

**Implementasi Arsitektur Enterprise Pola Finansial pada Aplikasi Berbasis
Microservices**

Arsitektur Enterprise (C)

Supangat, M.Kom., ITIL., COBIT.



Oleh

Novandika Yuda Pamungkas

1461900044

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS 17 AGUSTUS 1945 SURABAYA

2021

DAFTAR ISI

BAB 1 PENDAHULUAN	3
1.1 Latar Belakang	3
BAB II TINJAUAN PUSTAKA	4
2.1 Arsitektur Enterprise	4
2.2 Arsitektur Enterprise Pola Finansial	Error! Bookmark not defined.
2.3 Microservices	Error! Bookmark not defined.
BAB III PEMBAHASAN.....	9
BAB IV KESIMPULAN & SARAN.....	10

BABI PENDAHULUAN

1.1 Latar Belakang

Arsitektur Enterprise (AE) memberikan gambaran tingkat tinggi mengenai sebuah enterprise dari segi usaha, sistem IT-nya, bersama korelasinya . ilustrasi tingkat tinggi tersebut meliputi deksripsi mengenai tujuan, visi, taktik, info entitas, proses usaha, orang-orang yang terlibat, struktur organisasi, sistem di software, infrastruktur teknologi atau perangkat keras. di beberapa tahun belakangan ini sedang terjadi banyak revolusi arsitektur teknologi pada berbagai enterprise dari arsitektur monolitik ke arsitektur microservices. di tahun 2020, telah terdapat 78% organisasi yang menggunakan microservices, dan 61% di antaranya baru memakai microservices semenjak 1-5 tahun belakangan ini . Keterbatasan arsitektur monolitik pada skalabilitas serta ketergantungan bertenaga antar modulnya membentuk beberapa enterprise kesulitan untuk mengimbangi kompleksitas bisnisnya yang makin meningkat dari tahun ke tahun dan beralih ke microservices . Subjek penelitian yang digunakan pada makalah ini adalah AE pola finansial yang sudah tersedia sebelumnya di buku . AE pola finansial ini merupakan generalisasi dari AE berbagai departemen finansial di sebuah enterprise. AE pola finansial ini bertujuan untuk mengelola aset dan ekuitas milik enterprise.pada makalah ini, disajikan deskripsi tentang bagaimana proses pengimplementasian AE pola finansial sebagai perangkat lunak berbasis microservices. Artikel ini diharapkan bisa memberikan referensi terkait pengimplementasian AE di software berbasis microservices.

BAB II

TINJAUAN PUSTAKA

2.1 Arsitektur Enterprise

Bidang arsitektur enterprise beserta skema lapisan yang akan digunakan dalam makalah ini akan merujuk kepada kerangka kerja TOGAF, yakni: ranah arsitektur bisnis, ranah arsitektur data atau informasi, ranah arsitektur perangkat lunak, serta ranah arsitektur teknologi. TOGAF dipilih karena struktur ranah arsitekturnya lebih sederhana serta praktis dipahami. Meski begitu, TOGAF sudah bisa diklaim setara menggunakan kerangka kerja lainnya . TOGAF sendiri dibuat oleh konsorsium tingkat internasional serta mendorong pengembangan standar teknologi terbuka dan netral.

Ini merupakan 4 skema lapisan AE pada kerangka kerja TOGAF .

A. Ranah Arsitektur bisnis

Lapisan ini menggambarkan struktur serta proses bisnis di organisasi seperti: strategi penjualan produk atau jasa, struktur organisasi, dan sebagainya. Lapisan ini sangat krusial sebagai akibatnya lapisan-lapisan yang lain bergantung di lapisan ini.

B. Ranah Arsitektur Data

Lapisan ini menggambarkan lebih jelasnya data yang dipergunakan pada proses usaha seperti: bentuk data, relasi data, format data, dan sebagainya.

C. Ranah Arsitektur aplikasi

Lapisan ini menggambarkan segala aspek yang berkaitan menggunakan aplikasi untuk mendukung proses bisnis.

D. Ranah Arsitektur Teknologi

Lapisan ini menggambarkan infrastruktur teknologi yang digunakan untuk menjalankan aplikasi dalam rangka mendukung proses bisnis.

