



**PERANCANGAN ARSITEKTUR BACKEND MICROSERVICE
PADA STARTUP CAMPAIGN.COM**

TUGAS AKHIR



Zaky Riko Virgiawan
41517120048

UNIVERSITAS
MERCU BUANA

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2021**



**PERANCANGAN ARSITEKTUR BACKEND MICROSERVICE
PADA STARTUP CAMPAIGN.COM**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
Zaky Riko Virgiawan
41517120048

UNIVERSITAS
MERCU BUANA

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2021

LEMBAR PERNYATAAN ORISINALITAS

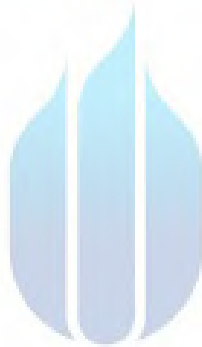
Yang bertanda tangan dibawah ini:

NIM : 41517120048

Nama : Zaky Riko Virgiawan

Judul Tugas Akhir : Perancangan Arsitektur Backend Microservice pada
Startup Campaign.com

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan di dalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.



Jakarta, 17 Januari 2022



Zaky Riko Virgiawan

UNIVERSITAS
MERCU BUANA

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Zaky Riko Virgiawan
NIM : 41517120048
Judul Tugas Akhir : Perancangan Arsitektur Backend Micoservice pada Startup Campaign.com

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 17 Januari 2022

UNIVERSITAS
MERCU BUANA



Zaky Riko Virgiawan

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Zaky Riko Virgiawan
 NIM : 41517120048
 Judul Tugas Akhir : Perancangan Arsitektur Backend Microservice pada Startup Campaign.com

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis	Status
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi	Diajukan
		Jurnal Nasional Terakreditasi	
		Jurnal International Tidak Bereputasi	Diterima
		Jurnal International Bereputasi	
Disubmit/dipublikasikan di :	Nama Jurnal	: <i>Alabam jurnal ilmiah keangsaan dan...</i>	
	ISSN	: 1907 - 4174	
	Link Jurnal	:	
	Link File Jurnal Jika Sudah di Publish	:	

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui
 Dosen Pembimbing TA

UNIVERSITAS
MERCU BUANA

Dr. Harwikarya, MT
Dr. Harwikarya, MT

Jakarta, 17 Januari 2022



Zaky Riko Virgiawan
Zaky Riko Virgiawan

LEMBAR PERSETUJUAN PENGUJI

NIM : 41517120048
Nama : Zaky Riko Virgiawan
Judul Tugas Akhir : Perancangan Arsitektur Backend Microservice pada Startup Campaign.com

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 24 Maret 2022



(Dr. Leonard Geormanto)

UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI

NIM : 41517120048
Nama : Zaky Riko Virgiawan
Judul Tugas Akhir : Perancangan Arsitektur Backend Microservice pada Startup Campaign.com

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 23 Februari 2022



(Eliyani, Dr. Ir.)

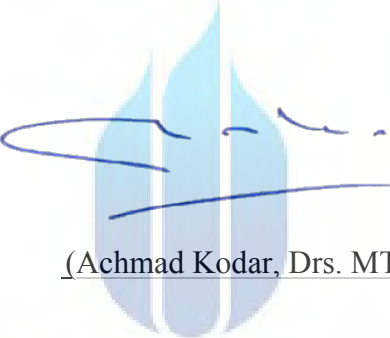
UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI

NIM : 41517120048
Nama : Zaky Riko Virgiawan
Judul Tugas Akhir : Perancangan Arsitektur Backend Microservice pada Startup Campaign.com

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 23 Februari 2022



(Achmad Kodar, Drs. MT)

UNIVERSITAS
MERCU BUANA

LEMBAR PENGESAHAN

NIM : 41517120048
Nama : Zaky Riko Virgiawan
Judul Tugas Akhir : Perancangan Arsitektur Backend Microservice pada Startup Campaign.com

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 23 Februari 2022

Menyetujui,



(Dr. Harwikarya, MT)

