

Implementasi Microservices di Situs Web Frontend

Bendra Wardana¹, Hermanto²

¹Teknik Komputer, Politeknik Unggulan Cipta Mandiri

²Teknologi Informasi, Fakultas Teknologi Kesehatan dan Sains, Universitas Muhammadiyah Muara Bungo

¹bendwardana@gmail.com, ²hermantoti22@gmail.com

Abstract

The current website development, docotel is still implementing monolithic architecture. This is inversely proportional to the dhealth system that has implemented microservices architecture, so that when developing dhealth systems, it can be done quickly and flexibly because microservices architecture means dividing applications into smaller and interconnected services one another. In addition, microservices makes every service that is built able to use a different technology which means it is not bound by a technology and the ability to handle load better than the architecture monolithic. So it requires the implementation of microservices architecture on dhealth websites that will make dhealth websites better and more modern. The eXtreme Programming (XP) methodology is a software development methodology that is included in the modern methodology. In this methodology the satisfaction of a product owner is the most important thing because the product owner is directly involved in giving feedback to the developer regarding the website being built. The website is built with Javascript programming language and uses framework reactjs and vuejs. The website that is built will be tested with stress test and performance testing methods to find out the difference from the implementation of microservices. The results of the research and testing of the implementation of the microservices architecture show that website performance and speed improvements are almost 2x better when implemented by microservices

Keywords: Microservices, eXtreme Programming, Stress Testing, Website, Performance Test.

Abstrak

Pengembangan situs web saat ini, docotel masih menerapkan arsitektur monolitik. Hal ini berbanding terbalik dengan sistem dhealth yang telah menerapkan arsitektur microservices, sehingga Ketika pengembangan sistem dhealth dapat dilakukan dengan cepat dan fleksibel karena arsitektur microservices berarti membagi aplikasi menjadi layanan-layanan yang lebih kecil dan saling terhubung satu sama lain. Selain itu, microservices membuat setiap layanan yang dibangun dapat menggunakan teknologi yang berbeda yang berarti tidak terikat oleh suatu teknologi dan kemampuan menangani beban lebih baik dibandingkan arsitektur monolitik. Sehingga membutuhkan implementasi arsitektur microservices pada situs web dhealth yang akan membuat situs web Dinkes menjadi lebih baik dan modern. Metodologi eXtreme Programming (XP) merupakan salah satu metodologi pengembangan perangkat lunak yang termasuk ke dalam metodologi modern. Dalam metodologi ini kepuasan pemilik produk adalah hal yang paling penting terpenting karena pemilik produk terlibat langsung dalam memberikan umpan balik kepada pengembang mengenai website yang sedang dibangun. Website dibangun dengan bahasa pemrograman Javascript dan menggunakan framework reactjs dan vuejs. Website yang dibangun akan diuji dengan metode stress test dan performance testing untuk mengetahui perbedaan dari implementasi layanan mikro. Hasil dari penelitian dan pengujian terhadap implementasi arsitektur microservices menunjukkan bahwa peningkatan performa dan kecepatan website hampir 2x lipat lebih baik ketika diimplementasikan dengan microservice

Kata Kunci: Microservices, Pemrograman eXtreme, Pengujian Stres, Situs Web, Uji Kinerja.

1. Pendahuluan

Perkembangan teknologi yang semakin pesat membuat perusahaan-perusahaan berbasis teknologi dituntut untuk terus mengikuti perkembangan agar dapat bertahan dalam persaingan di industri 4.0 [1]. Menerapkan teknologi dan metode terbaru pada tahap pengembangan adalah suatu keharusan jika Anda ingin tetap bertahan dan tetap kompetitif. Hal ini dapat dilihat dari perusahaan pemula yang menggunakan banyak teknologi dan metode baru yang sebelumnya sangat jarang digunakan atau bahkan tidak pernah digunakan sama sekali. Sebagai contoh, rata-rata startup saat ini menggunakan agile methodology, karena sangat cocok dengan tujuan mereka yaitu memberikan produk lebih cepat dan mendapatkan feedback sebanyak-banyaknya dari pelanggan [2]. Begitu pula dalam pengembangan

produknya, salah satu desain arsitektur yang sering digunakan oleh startup adalah arsitektur microservice, arsitektur ini memecah bagian aplikasi menjadi beberapa layanan yang lebih kecil dan saling berhubungan [3]. Konsep layanan mikro ini biasanya diterapkan hanya untuk sisi backend saat ini [4]. Namun dengan maraknya e-commerce saat ini, konsep microservice pada bagian backend dirasa masih kurang, karena e-commerce pada umumnya akan dikunjungi oleh pelanggan yang akan membeli atau mencari suatu produk. Sehingga banyak perusahaan yang mencari cara untuk mengimplementasikan microservice di bagian frontend [5]. Pada dasarnya penerapan microservice di bagian frontend bertujuan untuk mencapai hal yang sama dengan bagian backend. Pada penelitian ini, kami akan mencoba untuk