2.2 Arsitektur Enterprise Pola Finansial

TOGAF menggunakan pola arsitektur enterprise sebagai blok-blok penyusun dalam AE untuk menyelesaikan beberapa masalah yang berulang . Akan tetapi, TOGAF tidak menyediakan lebih lanjut pola-pola tersebut. Karena hal tersebut, pola yang dipergunakan di

makalah ini akan dari asal buku Enterprise Architecture Pattern . Kebetulan pola-pola pada buku tadi juga dibangun di atas skema lapisan TOGAF.

Desain yang akan diimplementasikan pada penelitian ini adalah pola finansial. Finansial merupakan cabang ilmu yang mempelajari mengenai sebab-akibat sebuah aset terhadap individu atau organisasi dalam jangka waktu pendek juga panjang. Pola finansial ini dapat diimplementasikan di banyak sekali perusahaan, baik besar juga kecil. Pola tersebut digunakan untuk membantu perusahaan mengelola aset beserta ekuitasnya .

Berikut merupakan beberapa persoalan finansial yang dialami banyak organisasi :

- Tagihan tersebut dibayar terlalu lama, sehingga membuat perusahaan melewatkan potongan pembayaran jangka pendek atau bahkan terkena denda sehingga melebihi batas pembayaran.
- Manajemen informasi finansial secara manual sangat melelahkan.
- Terjadi banyak kesalahan pada laporan finansial tersebut.

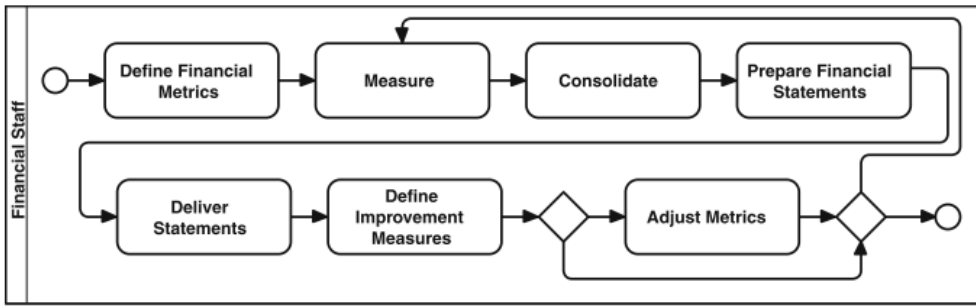
berdasarkan permasalahan di atas, berikut ini ialah AE pola finansial dari 3 perspektif.

- **Perspektif holistik**

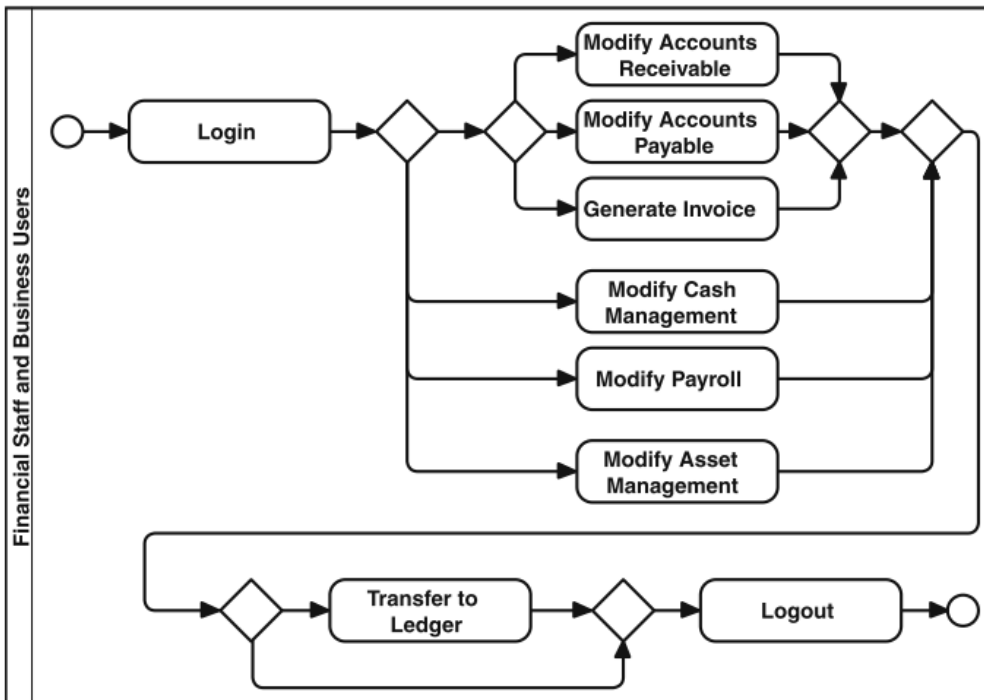
Perspektif keseluruhan merupakan garis besar dari 3 ranah arsitektur yang masing-masingnya dipisahkan memakai garis putus-putus. di lapisan paling atas termuat ranah arsitektur bisnis yang berupa aktor bersama proses bisnisnya. di lapisan tengah termuat ranah arsitektur aplikasi. di lapisan paling bawah termuat ranah arsitektur teknologi. lebih jelasnya tiap lapisan akan dijelaskan lebih lanjut di perspektif selanjutnya.