Dosen Pembimbing

MERCU BUANA

Mengetahui,



(Wawan Gunawan, S.Kom, MT)

Koord. Tugas Akhir Teknik Informatika



(Ir. Emil R. Kaburuan, Ph.D., IPM)

Ka. Prodi Teknik Informatika

KATA PENGANTAR

Puji syukur kita panjatkan karena rahmat dan hidayah-Nya yang senantiasa dilimpahkan kepada kita semua. Penulis menyadari bahwa tanpa pertolongan dan bantuan Allah SWT, dosen pembimbing, dan rekan-rekan di Universitas Mercu Buana, penulis tidak dapat menyelesaikan Tugas Akhir ini dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Orang tua dan sahabat yang selalu memberikan semangat dan bantuan secara tidak langsung selama proses perkuliahan
2. Pak Dr. Harwikarya, MT selaku pembimbing Tugas Akhir yang banyak memberi masukan ilmu dan arahan kepada penulis dalam proses penyusunan Tugas Akhir
3. Bapak Wawan Gunawan, S. Kom, MT selaku Kepala Program Studi Teknik Informatika di Universitas Mercu Buana
4. Seluruh dosen Fakultas Ilmu Komputer Universitas Mercu Buana yang telah memberikan ilmunya selama proses perkuliahan
5. Teman-teman yang memberikan semangat selama pelaksanaan Tugas Akhir Akhir kata, penulis berharap semoga laporan kegiatan ini dapat bermanfaat untuk banyak orang. Aamiin.

Jakarta, 17 Januari 2022

UNIVERSITAS
MERCU BUANA



Zaky Riko Virgiawan

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS	ii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	iii
SURAT PERNYATAAN LUARAN TUGAS AKHIR	iv
LEMBAR PERSETUJUAN PENGUJI.....	v
LEMBAR PENGESAHAN.....	viii
ABSTRAK.....	ix
ABSTRACT	x
KATA PENGANTAR	xi
DAFTAR ISI	xii
NASKAH JURNAL.....	1
KERTAS KERJA	10
BAB 1. LITERATURE REVIEW.....	11
BAB 2. ANALISIS DAN PERANCANGAN.....	17
BAB 3. SOURCE CODE.....	20
BAB 4. TAHAPAN EKSPERIMEN	25
BAB 5. HASIL SEMUA EKSPERIMEN	38
DAFTAR PUSTAKA	49
LAMPIRAN DOKUMEN HAKI	51
LAMPIRAN KORESPONDENSI	53

Perancangan Arsitektur Backend Microservice pada Startup Campaign.com

Zaky Riko Virgiawan^{#1}, Harwikarya^{*2}

[#]*Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercubuana*

¹41517120048@student.mercubuana.ac.id

²harwikarya@mercubuana.ac.id

Abstrak— Dalam memulai pengembangan perangkat lunak, salah satu poin terpenting adalah menentukan arsitektur teknologi sesuai dengan analisa kebutuhan produk. Sehingga nantinya dapat berjalan selaras bersamaan dengan visi produk tersebut. Namun, seiring berjalannya waktu dengan user dan data yang semakin bertambah, diperlukan adanya perombakan arsitektur kembali dengan membangun pondasi yang lebih kokoh. Hal ini dialami oleh perusahaan *startup* sosial Campaign.com.

Campaign.com merupakan sebuah aplikasi untuk mengambil aksi sosial yang dapat menyalurkan donasi dari sponsor. Desain sistem yang dibangun di campaign.com masih menggunakan arsitektur monolitik. Antarmuka pengguna, pemrosesan logika, dan akses data digabungkan menjadi satu program dan ditempatkan dalam satu basis data. Namun hal ini membuat aplikasi sering mengalami *bugs* dan *server downtime*. Dengan masalah yang sedang dihadapi ini penulis mencoba melakukan eksperimen dengan merancang dan menganalisa sistem arsitektur *backend*. Dengan menggunakan pendekatan arsitektur *microservice*, basis data akan dipisah berdasarkan modulnya, sehingga lebih efisien dalam penulisan kode

program. Pada sistem *microservice*, penulis juga sudah menggunakan *swagger* untuk dokumentasi API.

