

IMPLEMENTASI MICRO FRONTEND PADA PENGEMBANGAN APLIKASI WEB (BAPENDA WEB)

Waluyo Jati Raharjo¹, Lintang Yuniar Banowosari²

^{1,2}Universitas Gunadarma, Depok, Indonesia

e-mail: waluyojati.raharjo10@gmail.com¹, lintang@staff.gunadarma.ac.id²

Abstrak

Perkembangan aplikasi web telah melahirkan konsep microfrontend, di mana antarmuka pengguna dibagi menjadi bagian-bagian kecil yang dikelola secara terpisah oleh tim pengembang yang berbeda. Pendekatan ini memungkinkan pengembangan yang lebih cepat, skalabilitas, dan pemeliharaan yang lebih mudah. Seiring dengan itu, arsitektur microservices juga telah mendapatkan perhatian dan menjadi objek penelitian penting dalam bidang ilmu informasi. Implementasi arsitektur microservices pada backend telah memberikan solusi yang baik. Oleh karena itu, muncul ide untuk menggabungkan pemisahan frontend dan backend dengan mengimplementasikan microservices pada frontend, mengubah aplikasi web dari SPA menjadi kombinasi banyak aplikasi frontend kecil. Hal ini dapat mengatasi masalah dalam pemeliharaan dan penyesuaian aplikasi web, serta mengurangi ketergantungan antar modul. Provinsi XYZ memiliki rencana untuk menerapkan microfrontend dalam pengembangan aplikasi web mereka yang saat ini masih menggunakan arsitektur monolitik. kratif tidak berpengaruh signifikan terhadap pertumbuhan ekonomi Indonesia.

Kata kunci: Microservices, Frontend, Backend, Microfrontend

Abstract

The development of web applications has given birth to the concept of microfrontend, where the user interface is divided into small parts that are managed separately by different teams of developers. This approach allows for faster development, scalability, and easier maintenance. Along with that, microservices architecture has also gained attention and become an important object of research in the field of information science. The implementation of microservices architecture on the backend has provided a good solution. Therefore, the idea arose to combine the separation of frontend and backend by implementing microservices on the frontend, turning web applications from SPAs into combinations of many small frontend applications. This can solve problems in maintaining and customizing web applications, and reduce dependencies between modules. XYZ Province has plans to implement microfrontend in the development of their web applications which currently still use.

Keywords: Microservices, Frontend, Backend, Microfrontend.

PENDAHULUAN

Pengembangan aplikasi web telah mengalami banyak perubahan dan inovasi beberapa tahun terakhir. Salah satu pendekatan yang muncul adalah konsep micro frontend, yang mengadopsi prinsip modularitas dalam pengembangan frontend. Micro frontend memecah antarmuka pengguna menjadi bagian-bagian yang lebih kecil, yang dikelola secara terpisah oleh tim pengembang yang berbeda. Pendekatan ini memungkinkan pengembang untuk bekerja secara independen pada bagian bagian tertentu dari aplikasi, sehingga memfasilitasi pengembangan yang lebih cepat, skalabilitas, dan pemeliharaan yang lebih mudah.

Bersamaan dengan kenaikan konsep pemisahan frontend dan backend dan kedewasaan teknologi yang bersangkutan, arsitektur yang memisahkan frontend dan backend membuat pengembangan sistem dapat menerapkan decoupling yang sering disebut arsitektur microservices. Hasil implementasi ini menghasilkan single-page application (SPA). SPA itu sendiri secara teori, merupakan aplikasi dimana perubahan tidak memerlukan pemuatan halaman saat digunakan, dimana pengguna tidak berpindah halaman saat melakukan request kepada server setiap terjadi peristiwa pada aplikasi. Berdasarkan SPA, tim frontend membuat dan memelihara aplikasi menggunakan REST API untuk mengambil data dari layanan backend. Pendekatan ini memberikan pengalaman baik untuk pengguna namun membuat SPA tidak terskala dengan baik. Seiring dengan perjalanan waktu dan mengikuti perkembangan bisnis, kondisi ini membuat proyek frontend menjadi sangat sulit untuk dipelihara, terlebih pada aplikasi yang kaya fitur dan berdiri diatas backend dengan arsitektur microservices. Dalam beberapa tahun terakhir, arsitektur microservices mendapatkan perhatian dan menjadi salah

