

Implementasi Teknik Dokumentasi OpenAPI 3.0 Menggunakan Swagger Pada Usertip Singapura

William Wijaya

Fakultas Ilmu Komputer, Universitas Internasional Batam
e-mail: williamwijaya25@gmail.com

Abstrak

Usertip Singapura merupakan salah satu perusahaan *startup* yang memiliki beberapa pengembang aplikasi. Komunikasi antar pengembang merupakan salah satu faktor terpenting dalam siklus pengembangan, maka dari itu dengan adanya fitur/teknik dokumentasi dengan standar OpenAPI 3.0 dan Swagger sebagai alat untuk memberikan interaksi pengguna terhadap dokumentasi, hal ini dapat meningkatkan efisiensi komunikasi tidak langsung antar pengembang *back-end* dan *front-end* di saat ada pengembangan fitur baru di dalam aplikasi.

Abstract

Usertip Singapore is a startup company that has several application developers. Communication between developers is one of the most important factors in the development cycle, therefore with the documentation features / techniques with the OpenAPI 3.0 standard and Swagger as a tool to provide user interaction with documentation, this can increase the efficiency of indirect communication between back-end and front-end developers when there is new feature development in the application.

Keywords: *OpenAPI, Swagger, Documentation, Development Cycle*

Pendahuluan

Usertip Private Limited merupakan sebuah perusahaan berbasis *Software-as-a-Service* (SaaS) yang memungkinkan para pelanggan untuk memanfaatkan aplikasi yang dikembangkan oleh Usertip dengan basis *pay-as-you-go* dan berguna untuk menghilangkan kebutuhan dalam melakukan instalasi kompleks serta kebutuhan perangkat keras secara langsung [1]. Berlokasi di Singapura, Usertip juga merupakan *startup* yang sudah bermula sejak tahun 2020 dan saat ini sudah berkembang hingga menjadi salah satu *startup* dengan potensial tinggi di Singapura.

Usertip adalah *platform* tanpa kode untuk mengelola kebutuhan adopsi digital. Pembangun tanpa kode untuk mengelola panduan dalam aplikasi, pembaruan fitur, dan pelacakan pengguna memungkinkan respons yang cepat terhadap perubahan fitur dan kebutuhan pengguna. Baik dalam hal orientasi, pelatihan, atau dukungan,

fokus Usertip adalah selalu memberikan pengalaman yang baik bagi pelanggan dan pengguna.

Aplikasi yang dikembangkan oleh Usertip merupakan aplikasi dengan basis *no code platforms* yang memungkinkan untuk melakukan sebuah pengembangan berbasis *cloud* tanpa pengkodean tradisional dalam editor berbasis teks. *No code platforms* mengandalkan antarmuka grafis untuk meningkatkan kecepatan pengembangan perangkat lunak dan melonggarkan koneksi ke pengembang IT [2].

Sejak tahun 2020 saat Usertip masih bermula, hanya terdapat 2 pengembang utama yang mengembangkan aplikasi Usertip. Namun sejak tahun 2022, Usertip sudah memulai menambahkan sumber daya manusianya terutama di bagian *developer*. Hal ini juga diikuti dengan pemisahan dua tim yang berbeda yang mengerjakan dua bagian yang berbeda saat mengembangkan aplikasi Usertip.

Front-end yang berfokus pada pengembangan antarmuka pengguna yang menyajikan fungsi atau kegunaan dari sebuah situs dan juga interaksi secara langsung dengan pengguna dan juga *front-end* memiliki kaitan dengan pengalaman dari pengguna ketika berinteraksi dengan situs. *Front-End Developer* merupakan sebuah sebutan mereka yang melakukan pengembangan terhadap interaksi antarmuka pengguna situs [3].