Tujuan yang ingin dicapai dari perancangan arsitektur ini adalah mendeskripsikan proses penyederhanaan dari sebuah sistem arsitektur *backend microservice* agar mudah dalam pemeliharaan sistem dan penggunaan algoritma *Docker*.

Kata kunci— Arsitektur, *microservice*, *docker*, integrasi

I. PENDAHULUAN

Campaign.com merupakan sebuah aplikasi untuk mengambil aksi sosial yang mana dapat menyalurkan donasi dari sponsor. Di campaign.com desain sistem yang dibangun masih menggunakan arsitektur monolitik. Antarmuka pengguna, pemrosesan logika, dan akses data digabungkan menjadi satu program dan ditempatkan dalam satu basis data, namun hal ini membuat aplikasi sering mengalami bugs dan server downtime, dengan masalah yang sedang dihadapi ini penulis mencoba melakukan perbandingan dan menganalisa sistem arsitektur backend yang

digunakan oleh Gojek.

Gojek sukses menjadi startup pertama asal Indonesia yang mendapat gelar unicorn. Tidak butuh waktu lama bagi perusahaan aplikasi PT Go-Jek Indonesia untuk mendapatkan tempat dihati masyarakat, aplikasi karya anak bangsa ini memberikan dampak yang luar biasa untuk memajukan ekonomi kreatif, bagaimana tidak setelah namanya mencuat di Indonesia banyak sekali perusahaan startup bermunculan, mulai dari ide bisnis yang mirip-mirip sampai yang tidak kalah inovatifnya dengan Gojek. Salah satu faktor kesuksesan Gojek adalah teknologi yang digunakan untuk model bisnis yang mereka bangun. Di Dalam aplikasi Gojek kita dapat memesan makanan, jasa antar jemput, mengirim barang dan masih banyak lagi fitur lainnya, hal inilah yang membuat penulis tertarik untuk melakukan penelitian teknologi informasi yang gojek terapkan.

Wildan G. Budhi seorang *Tech and Macro Economics Enthusiasm* melalui websitenya <https://wildangbudhi.medium.com>, telah menulis artikel tentang arsitektur microservice yang digunakan oleh gojek untuk membangun sistem backend mereka. Dalam tulisannya ada satu kalimat menarik “1App, 1 Server, Many Product” ini yang menjadi identitas utama microservice, satu aplikasi, satu server namun ada beberapa produk di dalamnya[2].

Yang membedakan dengan penelitian sebelumnya atau referensi yang penulis cantumkan adalah adanya integrasi data dari sistem lama menuju sistem baru. Jadi bukan sekedar implementasi tetapi juga melakukan dekomposisi arsitektur basis data. Penelitian akan dilaksanakan secara bertahap sesuai dengan rancangan yang telah disusun. Mulai dari studi literatur sampai akhirnya implementasi pada subjek studi kasus[4-5].

II. LANDASAN TEORI

A. *Microservice*

Microservice adalah teknik pengembangan perangkat lunak yang mengatur aplikasi sebagai kumpulan layanan yang

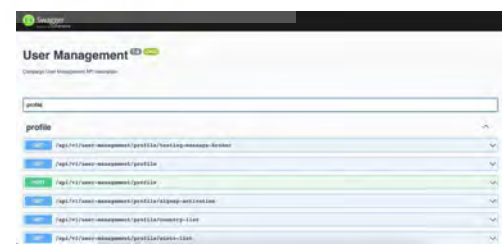
digabungkan secara longgar. Kopling adalah tingkat saling ketergantungan antara modul perangkat lunak. Jadi kita dapat mengatakan *Microservice* adalah tempat aplikasi atau program dibagi menjadi banyak modul kecil yang saling ketergantungan. Penggunaan satu aplikasi dan satu server dengan teknik *Microservices* dapat membagi semua program menjadi program atau aplikasi independen, baik itu di server terpisah atau di satu server tetapi menggunakan *Docker Container*.

