

**Implementasi Arsitektur Microservices Backend-for-Frontend  
(BFF) pada Sistem *E-Commerce* Menggunakan NestJS**

**Tugas Akhir 1**



**Oleh: Sherly Septiani**

**NIM: 672022259**

**Pembimbing: Yeremia Alfa Susetyo, S.Kom., M.Cs.**

**Program Studi Teknik Informatika Fakultas Teknologi Informasi**

**Universitas Kristen Satya Wacana Salatiga**

**Juli 2025**

## **Daftar Isi**

<b>1. Latar Belakang</b>	<b>3</b>
<b>2. Rumusan Masalah</b>	<b>4</b>
<b>3. Tujuan dan Manfaat Penelitian</b>	<b>4</b>
<b>4. Tinjauan Pustaka</b>	<b>5</b>
<b>5. Metode Penelitian</b>	<b>6</b>
<b>Daftar Pustaka</b>	<b>9</b>

## 1. Latar Belakang

Dengan kemajuan teknologi saat ini, kehidupan manusia telah mengalami banyak perubahan yang signifikan, termasuk dalam cara mereka berinteraksi dengan internet dan perangkat digital. Sejak kemunculan dan popularitas *smartphone*, peran internet menjadi jauh lebih besar di berbagai aspek kehidupan manusia [2]. Fenomena ini terbukti dengan masifnya jumlah situs *website* yang kini menjadi salah satu layanan digital yang tak luput dari revolusi internet. Menurut *Website Statistic Report* 2025, terdapat sekitar 1,13 miliar situs *website* di seluruh dunia yang terdaftar di internet [3]. Situs-situs tersebut mencakup berbagai jenis aspek dalam kehidupan manusia sehari-hari, mulai dari portal berita, media sosial yang telah menjadi bagian penting dari masyarakat, serta toko online yang mulai naik daun semenjak adanya *e-commerce*. Mayoritas situs *website* tersebut dapat diakses baik melalui smartphone maupun perangkat lainnya, seperti komputer, desktop, laptop, smartphone, tablet, konsol game, ataupun peralatan rumah tangga. Semakin banyak orang menggunakan beragam perangkat untuk menjelajahi web, dengan ukuran layar dan resolusi yang cukup bervariasi [1]. Hal ini menjadi bukti bahwa pengguna dapat mengakses internet dari layar yang memiliki variasi ukuran dan resolusi perangkat.

Perkembangan *website* menjadi bukti bahwa internet memiliki peran yang cukup signifikan bagi manusia. Dengan kemajuan teknologi, manusia tidak hanya terbatas pada satu jenis perangkat saja untuk mengakses internet, berbagai perangkat dapat dimanfaatkan untuk mengakses *website*, aplikasi, dan berbagai sistem online lainnya. Hal inilah yang mendorong kebutuhan akan sistem yang mampu beradaptasi dengan variasi perangkat yang memiliki karakteristik dan keterbatasan yang berbeda. Dalam konteks *website*, misalnya, diperlukan adanya perkembangan terbaru guna beradaptasi dengan berbagai variasi perangkat yang ada [5]. Hal ini tidak terbatas pada tampilan antarmuka saja. Seiring dengan meningkatnya kompleksitas dan skala aplikasi modern, diperlukan sistem yang memiliki tingkat keandalan dan ketangguhan tinggi agar dapat berfungsi secara optimal [4]. Sistem yang mampu melayani berbagai permintaan secara optimal tanpa mengorbankan performa untuk meningkatkan kepuasan pengguna. Beragam perangkat dan kompleksitas sistem inilah yang menjadi tantangan baru dalam pengembangan layanan digital.

Dalam suatu perangkat, arsitektur perangkat lunak memungkinkan pengembangan sistem yang kompleks [6]. Sama seperti aplikasi dan *website*, arsitektur perangkat lunak juga terus berkembang seiring dengan kemajuan teknologi. Saat ini, arsitektur microservices muncul sebagai pola untuk mengatasi pembangunan aplikasi yang semakin kompleks [7]. Arsitektur microservices ini dianggap fleksibel pada tiap tim yang memungkinkan untuk menggunakan bahasa pemrograman, database, dan kerangka kerja yang berbeda berdasarkan kebutuhan spesifiknya [7].