Back-end atau yang lebih sering di sebut sebagai *server side* pada dasarnya adalah sebuah tempat dimana seluruh proses dari sebuah aplikasi atau sistem berjalan. Proses yang terjadi di *back-end* biasanya berfungsi untuk mengubah, menghapus, atau menambahkan data yang bisa dimanfaatkan nantinya oleh *front-end* [4].

Dengan bertambahnya jumlah *developer* di dalam perusahaan, juga memunculkan masalah baru, yaitu bagaimana para *developer* dapat berkomunikasi dengan efektif agar mereka dapat mengerjakan sebuah aplikasi yang sama tapi juga tidak mengganggu atau memiliki dampak besar pada *developer* lainnya.

Salah satu metodologi umum yang dipakai oleh tim pengembang salah satunya adalah metodologi *Agile* yang juga dipakai oleh tim Usertip dalam melakukan proses pengembangan aplikasi perusahaan tersebut.

Metodologi *Agile* merupakan sebuah metodologi pengembangan *software* yang berfokus pada organisasi sendiri dan adaptasi berkelanjutan, dimana para pengembang harus selalu berkomunikasi dengan klien dan para pengembang dan bisa saling berkoordinasi untuk mencapai tujuan yang mutual daripada tetap fokus pada satu titik yang sudah disepakati dahulunya, namun bisa beradaptasi untuk menghadapi perubahan dalam proses pengembangan [5].

Dikarenakan metodologi *Agile* memfokuskan pada organisasi sendiri, dimana para pengembang dapat

beradaptasi dengan tugasnya untuk mencapai hasil yang lebih baik daripada yang direncanakan juga memberikan masalahnya sendiri, salah satu dari masalah tersebut adalah bagaimana para pengembang bisa memberikan informasi perubahan/adaptasi kepada pengembang lainnya secara efektif.

Ruang Lingkup yang akan menjadi fokus dalam pelaksanaan kerja praktek ini adalah sebagai berikut:

1. Kegiatan PkM ini dilaksanakan di Usertip Private Limited
2. Merancang aplikasi dokumentasi dengan basis OpenAPI 3.0, menggunakan Swagger
3. Pengembangan aplikasi ini akan menggunakan bahasa YAML, sebuah bahasa sterilisasi yang dapat dibaca manusia
4. Aplikasi ini juga akan dikembangkan hingga aplikasi tersebut dapat dijalankan secara lokal di setiap komputer para *developer* hanya dengan menggunakan *one line script command*
5. *Script* ini akan dibuat menggunakan bahasa Javascript dan Node.js sebagai *framework* dasarnya.

Tujuan dari pelaksanaan kegiatan PkM adalah mengembangkan sebuah alat atau platform dokumentasi yang bisa digunakan oleh setiap pengembang internal Usertip untuk bisa saling mengkomunikasikan panduan penggunaan API yang dikembangkan oleh pengembang lainnya. Dengan adanya *platform* ini, maka diharapkan dapat meningkatkan efisiensi pengembangan aplikasi di Usertip.

Masalah

Salah satu contoh dari masalah yang dihadapi adalah saat pengembang *back-end* merilis API (*Application Program Interface*) baru. API merupakan sekumpulan protokol yang menentukan bagaimana cara sebuah aplikasi bisa berinteraksi dengan aplikasi lain dalam rangka memfasilitasi pertukaran Informasi

atau dalam artian lain API merupakan sebuah protokol yang menyatukan protokol satu dengan lainnya dengan berbicara bahasa yang sama [6].

Para pengembang *back-end* perlu berkomunikasi kepada pihak *front-end* bagaimana cara menggunakan API tersebut, namun terdapat kasus umum dimana komunikasi langsung antar pihak dapat memakan waktu lama yang bisa menghambat efisiensi kecepatan pengembangan *software*. Maka dari itu, diperlukan sebuah solusi dimana saat pihak *back-end* merilis sebuah API baru. Userтип juga merilis sebuah dokumentasi atau panduan yang bisa dipakai dan dipahami secara mudah oleh pihak *front-end* bagaimana cara menggunakan API tersebut.