B. *Docker*

Docker adalah teknologi virtualisasi sistem operasi berbasis *Container* untuk membangun, menguji, dan menyebarkan aplikasi terdistribusi dalam lingkungan yang terisolasi. *Docker* tidak membangun mesin virtual sendiri, sehingga lebih hemat memori, processor dan storage. *Docker* menyediakan virtualisasi ringan dengan overhead hampir nol[2].

C. *Swagger*

Swagger merupakan tools yang digunakan untuk mendeskripsikan struktur API sehingga mudah untuk dibaca. Dengan membaca struktur API, secara otomatis membuat pendokumentasian menjadi interaktif karena dapat diuji secara langsung dan memiliki user interface. Keuntungan lainnya dari menggunakan *swagger* dapat digunakan secara bersamaan.

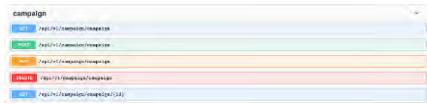


Gambar 1. User Management

Dokumentasi sangat penting sebagai alat komunikasi antara Back-end Developer dan Front-end Developer. Dengan dokumentasi yang readable, client akan lebih produktif.

D. *Monolith*

Aplikasi Monolitik adalah perangkat lunak di mana komponen yang berbeda (seperti otorisasi, logika bisnis, modul notifikasi, dll.) digabungkan menjadi satu program dari satu



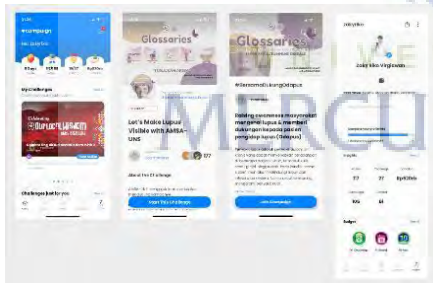
Gambar 6. API microservice melaluiSwagger



Gambar 7. API microservice Testing Online Tool

Pengujian dilakukan juga melalui <https://reqbin.com/> untuk mengecek kecepatan dan ukuran data balikan dari beberapa api, dibandingkan dengan api arsitektur monolitik, ini lebih cepat dengan rata-rata 800-900ms per request.

2. Testing secara produk, penulis langsung berkoordinasi dengan team front-end apps (ios & android), front-end web, QA, product. Testing pada tahap ini dapat diartikan bahwa API sudah diintegrasikan dengan produk-produk yang ada pada campaign.com [9-10].



Gambar 8. Aplikasi Campaign.com

Ini adalah hasil user interface dari beberapa halaman (home, user, campaign, challenge) aplikasi campaign.com yang telah menggunakan api sistem microservice.