Salah satu pendekatan arsitektur microservices yang dirancang untuk menangani kebutuhan penyesuaian antara sistem back-end pada jenis antarmuka adalah backend-for-frontend (BFF) [8]. Pola ini mulai banyak diterapkan seiring dengan berkembangnya pendekatan arsitektur microservices dalam merancang sistem back-end yang relevan [8]. Dalam pola backend-for-frontend, setiap jenis antarmuka memiliki back-end tersendiri yang disesuaikan dengan kebutuhan spesifik dari antarmuka. Contohnya, data yang dikirimkan untuk perangkat mobile lebih ringkas dan ringan dibandingkan data yang dikirimkan untuk web admin yang lebih kompleks. Backend-for-Frontend membantu mengoptimalkan komunikasi antara klien dan back-end tanpa membebani satu sisi, serta memberi fleksibilitas bagi tim pengembang untuk melakukan perubahan tanpa mengganggu keseluruhan sistem. Kehadiran backend-for-frontend dapat menjadi solusi yang tepat untuk mengatasi tantangan dalam pengembangan sistem modern yang melayani berbagai jenis perangkat dengan efisien.

Dengan adanya aplikasi yang semakin kompleks, perangkat yang varian, serta kebutuhan tampilan dan data yang perlu disesuaikan, maka arsitektur microservices menjadi solusi yang mampu menyelesaikan tantangan tersebut. Arsitektur microservices dianggap fleksibel untuk mengembangkan aplikasi berskala besar yang membantu tim menyesuaikan berdasarkan kebutuhan spesifiknya. Salah satu pola yang ditawarkan sebagai solusi secara khusus adalah backend-for-frontend. Backend-for-Frontend memungkinkan penyesuaian kebutuhan data, adaptasi perangkat, serta merespons kompleksitas sistem secara tepat.

## 2. Rumusan Masalah

Dalam pengembangan sistem modern, belum ada mekanisme penyesuaian pengiriman data pada sistem *e-commerce* berbasis microservices yang mampu beradaptasi dengan kebutuhan dan keseragaman dari berbagai jenis perangkat. Hal ini menyebabkan adanya beban logika yang tinggi di sisi klien dan inefisiensi pengelolaan data, sehingga diperlukan adanya penerapan arsitektur microservices backend-for-frontend dengan menggunakan NestJS sebagai solusi untuk menyesuaikan data yang dikirimkan oleh back-end.

## 3. Tujuan dan Manfaat Penelitian

Penelitian ini bertujuan untuk mengimplementasikan arsitektur backend-for-frontend pada sistem *e-commerce* menggunakan NestJS, dengan fokus untuk menyesuaikan pengiriman data dan adaptasi jenis perangkat.

Secara khusus, manfaat yang ingin dicapai dari penelitian ini adalah:

- Meningkatkan kemampuan sistem untuk beradaptasi pada multi-perangkat, sehingga tampilan dan data yang diterima oleh pengguna dapat disesuaikan dengan perangkat yang digunakan
- Mempercepat proses pengembangan produk dengan memisahkan logika front-end dan back-end secara terstruktur
- Mengurangi kompleksitas logika yang berbeda-beda pada sisi klien
- Meningkatkan efisiensi pengelolaan dan pengiriman data pada penyajian data ke berbagai platform
- Mendukung kebutuhan bisnis *e-commerce* yang membutuhkan fleksibilitas tinggi

#### **4. Tinjauan Pustaka**

Tinjauan Pustaka, memuat penelitian terdahulu dan penjelasan ringkas tentang teori/metode/praktek-praktek yang relevan dengan topik untuk menunjang perancangan atau penelitian yang dibuat.. Fakta-fakta yang dikemukakan sejauh mungkin diambil dari sumber aslinya. Semua sumber yang dipakai harus disebutkan nama penulis dan tahun penerbitan.