Metode

Teknik pengumpulan data dalam pengerjaan PkM ini adalah dengan melakukan observasi langsung, dikarenakan *platform* ini memiliki tujuan untuk melakukan dokumentasi langsung terhadap API yang sudah dan akan dikembangkan oleh pengembang di masa lalu dan masa depannya. Maka, perlu dilakukan pengumpulan data bagaimana cara menggunakan API tersebut dan memulai menyusun rencana dalam pembuatan *platform* tersebut.

Terdapat beberapa proses yang akan dilakukan saat melakukan pengembangan *platform* ini, dimulai dari *planning*, *benchmarking*, *designing*, *programming*, *testing*, dan *distribution*. Penjelasan setiap tahapan dapat di lihat sebagai berikut:

1. *Planning*

Tahap yang pertama adalah melakukan sebuah perencanaan dengan manajer proyek mengenai bagaimana cara mengembangkan *platform* ini mulai dari mencari cara memakai alat Swagger dan memahami konsep OpenAPI sebagai basis dasar dalam pembuatan *platform* ini.

2. *Benchmarking*

Tahap ini merupakan tahap untuk melakukan riset terhadap aplikasi atau mencari kasus lain yang memakai teknik yang sama sebagai sebuah referensi untuk bisa mengetahui pendekatan terbaik apa yang bisa di lakukan pengembang *platform* saat mengembangkan aplikasi dokumentasi ini.

3. *Designing*

Tahap selanjutnya merupakan tahap *designing*. Karena *platform* ini akan lebih mengarah untuk pemakaian oleh pengembang internal di dalam Userтип, maka tahap ini akan menjadi tahap yang tidak begitu diperlukan. Swagger juga sudah memiliki desain dasar yang bisa di gunakan dan sudah cukup ramah pengguna untuk para pengembang yang akan memakai *platform* ini.

4. *Programming*

Tahap ini merupakan tahap untuk melakukan pemrograman *platform* yaitu dimulai dengan melakukan pemrograman setiap dokumentasi API yang sudah pernah dikembangkan sebelumnya menggunakan bahasa YAML, selanjutnya akan di lanjutkan dengan pemrograman aplikasi *interface* menggunakan Javascript agar *platform* dokumentasi ini dapat dijalankan di setiap komputer pengembang dengan menggunakan *one line script*.

5. *Testing*

Setelah tahap *designing* dan tahap pemrograman sudah selesai, maka di perlukan tahap *testing* untuk memastikan bahwa setiap dokumentasi API yang sudah di ketik dan di program sudah sesuai dengan fungsionalitas mereka masing-masing, selain itu juga dilakukan *testing* untuk melihat apakah *user interface* yang di hasilkan oleh alat Swagger sudah cukup ramah pengguna untuk para pengembang.

6. *Distribution*

Platform yang sudah diuji dan tidak memiliki masalah dalam dokumentasi maupun *user interface*, maka dapat didistribusikan kepada para pengembang lain untuk bisa langsung digunakan sebagai *platform* untuk mendapatkan informasi lebih lanjut tentang bagaimana cara menggunakan API yang sudah dikembangkan oleh pengembang sebelumnya.

Proses pelaksanaan kegiatan PkM ini dibagi menjadi dua bagian, yaitu persiapan dan pelaksanaan. Penjelasan untuk setiap bagian adalah sebagai berikut:

1. Pada tahap persiapan, melakukan observasi kepada API yang sudah dikembangkan dan melakukan riset dalam cara penggunaan Swagger dan bagaimana struktur dokumentasi standar OpenAPI.
2. Pada tahap pelaksanaan, akan dilakukan 4 tahap *designing*, *programming*, *testing*, dan *distribution*. *Platform* ini akan dikembangkan, sehingga memiliki fitur pokok yang terpenuhi dan bisa di-*install* menggunakan *one line script*, maka pada akhirnya bisa didistribusikan kepada para pengembang lainnya.