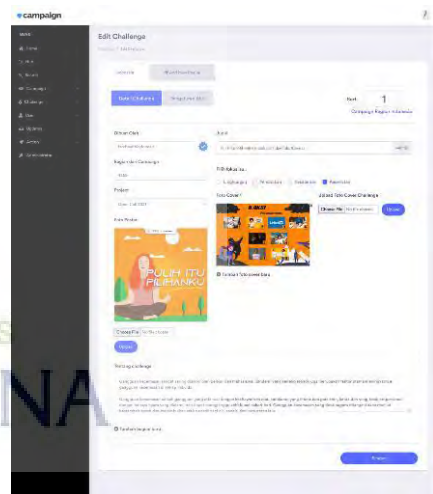
Gambar 9. Action Page



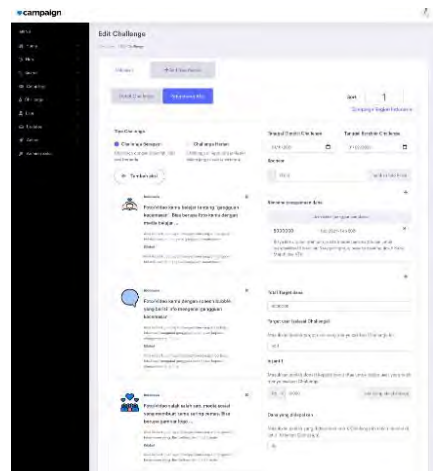
Gambar 10. Explore Page



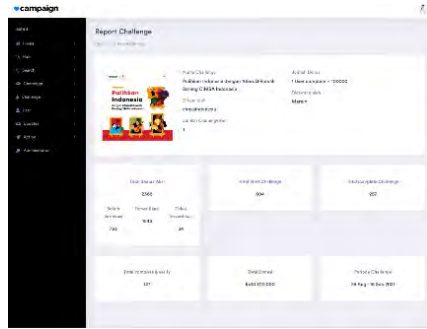
Gambar 11. Challenge Page
Dan ini adalah tampilan beberapa halaman website utama campaign.com tentunya dengan datayang disajikan telah menggunakan api sistem microservice.



Gambar 12. Edit Challenge Page



Gambar 13. Edit Detail Challenge Page



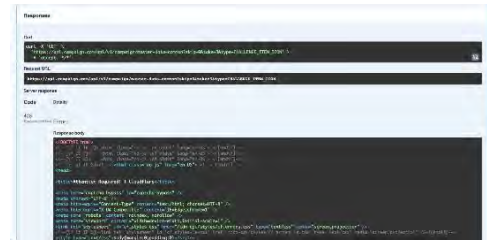
Gambar 14. Report Challenge Page

Ini adalah tampilan beberapa halaman admin dashboard campaign, website yang digunakan untuk mengatur dan mengolah data aplikasi campaign.com dan telah terintegrasi dengan api sistem microservice.

Namun dengan keterbatasan biaya dan skill penulis, sangat disayangkan belum dapat menerapkan automation testing atau bahkan unit testing sehingga arsitektur backend microservice yang telah selesai pada eksperimen ini belum teruji secara keseluruhan dengan kondisi yang mungkin belum sempat teruji pada dua poin yang penulis sampaikan diatas.

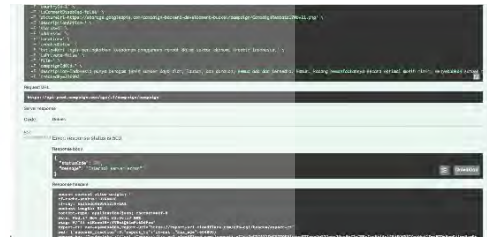
B. Bug Fixing

Setelah masa pengujian hasil eksperimen perlu adanya perbaikan atau yang sering disebut bug fixing. Dalam hal ini ketika pengujian berlangsung tidak jarang user mendapatkan berbagai macam error dari API sistem microservice, mulai dari response 500 Internal Server Error, 502 Bad Gateway dan 403 forbidden. Setiap pesan memiliki cara penanganannya tersendiri. Untuk cara perbaikan, penulis menggunakan metode debugging agar dapat mengidentifikasi error lalu menemukan lokasi yang menyebabkan error dan kemudian melakukan analisa terhadap error yang telah ditemukan.



Gambar 15. Bug Fix Error API Status 403

Error response 403 biasanya terjadi karena tidak mendapatkan permissions terkait file atau data pada function sebuah API. Dan dapat dilihat bahwa letak kesalahan function tersebut telah terdeteksi oleh swagger, ini juga suatu keuntungan tersendiri untuk penulis yang telah menggunakan swagger dengan memudahkan proses debugging sehingga tidak perlu mencari file secara keseluruhan project.



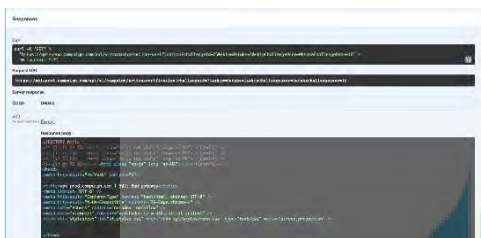
Gambar 16. Bug Fix Error API Status 500

Lalu response error status 500 ini yang paling sering ditemukan pada saat pengujian dan API diimplementasi oleh front-end, sebabnya bermacam-macam mulai dari salah mengirim parameter, pencegahan atau pembatasan nilai null, dan yang cukup sering terjadi kesalahan penulisan query pada controller. Perbaikannya pun simple dan tidak terlalu rumit karena sama dengan error sebelumnya yaitu sudah dijelaskan dan ditunjukkan oleh pesan swagger.