- **Perspektif Proses bisnis**

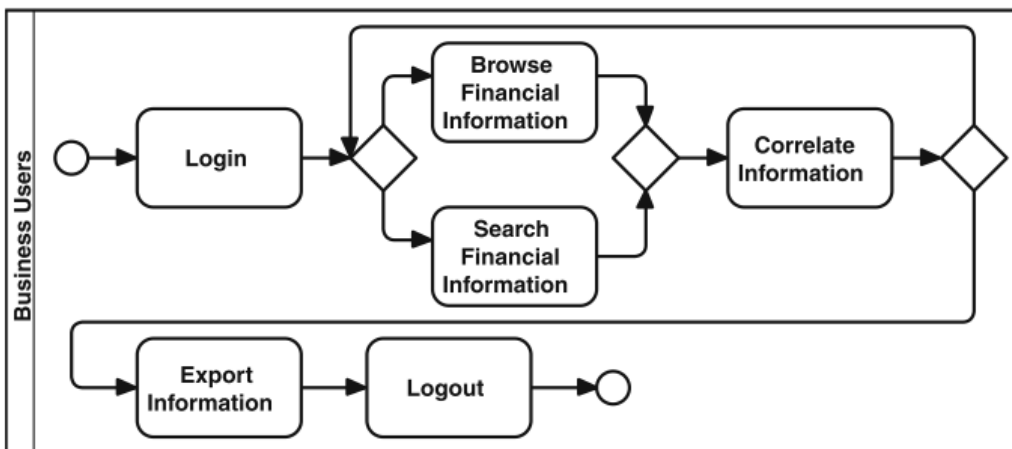
Perspektif ini membuat proses bisnis di pola finansial secara lebih mendetail. pada Gambar 2.2.1 termuat proses bisnis analisis serta perencanaan. pada Gambar 2.2.2 termuat proses bisnis masukan informasi finansial. pada Gambar 2.2.3 termuat proses bisnis baca berita finansial.