satu objek penelitian kunci pada bidang ilmu informasi. Pada perkembangan arsitektur microservices pada backend, memberikan implementasi solusi yang baik. Dari sini kemudian muncul ide untuk mengkombinasikan pemisahan pada bagian frontend dan backend, dan membawa implementasi microservices pada bagian frontend, mengubah bentuk aplikasi web dari SPA menjadi kombinasi banyak aplikasi frontend kecil. Ide dibelakang micro frontend adalah untuk memperlakukan aplikasi sebagai kombinasi fitur yang dimiliki oleh tim yang berbeda, dimana setiap tim mempunyai fungsi bisnis independen yang mereka fokuskan. Sebuah tim adalah lintas-fungsional dan mengembangkan fitur-fiturnya dari ujung ke ujung, dari backend ke frontend.

Aplikasi web milik salah satu instansi selevel provinsi ini adalah aplikasi web yang menggunakan arsitektur monolitik dan digunakan untuk melakukan aktifitas pencatatan, pembayaran dan pemeriksaan pajak. Aplikasi ini sering mendapatkan perubahan dan penyesuaian seringkali terjadi malfungsi yang menyebabkan pengalaman pengguna dalam menggunakan aplikasi menurun, dan efektifitas pengembang dalam pemeliharaan menurun pula, selain itu, seiring dengan pertumbuhan dari aplikasi – aplikasi yang ada melahirkan ketergantungan yang cukup kuat antar modul sehingga pada tahap pengembangan kedepannya direncanakan perombakan arsitektur. Dari rencana ini kemudian muncul ide untuk mengimplementasikan micro frontend pada pengembangan aplikasi Bapenda Web di provinsi XYZ.

METODE

Jenis Penelitian

Penelitian ini menggunakan pendekatan kualitatif dengan studi kasus sebagai metode utama.

Lokasi dan Subjek Penelitian

Lokasi penelitian dilakukan di tim pengembang BAPENDA yang bertanggung jawab untuk aplikasi web. Subjek penelitian meliputi arsitek, pengembang, dan pemelihara aplikasi web.

Pengumpulan Data

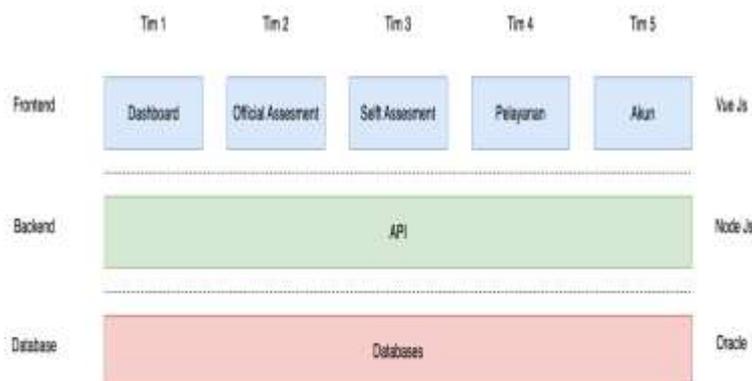
Pengumpulan data dilakukan melalui wawancara dengan anggota tim pengembang, observasi terhadap proses pengembangan, dan analisis dokumen terkait.

Analisis Data

Data yang terkumpul dianalisis dengan pendekatan analisis konten untuk mengidentifikasi dampak implementasi micro frontend terhadap skalabilitas, maintainabilitas, dan efisiensi pengembangan.

HASIL DAN PEMBAHASAN

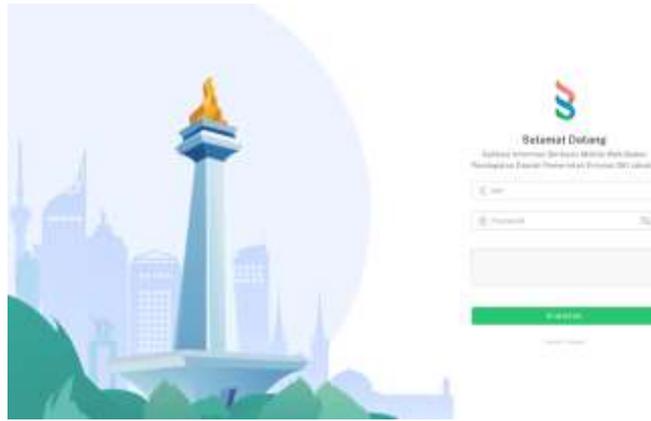
Arsitektur Aplikasi



Gambar 1. Arsitektur Aplikasi

Gambar diatas menunjukkan arsitektur aplikasi dengan menerapkan konsep microservices dan microfrontend . Dengan memisahkan antara frontend dan backend dengan frontend mengkonsumsi API dari backend .Gambar diatas menunjukkan arsitektur aplikasi dengan menerapkan konsep microservices dan microfrontend . Dengan memisahkan antara frontend dan backend dengan frontend mengkonsumsi API dari backend .