Gambar 17. Bug Fix Error API Status 200

Hasil apabila API sudah dilakukan bug fixing akan merespon status 200 seperti gambar diatas. Pada response body juga sudah mengembalikan data yang sesuai dengan parameter yang dikirim oleh user.



Gambar 18. Bug Fix Error API Status 502

Sempat juga mengalami error response status 502 yang menjadikan API terasa lambat dan bila mendapat request yang cukup tinggi sampai downtime dan solusi yang diterapkan untuk masalah ini adalah melakukan indexing pada tabel database agar error 502 dapat teratasi.

Meskipun beberapa contoh error yang penulis sampaikan diatas hal yang perlu diperhatikan adalah bahwa setiap kali mengalami kerusakan pada sebuah function API tidak mengakibatkan satu aplikasi tidak dapat berjalan dan berdampak ke seluruh fitur yang ada, hanya pada function module yang sedang mengalami perbaikan saja yang akan terdampak, disini sangat terasa sekalimannya dari arsitektur microservice karena jika masih menggunakan monolitik, satu API yang mengalami masalah dapat berdampak satu aplikasi force close tidak dapat diakses secara keseluruhan.

V. KESIMPULAN DAN SARAN

Pada akhirnya hasil dari karya ilmiah ini telah menjawab dua pertanyaan dari masalah yang ada yaitu bagaimana proses perancangan arsitektur backend microservice dan bagaimana cara mengintegrasikan sistem monolitik menjadi microservice. Tentunya dengan kekurangan dan keterbatasan ilmu, penulis sudah memberikan penjelasan dan menyusun karya ilmiah ini dengan maksimal. Yang intinya dapat diambil pelajaran bahwa pemilihan sistem arsitektur backend dalam membangun sebuah software dan proses pengembangannya harus dianalisis sesuai dengan kebutuhan. Dua arsitektur yang penulis kaji yaitu monolitik dan microservice memiliki kelebihan dan kekurangannya masing-masing. Berikut perbandingan dalam bentuk tabel.

Monolith		Microservices	
Kelebihan	Kekurangan	Kelebihan	Kekurangan
Cepat untuk diterapkan (fast deployment)	Tidak dapat dipercaya, terdapat error kecil di satu tempat akan merusak keseluruhan	Skalabilitas dapat menangani jutaan request per detik	Kompleksitas teknologi pembelajaran untuk mendukung layanan mikro
Mudah untuk dipahami	Perbarui satu baris kode harus memperbaharui seluruh aplikasi	Sambungan antar komponen, satu sistem mati tidak mempengaruhi keseluruhan sistem	Memantau setiap komponen, belajar membaca grafik untuk memeriksa situasi atau error
-	Teknologi tidak dapat terdiri dari beberapa bahasa pemrograman	-	Kompleksitas dalam pengujian

Berdasarkan eksperimen karya ilmiah ini, penulis mengetahui perbedaan antara arsitektur monolitik dan microservice dengan kekurangan dan kelebihanannya. Yang mana Campaign.com sebenarnya dapat menggunakan kedua arsitektur tersebut, namun lebih efisien jika menggunakan arsitektur microservice. Hal ini terbukti dari hasil eksperimen yang telah dilakukan.

Dan dengan menggunakan arsitektur microservice secara devops tagihan untuk server menjadi terjangkau dikarenakan database sudah dipisah-pisah sesuai dengan modul dan maintenance untuk load balancing pun jauh lebih mudah karena apabila traffic sedang naik tidak harus semua dinaikan cukup modul database yang terkait saja.