Gambar 2.2.1 Proses bisnis analisis serta perencanaan

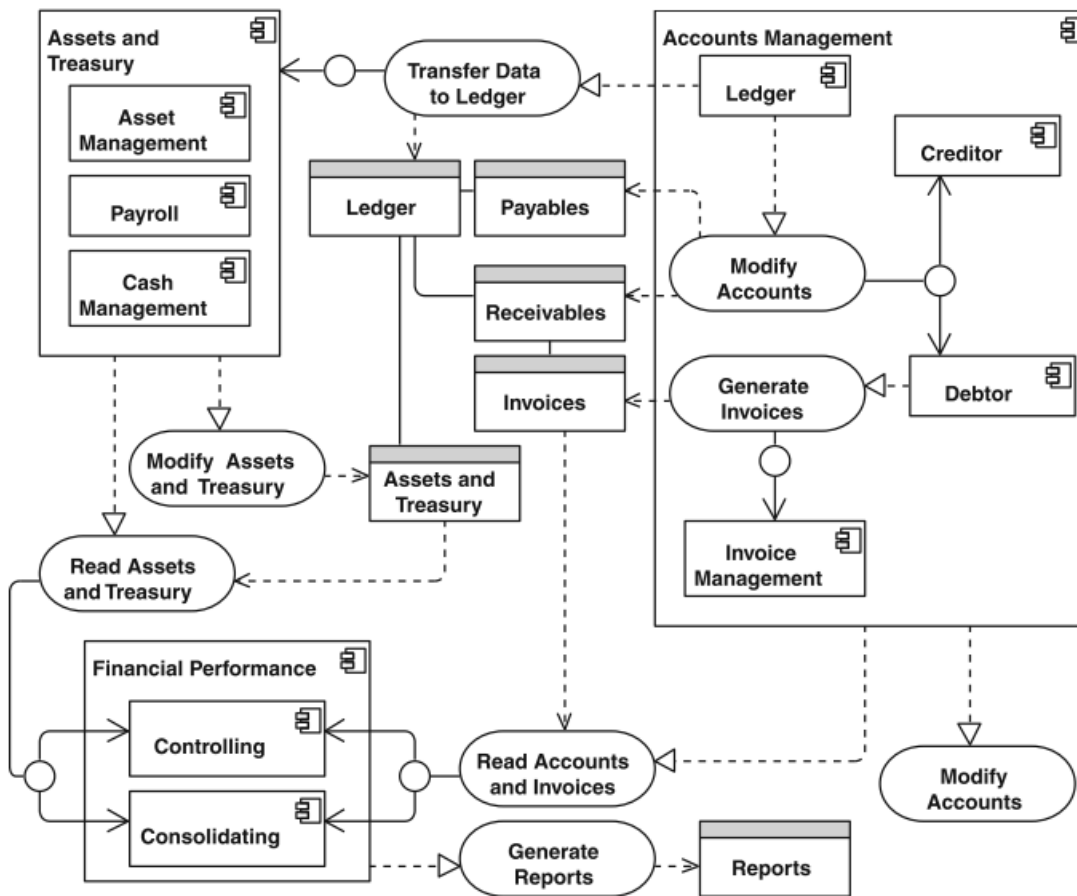


Gambar 2.2.2 Proses masukan informasi finansial



Gambar 2.2.3 Proses baca informasi finansial

- **Perspektif Data dan Aplikasi**



Gambar 2.2.4 Perspektif data dan aplikasi

Perspektif ini memuat seluruh aliran pada aplikasi serta skema data yang digunakan. Dalam Gambar 2.2.4, terdapat 3 kelompok aplikasi yakni:

- **Asset & Treasury,**

Dalam kelompok asset and treasury terdapat 3 aplikasi yang dapat mengakses 3 Fasilitas serta 1 basis data. aplikasi tersebut yakni: asset management, payroll, cash management. 3 Fasilitas tersebut yakni: modify asset and treasury, read asset and treasury, transfer to ledger. Ke 1 basis data tersebut yakni: asset and treasury.

- **Accounts Management,**

Kelompok accounts management memiliki 4 aplikasi yang dapat mengakses 4 Fasilitas serta 3 basis data. 4 aplikasi tersebut yakni: ledger, creditor, debtor, invoice management. 4

Fasilitas ialah: transfer data to ledger, modify accounts, generate invoice, read accounts and invoice. Tiga basis data yakni: payables, receivable, invoices.

➤ **Financial Performance**

Kelompok financial performance memiliki 2 aplikasi yang dapat mengakses 3 Fasilitas serta 2 basis data. 2 aplikasi tersebut yakni: controlling, consolidating. Tiga Fasilitas tersebut yakni: read assets and treasury, read accounts and invoices, generate reports. Satu basis data tersebut yakni: reports.

2.2 Microservices

Mengembangkan aplikasi sebagai layanan-layanan kecil yang independen memberikan beberapa keuntungan dan kerugian. Keuntungan tersebut yakni: fleksibilitas pemilihan teknologi, resiliensi lebih terhadap bug, kemudahan dalam skalabilitas, kemudahan dalam mengembangkan masing-masing Fasilitas . Sedangkan kerugiannya yakni: kerumitan dalam mengelola seluruh Fasilitas, sulit menjaga konsistensi terhadap data, serta risiko kegagalan komunikasi yang cukup besar. Microservices memiliki pola-pola yang sudah tersedia untuk mempermudah pengembangannya. Pola yang digunakan dalam makalah ini adalah pola saga dan pola orkestrasi. Pola saga merupakan pola yang digunakan untuk menjaga integritas data saat aplikasi melakukan transaksi yang melibatkan lebih dari 1 basis data. Pola orkestrasi merupakan pola yang menyimpan seluruh urutan aktivitas proses bisnis pada 1 Fasilitas. Selanjutnya, 1 Fasilitas tersebut mendelegasikan aktivitas-aktivitasnya kepada Fasilitas lain untuk dikerjakan dan keluarannya akan disimpan kembali pada 1 Fasilitas orkestrasi tersebut.

