A., Heydarnoori, A., Pooyan, J., Tamburri, D. A., & Lynn, T. (2018). Microservices Migration Patterns. *Software: Practice and Experience*, 48(1), 2019-2042. doi:10.1002/spe.2608
- Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The Journey So Far and Challenges Ahead. *IEEE Software*, 35(3), 24-35. doi:https://doi.org/10.1109/MS.2018.2141039
- Pahl, C., & Pooyan, J. (2016). Microservices: A Systematic Mapping Study. *CLOSER*(1), 137-146.

Shadija, D., Rezai, M., & Hill, R. (2017). Towards an Understanding of Microservices. *2017 23rd International Conference on Automation and Computing (ICAC)*, 1-6. doi:10.23919/ICOnAC.2017.8082018

Xu, R., Jin, W., & Kim, D. (2019). Microservice Security Agent Based On API Gateway. *Sensors*, *19*. doi:<https://doi.org/10.3390/s19224905>

Zhao, J. T., Jing, S. Y., & Jiang, L. Z. (2018, September). Management of API Gateway Based on Micro-service Architecture. *Journal of Physics: Conference Series*, *1087*, 1-7. doi:10.1088/1742-6596/1087/3/032032