Kemudian berikut perbandingan dengan metode sejenis dari referensi yang telah dicantumkan:

Metode yang digunakan	Metode Sejenis
Penelitian ini ketika migrasi data tidak melakukan research secara mendalam hanya berfokus kepada perpindahan data dari struktur monolitik menuju microservice.	[15] dijelaskan data migration perlu memperhatikan tiga hal utama yaitu cost, modularity, and willingness to adopt agar perpindahan data dapat berjalan dengan lancar.
Algoritma docker yang masih sederhana dan belum menggunakan DockerAnalyser. Dikarenakan container images belum ada yang multiple.	Namun referensi [14] telah menggunakan dockerAnalyser pada sistem microservice guna menganalisa multiple container image nya.
Penggunaan docker begitu efektif dalam pendekatan backend arsitektur microservice karena service telah dipisah kecil-kecil dan spesifik.	Sama halnya pada jurnal [4] pada kesimpulannya bahwa docker cocok untuk penyebaran aplikasi berbasis sistem microservice.

REFERENSI

- [1] M. Allahyari *et al.*, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," 2017.
- [2] I. Arsitektur, M. Pada, B. End, I. Atlantas, and B. Website, "Jurnal Teknologi Terpadu INFORMASI ATLANTAS BERBASIS WEBSITE," vol. 6, no. 2, pp. 96–104, 2020.
- [3] K. Gos and W. Zabierowski, "The Comparison of Microservice and Monolithic Architecture," *International Conference on Perspective Technologies and Methods in MEMS Design*, 2020, pp. 150–153. doi: 10.1109/MEMSTECH49584.2020.9109514.
- [4] R. Khalida, A. Muhajirin, S. Setiawati, J. Raya, J. Raya, and P. B. Utara, "Teknis Kerja Docker Container untuk Optimalisasi Penyebaran Aplikasi."
- [5] A. Darmayantie, "DESAIN SISTEM TERFEDERASI DENGAN PENDEKATAN MICROSERVICE ARCHITECTURE PADA KASUS STUDI SISTEM PELAPORAN PAJAK," *Jurnal Ilmiah Informatika Komputer*, vol. 25, no. 1, pp. 50–63, 2020, doi: 10.35760/ik.2020.v25i1.2523.
- [6] M. A. A. Putra, I. Fitri, and A. Iskandar, "Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker," *JURNAL MEDIA INFORMATIKABUDIDARMA*, vol. 4, no. 1, p. 9, Jan. 2020, doi: 10.30865/mib.v4i1.1729.
- [7] R. Khalida, A. Muhajirin, and S. Setiawati, "Teknis Kerja Docker Container untuk Optimalisasi Penyebaran Aplikasi," *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, vol. 7, no. 2, pp. 167–176, 2019, doi: 10.33558/piksel.v7i2.1819.
- [8] M. Waseem, P. Liang, M. Shahin, A. di Salle, and G. Márquez, "Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective," Aug. 2021, doi: 10.1016/j.jss.2021.111061.
- [9] V. K. Pachghare, "Microservices Architecture for Cloud Computing," 2016. [Online]. Available: <https://www.researchgate.net/publication/317637883>
- [10] R. Elsen, M. R. Nashrulloh, R. Cahyana, A. Mulyani, and A. Latifah,