2.3 MERN Stack

MERN stack merupakan istilah dari susunan 4 kerangka kerja yang digunakan untuk membangun aplikasi berbasis web. Secara berurutan MERN tersusun atas: MongoDB, ExpressJS, NodeJS, dan ReactJS. MongoDB yaitu sebuah basis data yang bersifat terbuka yang berorientasi dokumen. MongoDB memiliki keunggulan pada fleksibilitas struktur data. Selanjutnya, ExpressJS merupakan kerangka kerja terbuka berbasis Javascript yang dibangun di atas platform NodeJS. Dalam MERN, ExpressJS digunakan untuk mengembangkan aplikasi bagian server. ExpressJS ini memiliki keunggulan pada proses pengembangannya yang sangat mudah dan cepat

BAB III

PEMBAHASAN

Tujuan AE pola finansial untuk mengelola aset dan ekuitas dapat dicapai setelah diimplementasikan. Untuk pembuktiannya, penulis mencoba menggunakan aplikasi untuk menyelesaikan studi kasus sederhana mengenai keuangan enterprise. Pengembangan aplikasi microservices terbukti mudah ketika mengembangkan dan memperbaiki bug pada masing-masing layanan. Hal tersebut dikarenakan masing-masing layanan pada microservices bersifat independen. Jadi, lingkup pengembangan serta perbaikan bug hanya terbatas per layanan saja.

Pengembangan aplikasi microservices terbukti sulit ketika mengelola seluruh komunikasi dan menjaga integritas data antar layanannya. Pada microservices sering terdapat beberapa duplikasi data antar layanannya. Kasus kegagalan komunikasi antar layanan dapat membuat data yang seharusnya sama antar layanan menjadi berbeda. Diperlukan kode exception error yang cukup kompleks untuk mengatasi hal tersebut.

Penggunaan MERN stack terbukti menghemat waktu pengembangan. Hal tersebut dikarenakan penggunaan basis data berorientasi dokumen yang tidak terikat dengan skema mampu menghemat waktu ketika melakukan penyesuaian struktur data. Selanjutnya ExpressJS dan ReactJS yang sama-sama berbasis bahasa Javascript juga dapat menghemat waktu untuk belajar. Selain itu, prinsip penggunaan ulang kode pada ReactJS juga sangat menghemat waktu ketika menulis kode program.

BAB IV

KESIMPULAN & SARAN

Berdasarkan penelitian yang sudah dilakukan, terdapat 3 tahapan utama dalam mengimplementasikan AE menjadi aplikasi berbasis microservices.

- Tahap perancangan AE atau pola AE

Dalam tahap ini dilakukan proses perancangan AE atau pola AE yang akan diimplementasikan. Tergantung situasi dan kondisinya, perancangan ini dapat berupa membuat AE atau pola AE baru ataupun memodifikasi AE atau pola AE yang sudah tersedia. AE ataupun pola AE yang dirancang di tahap ini setidaknya harus memiliki gambaran yang setara dengan pola AE finansial yang digunakan pada makalah ini.

- Tahap pengembangan aplikasi

Dalam tahap ini dilakukan proses pengembangan aplikasi berdasarkan AE atau pola AE yang sudah dirancang sebelumnya. Terdapat 3 hal utama yang harus dikembangkan dalam tahap ini, yakni: basis data, API layanan, dan antarmuka. Arsitektur microservices ini diterapkan pada API layanan dan berdampak langsung pada struktur basis data. Pada praktiknya, aplikasi berbasis microservices ini memiliki beberapa pola yang dapat digunakan untuk mengatasi masalah yang sering terjadi pada proses pengembangannya.

- Tahap pengujian aplikasi

Dalam tahap ini dilakukan proses pengujian dari aplikasi yang sudah dikembangkan pada tahap sebelumnya. Pengujian ini dilakukan untuk memastikan bahwa seluruh fungsionalitas sudah berjalan sesuai dengan yang diharapkan.

SARAN

Artikel ini hanya menyajikan mengenai deskripsi proses implementasi AE pola finansial menjadi aplikasi berbasis microservices. Finansial sendiri merupakan domain enterprise yang sudah ada sejak ratusan tahun yang lalu. Adapun saat ini terdapat banyak domain enterprise. Maka dari itu, untuk menyempurnakan penelitian ini perlu penelitian lebih lanjut mengenai penerapan AE pada domain lain menjadi aplikasi berbasis microservices.