Dalam penelitian berjudul "*Implementation of microservices Architecture on E-Commerce Web Service*", dijelaskan bahwa *e-commerce* dimanfaatkan sebagai sarana untuk melakukan transaksi jual beli produk secara daring melalui layanan berbasis internet. Dengan meningkatnya pertumbuhan *e-commerce*, maka diperlukan layanan yang memiliki ketersediaan tinggi dan mudah untuk dikembangkan dan dipelihara. Arsitektur microservices mampu mengatasi masalah tersebut. Evaluasi menunjukkan bahwa arsitektur microservices dapat meningkatkan ketahanan dan fleksibilitas sistem. Ketika dapat terdapat gangguan pada satu layanan, layanan lain masih tetap berfungsi secara stabil. Sehingga arsitektur ini dianggap relevan untuk diterapkan dalam sistem *e-commerce* untuk meningkatkan performa dan skalabilitas. Namun, penelitian ini belum membahas penyesuaian respons data pada tiap perangkat yang memiliki antarmuka berbeda dalam penerapan arsitektur microservices.

Sementara itu, pada penelitian berjudul "*Backend for Frontend in microservices*" menegaskan bahwa pendekatan pola backend-for-frontend merupakan pemisahan antara front-end dan back-end untuk menghilangkan kemungkinan konflik dan meningkatkan performa serta konsistensi sistem. Penelitian ini menunjukkan bahwa arsitektur backend-for-frontend dapat meningkatkan performa dan pengalaman pengguna. Backend-for-Frontend mampu memberikan pengalaman pengguna yang mulus dengan menggunakan back-end untuk memenuhi kebutuhan pada antarmuka perangkat yang berbeda dan mampu memanfaatkan sumber daya yang lebih efisien untuk setiap penggunaan. Dengan memindahkan logika pemrograman ke lapisan backend-for-frontend yang tidak berjalan di browser dan memfilter data yang akan dikirim oleh backend-for-frontend, maka beban di browser akan berkurang. Namun, penggunaan backend-for-frontend masih menghadapi

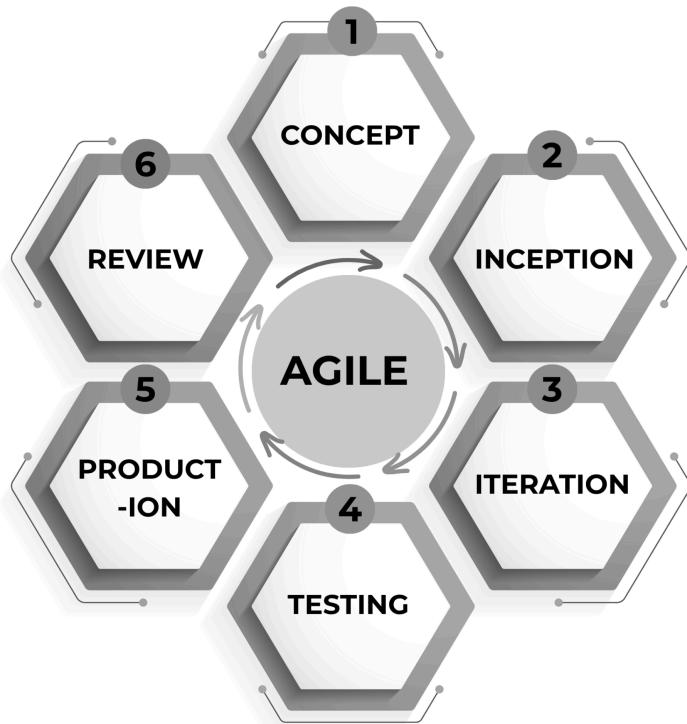
tantangan seperti duplikasi logika dan kompleksitas pengelolaan, sehingga diperlukan implementasi yang lebih terstruktur untuk mengoptimalkan respons data antarmuka dalam sistem *e-commerce*.

Penelitian berjudul “*Development of microservices Architecture with RESTful API Gateway using backend-for-frontend Pattern in Higher Education Academic Portal*” menjelaskan bahwa backend-for-frontend merupakan implementasi dari API Gateway karena setiap front-end mendapatkan backend-for-frontend nya sendiri, sehingga front-end dapat menyesuaikan backend-for-frontend agar sesuai dengan setiap kebutuhan klien. Pola backend-for-frontend pada arsitektur microservices dipilih karena konsumsi API ditujukan untuk aplikasi mobile, sehingga membutuhkan lapisan back-end yang mampu menyesuaikan dengan kebutuhan klien. Pendekatan ini memungkinkan setiap perangkat klien memiliki gateway sendiri untuk memproses dan menyesuaikan permintaan data sesuai dengan kebutuhan tampilan dan interaksi. Penerapan ini dianggap efektif untuk menangani variasi kebutuhan antar platform, meningkatkan performa, dan menjaga modularitas sistem backend. Meski penelitian ini berhasil menunjukkan penerapan arsitektur microservices backend-for-frontend yang menyederhanakan interaksi antara aplikasi mobile dan backend, tapi belum memberikan kebutuhan spesifik pada platform lainnya selain mobile.