mengimplementasikan microservice pada bagian frontend, namun tentu saja bagian frontend tidak bisa begitu saja mengimplementasikan microservice karena akan dilakukan proses migrasi terlebih dahulu dari sistem yang telah dibangun saat ini. Untuk menghindari hal-hal seperti kegagalan implementasi microservice pada sisi frontend, maka akan dibuat sebuah website yang akan mengimplementasikan microservice sebagai prototype dari arsitektur

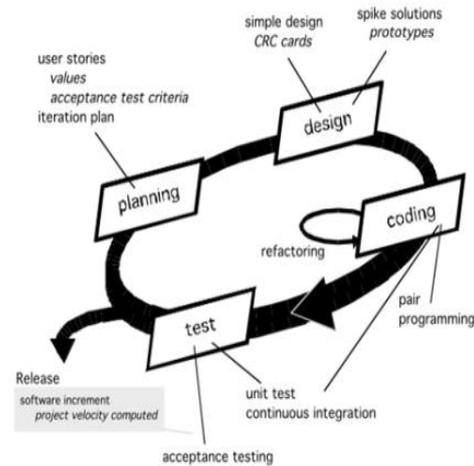
2. Metodologi Penelitian

Metodologi penelitian adalah analisis teoritis dari suatu metode atau cara. Melakukan suatu serangkaian cara atau langkah untuk memecahkan suatu masalah berdasarkan data yang diperoleh dari hasil dalam suatu penelitian. Penelitian ini berbentuk studi kasus dengan menggunakan metode penelitian research & development (R&D) serta pendekatan model extreme programming [6].

2.1. Extreme Programming

Extreme programming sendiri merupakan sebuah pendekatan yang banyak digunakan untuk pengembangan perangkat lunak. Penggunaan model ini juga karena sesuai dengan sifat aplikasi yang dikembangkan dengan cepat dan mengedepankan kepuasan pelanggan. Ada beberapa tahapan dalam XP seperti yang terlihat pada gambar, yaitu: planning/perencanaan, design/desain, coding/pengkodean, testing/pengujian [7]. Tahapan dalam metodologi perencanaan ekstrim menurut pressman meliputi : Tahap awal adalah mendengarkan dan mengumpulkan kebutuhan perangkat lunak sesuai dengan yang diinginkan pelanggan. Disini tim dan pelanggan akan bekerja sama untuk memutuskan peningkatan perangkat lunak berdasarkan cerita pelanggan. Pada penelitian ini cerita yang diberikan adalah membuat website D'health yang menarik dan informatif yang menarik, informatif, dan telah menerapkan teknologi terkini. Dari cerita tersebut akan dibuat beberapa task dimana task dengan nilai tertinggi akan diimplementasikan terlebih dahulu. Pada tahap desain dibuat pemodelan sistem berdasarkan hasil perencanaan yang telah diputuskan bersama. Pemodelan sistem yang digunakan adalah Unified Modeling Language (UML) yang terdiri dari beberapa diagram antara lain Use-Case Diagram, Activity Diagram, Component Diagram dan Deployment Diagram. Tahapan ini merupakan implementasi dari tahapan desain yang akan menghasilkan prototype perangkat lunak. Dalam pembangunan website d'health menggunakan bahasa pemrograman javascript dan mengimplementasikan arsitektur layanan mikro. Kerangka kerja yang digunakan dalam prototipe ini adalah reactjs dan vuejs dan data penyimpanan dalam bentuk javascript object notation (JSON). Tahap terakhir adalah melakukan pengujian fitur dan fungsionalitas sesuai dengan user acceptance dengan user story yang diberikan oleh pelanggan. Pengujian

pada penelitian ini menggunakan metode blackbox untuk menguji fungsionalitas dan stress testing untuk menguji performa website d'health yang telah menerapkan arsitektur microservices dengan tools WAPT.