DAFTAR PUSTAKA

- Supangat and Amna, A. R. (2019) 'Pemanfaatan Fitur Analisis Data Menggunakan K-Means Cluster Dalam Point of Sales (POS)', *Teknika*, 8(2), pp. 97–102. doi: 10.34148/teknika.v8i2.157.
- T. Tamm, P. B. Seddon, G. Shanks, and P. Reynolds, "How does enterprise architecture add value to organisations?," *Commun. Assoc. Inf. Syst.*, vol. 28, no. 1, pp. 141–168, 2011, doi: 10.17705/1cais.02810.
- C. M. Pereira, "Enterprise Architecture : Business and IT Alignment," pp. 1344–1345, 2005.
- T. Perroud and R. Inversini, *Enterprise Architecture Patterns*. 2013.
- B. D. Rouhani, M. N. Z. R. Mahrin, F. Nikpay, R. B. Ahmad, and P. Nikfard, "A systematic literature review on Enterprise Architecture Implementation Methodologies," *Inf. Softw. Technol.*, vol. 62, no. 1, pp. 1–20, 2015, doi: 10.1016/j.infsof.2015.01.012.
- M. Loukides and S. Swoyer, "Microservices Adoption in 2020 – O'Reilly," Jul. 15, 2020. <https://www.oreilly.com/radar/microservices-adoption-in-2020/> (accessed Jun. 06, 2021).
- K. Indrasiri and P. Siriwardena, *Microservices for the Enterprise: Designing, Developing, and Deploying*. 2018.
- J. Lewis and M. Fowler, "Microservices," 2014. <https://martinfowler.com/articles/microservices.html> (accessed May 10, 2021).
- S. Newman, *Building Microservices*. 2015.
- R. Winter and R. Fischer, "Essential Layers , Artifacts , and Dependencies of Enterprise Architecture," 2006.
- "The TOGAF Standard, Version 9.2 - Architecture Patterns." <https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap22.html> (accessed Jun. 06, 2021).
- C. Richardson, "Developing event-driven microservices with event sourcing and CQRS svcc 2015.key."
- C. Richards, Chris Richards, vol. 2018, no. March. 2018.
- M. P. Widodo, "Pengembangan Aplikasi Pelaporan Progress-Plan- Problem untuk Manajemen Tugas dan Penentuan OKR di Krafthaus Indonesia."
- A. Samikshya, "MERN STACK WITH MODERN WEB PRACTICES-Developers Connecting Application," 2020.
- S. Aggarwal and J. Verma, "Comparative analysis of MEAN stack and MERN stack," *Int. J. Recent Res. Asp.*, vol. 5, no. 1, pp. 127–132, 2018.

- I. Ilin, A. Levina, and O. Iliashenko, "Enterprise architecture approach to mining companies engineering," *MATEC Web Conf.*, vol. 106, 2017, doi: 10.1051/mateconf/201710608066.
- M. O. Riku and D. B. Setyohadi, "Strategic plan with enterprise architecture planning for applying information system at PT. Bestonindo Central Lestari," *2017 5th Int. Conf. Cyber IT Serv. Manag. CITSM 2017*, 2017, doi: 10.1109/CITSM.2017.8089274.
- P. Sidiq and D. Bhakti, "Architecture Planning Information Manufacture Systems Using Enterprise Architecture Planning At PT. Tin Tin," no. January 2020, 2020, doi: 10.4108/eai.11-7-2019.2298090.
- A. Bucchiarone, N. Dragoni, S. Dustdar, S. T. Larsen, and M. Mazzara, "From Monolithic to Microservices: An Experience Report from the Banking Domain," *IEEE Softw.*, vol. 35, no. 3, pp. 50–55, 2018, doi: 10.1109/MS.2018.2141026.
- C. Y. Fan and S. P. Ma, "Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report," *Proc. - 2017 IEEE 6th Int. Conf. AI Mob. Serv. AIMS 2017*, pp. 109–112, 2017, doi: 10.1109/AIMS.2017.23.
- W. Hasselbring and G. Steinacker, "Microservice architectures for scalability, agility and reliability in e-commerce," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. Work. ICSAW 2017 Side Track Proc.*, pp. 243–246, 2017, doi: 10.1109/ICSAW.2017.11.
- H. Suryotrisongko, D. P. Jayanto, and A. Tjahyanto, "Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot," *Procedia Comput. Sci.*, vol. 124, pp. 736–743, 2017, doi: 10.1016/j.procs.2017.12.212.
- M. Wu, X. Ding, and R. Hou, "Design and implementation of B2B E-commerce platform based on microservices architecture," *ACM Int. Conf. Proceeding Ser.*, pp. 30–34, 2019, doi: 10.1145/3339363.3339369.

83%

Konten unik

17%

Konten yang dijiplak

✓ COMPLETED

100%