Berdasarkan penelitian terdahulu mengenai arsitektur microservices Backend-for-Frontend, dapat disimpulkan bahwa pendekatan arsitektur microservices dan pola backend-for-frontend memiliki potensi besar dalam meningkatkan performa, fleksibilitas, dan pengalaman pengguna pada sistem yang terdistribusi seperti *e-commerce*. Maka, penelitian ini dibuat untuk mengimplementasikan arsitektur backend-for-frontend pada sistem *e-commerce* menggunakan NestJS untuk menyesuaikan pengiriman data dan adaptasi perangkat.

## 5. Metode Penelitian

Penelitian ini dirancang untuk mengimplementasikan arsitektur *microservices* backend-for-frontend dengan *framework* NestJS pada sistem *e-commerce*. Proses penelitian ini dilakukan dengan melalui beberapa tahapan sistematis yang bertujuan untuk merancang dan membangun sistem agar dapat memenuhi kebutuhan penyesuaian data pada platform klien. Metodologi yang digunakan dalam merancang mengimplementasikan ini adalah *Agile*.



**Gambar 1.** Diagram Metodologi Agile

*Agile Methodology* merupakan framework manajemen proyek yang memisahkan suatu projek dalam berbagai fase. Penelitian ini menggunakan metode *Agile* untuk mendukung proses pengembangan sistem yang fleksibel dan adaptif. Gambar 1 menunjukkan urutan atau tahap dalam mengembangkan sistem dengan menggunakan *Agile*. Yang pertama, *Concept*, mengidentifikasi perumusan masalah dan tujuan dari penelitian. Tahap *Inception* mencakup perancangan arsitektur microservices backend-for-frontend dengan menggunakan NestJS. Setelah tahap *Inception*, dilakukan pembangunan fitur-fitur dari layanan yang telah dirancang, serta integrasi dengan backend-for-frontend pada tahap *Iteration*. Tahap *Release* meliputi pengujian dan deployment agar sistem siap diuji dan dievaluasi. Tahap *Maintenance* diisi dengan pemantauan dan peningkatan sistem berdasarkan hasil pengujian. Terakhir, tahap *Retirement*, evaluasi sistem untuk meningkatkan lebih lanjut serta didokumentasikan.

Tahap awal, dengan melakukan identifikasi permasalahan yang menjadi dasar dari penelitian, yaitu kebutuhan sistem multi-platform yang mampu menyesuaikan format data berdasarkan jenis klien. Dengan menggunakan studi literatur dan pengamatan sistem yang serupa, dapat disimpulkan bahwa arsitektur microservices backend-for-frontend dapat menjadi solusi yang relevan.

Tahap kedua mulai berfokus pada perancangan awal sistem yang akan dibangun. Pada tahap ini, arsitektur microservices backend-for-frontend mulai dirancang dengan memisahkan layanan seperti Order, Product, User, dan layanan lainnya. Setelah itu, perancangan backend-for-frontend diterapkan untuk melayani kebutuhan dari klien dengan format response yang berbeda.

Pada tahap ketiga, dilakukan proses pengembangan secara bertahap sesuai dengan perancangan awal. Dimulai dengan membangun layanan inti yang dibutuhkan, serta melakukan integrasi dengan backend-for-frontend. Setiap iterasi dapat menghasilkan fitur yang nantinya akan diuji dan ditingkatkan pada pengembangan selanjutnya.

Setelah sistem dibangun dan diuji, sistem disiapkan untuk digunakan secara menyeluruh. Pengujian ini dilakukan dengan menggunakan Postman untuk memastikan bahwa *endpoint* yang dibangun sesuai dengan yang diharapkan. Setelah itu dilakukan integrasi antar layanan untuk memastikan bahwa data dapat tersalurkan dengan baik melalui backend-for-frontend.