a. Frontend



Gambar 2. Tampilan Frontend

b. Backend



Gambar 3. Tampilan Backend

Tabel 1. Pembagian Tim

Modul	Tim 1	Tim 2	Tim 3	Tim 4	Tim 5
Dashboard	√				
Official Assesment <ul style="list-style-type: none"> • PBB • Pajak Reklame 		√			
Self Assesment <ul style="list-style-type: none"> • Pajak Hiburan • Pajak Restoran • Pajak Hotel • Pajak Parkir • Pajak Bahan Bakar 			√		
Pelayanan <ul style="list-style-type: none"> • LHP Lapangan 				√	
Akun <ul style="list-style-type: none"> • Akun • Tentang • Keluar 					√

Tabel diatas merupakan tabel pembagian tim dalam arsitektur microfrontend dalam pengembangan aplikasi Bapenda Mobile. Pembagian tim di tunjukan untuk membagi dalam mengembangkan sisi frontend yang meliputi didalamnya yaitu bertanggung jawab atas style, view, dan validation.

Routing dan Server-Side

Pengembangan aplikasi dengan arsitektur microfrontend bagian terpenting adalah routing. Berikut routing yang dilakukan:

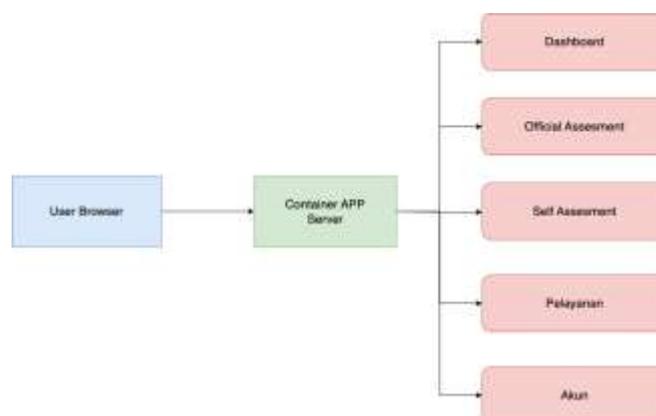
a. Pada file vite.config.js

```
export default defineConfig({
  base: '/mobile/',
  publicPath: '/mobile/',
  outputDir: 'dist',
  assetsDir: 'static'})
```

Gambar 4. File Vite

b. Pada vue router

```
export default [
  { heading: 'Dashboard Analisis', role:'user' },
  { title: 'Dashboard', to: { name: 'dashboard' }, icon: { icon: 'tabler-dashboard' },
    role:'user'},
  { heading: 'Official Assessment', role:'user' },
  { title: 'PBB', to: { name: 'pbb' }, icon: { icon: 'mdi-home-city' },role:'user'},
  { title: 'Pajak Reklame', to: { name: 'pajak-reklame' }, icon: { icon: 'mdi-developer-board' },role:'user'},
  { heading: 'Self Assesment', role:'user' },
  { title: 'Pajak Hiburan',to: { name: 'pajak-hiburan' },icon: { icon: 'mdi-party-popper' },role:'user' },
  { title: 'Pajak Restoran', to: { name: 'pajak-restoran' }, icon: { icon: 'mdi-food' },role:'user'},
  { title: 'Pajak Hotel', to: { name: 'pajak-hotel' }, icon: { icon: 'mdi-bed' },
    role:'user'},
  { title: 'Pajak Parkir',to: { name: 'pajak-parkir' }, icon: { icon: 'mdi-parking' }, role:'user'},
  { title: 'Pajak Bahan Bakar', to: { name: 'pajak-bahan-bakar' }, icon: { icon: 'mdi-oil' },role:'user' },
  },
  { heading: 'Pelayanan', role:'user' },
  { title: 'LHP Lapangan', to: { name: 'lhp-lapangan' }, icon: { icon: 'mdi-face-agent' },role:'user'},
  { heading: 'Akun', role:'user' },
  { title: 'Akun', to: { name: 'akun' }, icon: { icon: 'tabler-user' }, role:'user' },
  { title: 'Tentang', to: { name: 'tentang' }, icon: { icon: 'mdi-information-outline' },role:'user'},
  { title: 'Keluar',to: { name: 'login' },icon: { icon: 'tabler-logout' },role:'user'},]
```



Gambar 6. Server Side

Gambar tersebut merupakan server side aplikasi Bapenda Web dalam microfrontend fitur dan fungsional terbagi menjadi lebih kecil dan saling terintegrasi.