- “Microservice architecture design for autograder using distributed architecture,” *IOP Conference Series: Materials Science and Engineering*, vol. 1098, no. 3, p. 032083, Mar. 2021, doi: 10.1088/1757-899x/1098/3/032083.
- [11] H. Suryotrisongko, D. P. Jayanto, and A. Tjahyanto, “Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot,” in *Procedia Computer Science*, 2017, vol. 124, pp. 736–743. doi: 10.1016/j.procs.2017.12.212.
- [12] T. Pötter, “Cloud with Kubernetes, Docker, Microservices, Serverless, Versioning,” doi: 10.13140/RG.2.2.23617.10088.
- [13] A. Raj, K. S. Jasmine, and P. G. Student, “Building Microservices with Docker Compose.”
- [14] A. Brogi, D. Neri, and J. Soldani, “A microservice-based architecture for (customisable) analyses of Docker images,” *Software - Practice and Experience*, vol. 48, no. 8, pp. 1461–1474, Aug. 2018, doi: 10.1002/spe.2583.
- [15] F. Auer, V. Lenarduzzi, M. Felderer, and D. Taibi, “From monolithic systems to Microservices: An assessment framework,” *Information and Software Technology*, vol. 137, Sep. 2021, doi: 10.1016/j.infsof.2021.106600.
- [16] T. Goethals, D. Kerkhove, L. van Hoye, M. Sebrechts, F. de Turck, and B. Volckaert, “FUSE: a Microservice Approach to Cross-domain Federation Using Docker Containers.” [Online]. Available: <https://wiki.confine-project.eu/oml:start>
- [17] I. Braun, M. Hoffmann, and R. Mörseburg, *IMPLEMENTATION OF A WEB-BASED AUDIENCE RESPONSE SYSTEM AS MICROSERVICE APPLICATION VS. MONOLITHIC APPLICATION*. 2019.
- [18] Y. Wang and D. Ma, “Developing a Process in Architecting Microservice Infrastructure with Docker, Kubernetes, and Istio,” Nov. 2019, [Online]. Available: <http://arxiv.org/abs/1911.02275>
- [19] A. Duc Pham Abstract Author Phamand A. Duc, “DEVELOPING BACK- END OF A WEB APPLICATION WITH NESTJS FRAMEWORK Title of publication Developing back-end of a web application with NestJS framework Case: Integrify Oy’s studentmanagement system,” 2020.
- [20] F. Ponce Mella, G. Márquez, H. Astudillo, and F. Ponce, “Migrating from monolithic architecture to microservices: A Rapid Review,” 2019. [Online]. Available: <https://www.researchgate.net/publication/335716451>
- [21] D. Sutohap, D. Surya Dwi Putra, S. Margita, and M. Subhana, “PERANCANGAN SISTEM DELIVERY UNTUK PT. JEJARING HIJAU INDONESIA DENGAN ARSITEKTUR MICROSERVICE,” *JURNAL ALGOR*, vol. II, no. 2, 2021, [Online]. Available: <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- [22] R.R Hidayat and D. Jatikusumo, “MONITORING SISTEM BERBASIS WEB KEAMANAN TRANSAKSI PENGIRIMAN UANG PADA PENYELENGGARA TRANSFER DANA DENGAN MENGGUNAKAN PERATURAN BANK INDONESIA ANTI PENCUCIAN UANG & PENCEGAHAN PENDANAAN TERORISME,” vol. 12, no. 1, 2019, [Online]. Available: <https://stt-pln.e-journal.id/petir/article/view/415>
- [23] Y. Devianto and S. Dwiasnati, “Rancang Bangun Web Portal Berita Sebagai Sumber Informasi Berita Tentang Pertanian,” vol. 8, no. 2, 2021, [Online]. Available: <http://jurnal.mdp.ac.id/jatiasi@mdp.ac.id/12>

KERTAS KERJA

Ringkasan

Dalam memulai pengembangan perangkat lunak, salah satu poin terpenting adalah menentukan arsitektur teknologi sesuai dengan analisa kebutuhan produk agar nantinya dapat berjalan selaras bersamaan visi produk tersebut[11]. Namun, seiring berjalannya waktu dengan user dan data yang semakin bertambah, perlu ada perombakan arsitektur kembali dengan membangun pondasi yang lebih kokoh. Hal ini dialami oleh perusahaan startup sosial Campaign.com.

Campaign.com merupakan sebuah aplikasi untuk mengambil aksi sosial yang dapat menyalurkan donasi dari sponsor. Desain sistem yang dibangun di campaign.com masih menggunakan arsitektur monolitik. Antarmuka pengguna, pemrosesan logika, dan akses data digabungkan menjadi satu program dan ditempatkan dalam satu basis data, namun hal ini membuat aplikasi sering mengalami bugs dan server downtime. Dengan masalah yang sedang dihadapi ini penulis mencoba melakukan eksperimen dengan merancang dan menganalisa sistem arsitektur backend.

Tujuan yang ingin dicapai dari perancangan arsitektur ini adalah mendeskripsikan proses penyederhanaan dari sebuah sistem arsitektur *backend* microservice agar mudah dalam pemeliharaan sistem dan penggunaan algoritma Docker yang akan membuat sistem menjadi efisien terhadap penulisan kode.