Tahap selanjutnya mencakup pemantauan hasil pengujian dan identifikasi permasalahan dalam tahap sebelumnya. Berdasarkan temuan tersebut, dilakukan perbaikan pada struktur logika backend-for-frontend dan optimalisasi komunikasi antar layanan agar sistem berjalan lebih stabil dan efisien.

Bagian akhir dari proses pengembangan adalah evaluasi sistem untuk menilai keberhasilannya dalam menyelesaikan permasalahan yang diangkat dalam penelitian. Evaluasi ini mencakup efektivitas penggunaan arsitektur microservices backend-for-frontend untuk menyesuaikan data pada berbagai jenis klien serta memudahkan mengelola layanan. Hasil dari seluruh tahapan ini didokumentasikan dalam laporan yang dapat dijadikan acuan penelitian lanjutan.

## Daftar Pustaka

- [1] Bhanarkar, N., A. Paul, dan A. Mehta. 2023. *Responsive web design and its impact on user experience*. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) 3(4): 50–55. [online]. <https://doi.org/10.48175/IJARSCT-9259>. Diakses tanggal 18 Mei 2025.
- [2] Alkhodary, S. 2022. *The evaluation of using Backend-for-Frontend in a microservices environment* (Master's thesis, Lund University). Lund University Publications. [online]. <https://lup.lub.lu.se/student-papers/search/publication/9086847>. Diakses tanggal 18 Mei 2025.
- [3] Reboot Online. 2025. *Website statistics report 2025*. Reboot Online Marketing Ltd. [online]. <https://www.rebootonline.com/website-statistics/>. Diakses tanggal 17 Mei 2025.
- [4] Shabani, I., E. Mëziu, B. Berisha, dkk. 2021. *Design of modern distributed systems based on microservices architecture*. International Journal of Advanced Computer Science and Applications (IJACSA) 12(2): 153–158. [online]. <https://doi.org/10.14569/IJACSA.2021.0120219>. Diakses tanggal 19 Mei 2025.
- [5] Kaur, J., dan I. Bharol. 2024. *The future of responsive web design: Challenges and opportunities in multi-device compatibility*. International Journal for Research in Applied Science & Engineering Technology (IJRASET) 12(IV). [online]. <https://doi.org/10.22214/ijraset.2024.60818>. Diakses tanggal 18 Mei 2025.
- [6] Razzaq, A. 2020. *A systematic review on software architectures for IoT systems and future direction to the adoption of microservices architecture*. SN Computer Science 1(350). [online]. <https://doi.org/10.1007/s42979-020-00359-w>. Diakses tanggal 19 Mei 2025.
- [7] Busi, S. P. 2025. *Understanding microservices architecture: A comprehensive guide*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology 11(1): 1440–1447. [online]. <https://doi.org/10.32628/CSEIT251112144>. Diakses tanggal 19 Mei 2025.
- [8] Falkevych, V., dan A. Lisniak. 2024. *Client state management using Backend for Frontend pattern architecture in B2B segment*. Artificial Intelligence

- 2024(2): 49–56. [online]. <https://doi.org/10.15407/jai2024.02.049>. Diakses tanggal 16 Mei 2025.
- [9] Suthendra, J. A., dan M. A. I. Pakereng. 2020. *Implementation of microservices architecture on e-commerce web service*. *ComTech: Computer, Mathematics and Engineering Applications* 11(2): 89–95. [online]. <https://doi.org/10.21512/comtech.v11i2.6453>. Diakses tanggal 20 Mei 2025.
- [10] Sharma, H., dan N. Bhat. 2022. *Backend for Frontend in microservices*. *International Research Journal of Engineering and Technology (IRJET)* 9(6): 3075–3079. [online]. <https://www.irjet.net/archives/V9/i6/IRJET-V9I6629.pdf>. Diakses tanggal 20 Mei 2025.
- [11] Senduk, F. X., X. B. N. Najoan, dan S. R. U. A. Sompie. 2023. *Pengembangan arsitektur microservices dengan RESTful API Gateway menggunakan backend-for-frontend pattern pada portal akademik perguruan tinggi*. *Jurnal Teknik Informatika* 18(1): 315–324. [online]. <https://ejournal.unsrat.ac.id/index.php/informatika>. Diakses tanggal 20 Mei 2025.