Tampilan Bapenda Web



Gambar 6. Tampilan Aplikasi Bapenda Web

Gambar diatas merupakan gambar tampilan aplikasi Bapenda Web yang terbagi menjadi lima tim dalam pengembangannya dimana masing masing tim mengerjakan modul dan fungsional yang berbeda namun saling berintegrasi dan membentuk satu sistem besar bernama Bapenda Web.

SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa aplikasi Bapenda Web yang baru memenuhi tujuan utama penelitian, yaitu mengimplementasikan arsitektur micro frontend, dengan memisah struktur aplikasi dari satu bagian besar menjadi bagian bagian kecil yang independen, sehingga saat pengembang melakukan pemeliharaan pada suatu bagian dari aplikasi, maka bagian lain dapat tetap digunakan oleh pengguna dan tidak dipengaruhi oleh proses maupun perubahan yang dilakukan. Hal ini membuat pemeliharaan lebih mudah, karena masing masing tim dapat fokus pada bagian yang menjadi tanggung jawabnya tanpa memikirkan pengaruh perubahan yang terjadi pada bagian lain karena masing masing layanan sudah berdiri secara independen. Perancangan serta implementasi pada antarmuka aplikasi juga dibuat untuk memberikan kemudahan dan pengalaman UI/UX yang baik untuk pengguna sehingga aplikasi mudah untuk digunakan. Tentu, masih ada ruang untuk perbaikan lebih lanjut, dimana beberapa ide perbaikan yang dapat diimplementasikan penulis tuang pada bagian saran dibawah.

SARAN

Saran yang dapat diberikan oleh penulis untuk meningkatkan performa aplikasi dan efisiensi pemeliharaan aplikasi adalah mengimplementasikan arsitektur microservices pada bagian belakang (backend) agar kinerja aplikasi dan implementasi yang sudah dilakukan dapat dimaksimalkan, dalam artinya pemisahan layanan dan pemeliharaan dari layanan tersebut dapat berjalan dengan efisien, dan layanan dapat digunakan sesuai kebutuhan dengan baik. Selain itu, penulis juga menyarankan penggunaan arsitektur clean code, untuk memudahkan pengembangan aplikasi agar lebih mudah untuk dibaca dan dipahami oleh anggota tim, membuat struktur kode serapi mungkin, sehingga aplikasi lebih mudah untuk diuji (testable). Implementasi ini juga dapat meminimalisir refactoring kode yang dapat menguras waktu dan tenaga.

DAFTAR PUSTAKA

- Dragoni N, Giallorenzo S, Lafuente A L, et al. "Microservices: yesterday, today, and tomorrow [M]". Present and Ulterior Software Engineering. Springer, Cham, 2017: 195-216...
<https://martinfowler.com/articles/micro-frontends.html>
- CaifangYang,etal."ResearchandApplicationofMicroFrontends"2019IOP Conf. Ser.: Mater. Sci. Eng. 490 062082.
- Richardson, Chris. "Microservice architecture pattern". [Online] Available: microservices.io. [Accessed 2021-06-19].
- P. Y. Tilak, V. Yadav, S. D. Dharmendra and N. Bolloju, "A platform for enhancing application developer productivity using microservices and micro- frontends," 2020 IEEE-HYDCON, 2020, pp. 1-4, doi: 10.1109/HYDCON48903.2020.9242913.

H. Park, W. Jung and S. Moon, "Javascript ahead-of-time compilation for embedded web platform," 2015 13th IEEE Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia), 2015, pp. 1-9, doi: 10.1109/ESTIMedia.2015.7351768.

M. Jazayeri, "Some Trends in Web Application Development," Future of Software Engineering (FOSE '07), 2007, pp. 199-213, doi: 10.1109/FOSE.2007.26.

<https://vuejs.org/guide/introduction.html#what-is-vue>

<https://vitejs.dev/guide/>

<https://nodejs.org/en/docs>