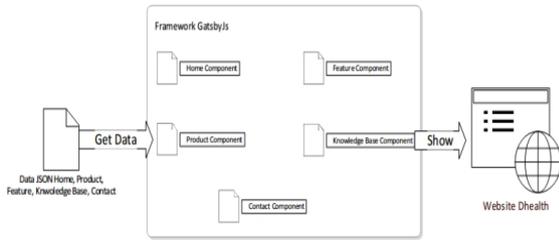
Gambar 1. Proses Tahapan *Extreme Programming*

2.2. Micro Frontends

Micro Frontends adalah konsep desain arsitektur layanan mikro yang berada di sisi depan (frontend). Istilah ini pertama kali muncul di Radar Teknologi Thought Works pada akhir tahun 2016. Saat ini sedang menjadi Tren untuk membangun sebuah aplikasi berbasis website yang memiliki banyak fitur atau yang menggunakan konsep aplikasi satu halaman masih dibangun di atas layanan yang sama. Seiring berjalannya waktu jika ada pengembangan atau maintenance akan mengalami kesulitan karena sistem yang dibangun dibangun di atas layanan yang sama. Konsep seperti ini Konsep ini disebut dengan frontend monolith. Itulah alasan mengapa arsitektur micro frontends ini dapat dibuat. Sehingga nantinya jika arsitektur ini diimplementasikan, aplikasi yang dibangun akan memiliki layanan sendiri dan dapat dikerjakan oleh tim yang berbeda dan tidak bergantung pada teknologi yang sama [8]digunakan untuk merumuskan relasi kondisional antara 2 atau lebih himpunan fuzzy [15].

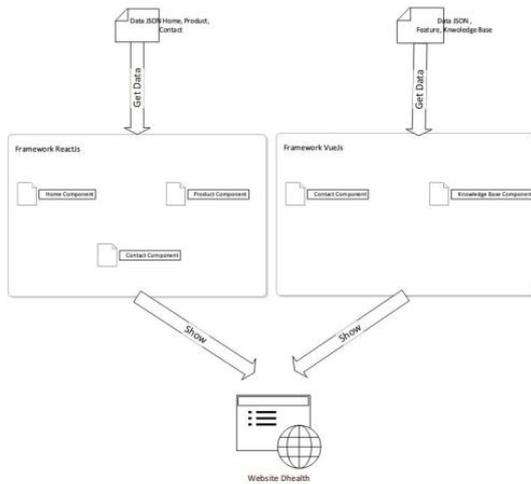
3. Hasil dan Pembahasan

Sebelum mengimplementasikan layanan mikro pada prototype, peneliti terlebih dahulu akan melakukan analisis terhadap arsitektur website yang saat ini dibangun. Arsitektur yang saat ini digunakan pada website ini adalah monolitik. Desain arsitektur ditunjukkan pada gambar 2.



Gambar 2. Arsitektur Monolitik Situs Web D'health

Situs web ini akan diimplementasikan dalam arsitektur microservices sehingga akan menghasilkan arsitektur micro frontends seperti yang terlihat pada gambar 3



Gambar 3. Arsitektur Micro Frontends di Situs Web D'health

Selanjutnya dalam implementasinya akan digunakan model pendekatan XP, Dimana website akan menekankan pada kebutuhan seorang product owner sekaligus mengimplementasikan teknologi micro frontends. Kebutuhan seorang Product Owner yang ingin segera melakukan launching dari website dhealth, membuat website yang dibangun memiliki waktu pengembangan yang relatif singkat. Maka dari itu metodologi yang digunakan dalam mengembangkan website menggunakan pendekatan XP (eXtreme Programming). XP dianggap cocok karena merupakan metodologi modern yang bersifat agile development [9]. Sebelumnya telah dijelaskan bahwa bahwa Product owner memiliki cerita yaitu membuat website D'health yang menarik dan informatif serta mengimplementasikan teknologi terkini, sehingga menghasilkan task seperti yang terlihat pada tabel 1

Tabel 1. Fungsi task Pengembangan Situs Web Dhealth

Task	Priority	Information
Home Page	1	Sebagai pelanggan, saya ingin melihat sorotan dari sistem d'health
Product page	2	Sebagai pelanggan, saya ingin melihat detail produk dari sistem dhealth

Features Page	3 Pages	Sebagai pelanggan saya ingin melihat fitur-fitur sistem dhealth secara detail
Knowledge -base page	4	Sebagai pelanggan, saya ingin melihat aturan dasar dari sistem dhealth
Contact page	5	Sebagai pelanggan, saya ingin mengetahui alamat yang dapat dihubungi
Latest Technology Implementation	6	Sebagai pelanggan, saya ingin melihat situs web dhealth yang menarik

Semua tugas dalam tabel 1 telah dirancang sebelumnya sehingga tim pengembang dapat melakukan tahap Implementasi. Pada tahap Design dibuat user interface yang memudahkan tim pengembang untuk mengimplementasikannya pada website yang dibuat. Pada tahap Code tim pengembang melakukan implementasi dan refactoring beberapa kali sesuai dengan feedback yang diterima dari product pemilik produk. Pada tahap pengujian ini dilakukan uji coba fungsional dengan teknik black box testing. Pengujian blackbox testing akan menguji hasil akhir yang didapatkan [10]. Hasil pengujian dapat dilihat pada tabel 2

Tabel 2. Hasil Pengujian BlackBox

Test Class	Output	Uji Identifikasi	Kesimpulan
Home Page	Dapat menampilkan sorotan dari dhealth	UC-01	Berhasil Masuk Aplikasi
Product page	Dapat menampilkan detail produk untuk dhealth	UC-02	Berhasil
Features Page	Dapat menampilkan dhealth	UC-03	Berhasil
Knowled ge-base page	Dapat menampilkan tampilan dasar dari dhealth	UC-04 Rule Aplikasi	Berhasil
Contact Page	Dapat menampilkan alamat yang dapat dihubungi	UC-05	Berhasil

Setelah website berhasil dibangun dan diimplementasikan microservices, maka tahap terakhir dari penelitian ini adalah melakukan pengujian dan membandingkan hasil dari implementasi arsitektur microservices dan arsitektur monolitik. Metode pengujian akan menggunakan metode load testing untuk mengukur seberapa kuat masing-masing arsitektur ketika menangani beban yang besar. Dalam pengujian beban, digunakan sebuah tools yaitu web application load stress and performance testing (WAPT). Tools ini akan membantu mengukur skenario yang sama pada arsitektur yang berbeda, dan juga menggunakan GTMetrix sebagai tools tambahan untuk

mengukur performa masing-masing arsitektur. Hasil pengukuran dan pengujian dapat dilihat pada tabel 3, dimana dapat dilihat hasil yang dapat menggambarkan keunggulan dari arsitektur microservices ini yang hampir menghasilkan performa yang lebih baik hampir 2x lipat

4. Kesimpulan

Dari penelitian ini, dapat disimpulkan Situs web yang telah dibangun sebagai prototipe dari penelitian ini telah mengimplementasikan arsitektur layanan mikro. Situs web memiliki hubungan berdasarkan cerita yang diberikan oleh pemilik produk. Hasil uji coba implementasi arsitektur layanan mikro dapat mengelola situs web dan kinerja yang lebih baik dari arsitektur monolitik. Untuk pengembangan lebih lanjut berdasarkan penelitian ini yaitu penelitian ini hanya menggunakan 2 framework yaitu reactjs dan vuejs. Untuk penelitian selanjutnya dapat ditambahkan framework lain. kedepannya dapat ditambahkan framework lain, dan juga penggunaan framework juga akan menentukan hasil kecepatan dan kemampuan sistem dalam menangani beban yang besar

Daftar Rujukan

- [1] Lubis, Nazwa Salsabila, and Muhammad Irwan Padli Nasution. "Perkembangan Teknologi Informasi Dan Dampaknya Pada Masyarakat." *Kohesi: Jurnal Sains dan Teknologi* 1.12 (2023): 41-50.
- [2] N. M. Devadiga, "Software engineering education: Converging with the startup industry," in *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2017, pp. 192–196.
- [3] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: A systematic mapping study," *Information and Software Technology*, vol. 56, no. 10, pp. 1200–1218, 2014
- [4] A. Bucchiarone, N. Dragoni, S. Dustdar, S. T. Larsen, and M. Mazzara, "From monolithic to microservices: An experience report from the banking domain," *Ieee Software*, vol. 35, no. 3, pp. 50–55, 2018.
- [5] W. Hasselbring and G. Steinacker, "Microservice architectures for scalability, agility and reliability in ecommerce," in *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. IEEE, 2017, pp. 243–246
- [6] S. Haryati, "Research and development (r&d) sebagai salah satu model penelitian dalam bidang pendidikan," *Majalah Ilmiah Dinamika*, vol. 37, no. 1, p. 15, 2012.
- [7] R. S. Pressman, "Rekayasa perangkat lunak: Pendekatan praktisi," 2012.
- [8] M. Geers, "Micro frontends," Aug 2017. [Online]. Available: <https://micro-frontends.org/>
- [9] B. S. Akpolat and W. Slany, "Enhancing software engineering student team engagement in a high-intensity extreme programming course using gamification," in *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2014, pp. 149–153.
- [10] T. E. Vos, F. F. Lindlar, B. Wilmes, A. Windisch, A. I. Baars, P. M. Kruse, H. Gross, and J. Wegener, "Evolutionary functional black-box testing in an industrial setting," *Software Quality Journal*, vol. 21, no. 2, pp. 259–288, 2013