Lokasi kegiatan dilakukan di perusahaan Usertip Singapura pada Januari 2023 hingga Juli 2023. Pembagian tugas dari setiap periode adalah sebagai berikut:

1. Januari 2023, Minggu pertama dari bulan ini akan difokuskan pada perancangan awal dan melakukan observasi dasar pada *basecode* dari aplikasi yang ingin didokumentasikan. Dilanjutkan dengan Minggu kedua hingga akhir bulan difokuskan untuk memulai mengkodekan setiap dokumentasi dari setiap API yang terdapat pada setiap servis aplikasi.
2. Februari 2023 hingga April 2023, difokuskan untuk menyelesaikan

kode dokumentasi untuk setiap API atau fitur yang terdapat di dalam aplikasi. Pengkodean ini menggunakan bahasa YML.

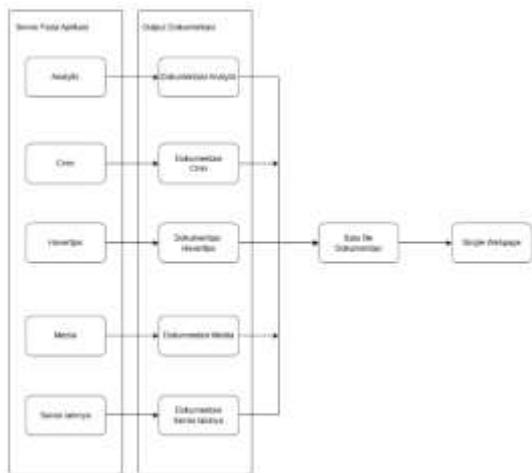
3. Juni 2023, periode ini digunakan untuk mengkodekan sebuah aplikasi atau *script* simpel yang memiliki fungsi untuk menjalankan program dokumentasi untuk memberikan sebuah interaksi *user* kepada para pengguna, hal ini juga bertujuan untuk agar para pengguna tidak perlu mengetahui bagaimana cara kerja kode dokumentasi secara teknis.
4. Juli 2023, periode ini sebagian besar untuk melakukan *testing* dan memperbaiki *bug* atau kerusakan kecil yang ditemukan saat melakukan *testing* pada program dokumentasi dan akhirnya melakukan distribusi kepada para pengembang aplikasi yang membutuhkan informasi dokumentasi tentang fitur yang dikembangkan oleh pengembang lainnya.

Pembahasan

Hasil yang dicapai dalam masa kerja praktek ini adalah sebuah *platform* atau aplikasi dokumentasi dari setiap API yang sudah pernah dikembangkan oleh pengembang sebelumnya. Terdapat 12 kategori servis API yang didokumentasikan, setiap kategori memiliki bagian mereka sendiri, maka dari itu diperlukan *script* yang berfungsi untuk menggabungkan setiap kategori menjadi satu *platform*.



Gambar 1. Tampilan UI/UX Dokumentasi



Gambar 2. Flow Rencana Penggabungan Setiap Kategori

Setiap kategori yang sudah memiliki *output* dokumentasi sendiri akan diproses lagi menjadi satu dokumentasi yang berisikan dokumentasi dari setiap servis. Setelah itu dokumen tersebut akan di proses oleh Swagger untuk menginput dan meng-*hosting* sebuah *web server* lokal yang bisa diakses oleh pengembang untuk melihat sebuah dokumentasi API yang sudah tercatat di dalam dokumentasi tersebut.



Gambar 4. Tampilan Gabungan Antar Dua Kategori Dokumentasi Berbeda

Setelah situs dokumentasi lokal sudah bisa diakses oleh pengembang, mereka bisa melihat setiap API yang pernah dikembangkan oleh pengembang sebelumnya mulai dari cara penggunaan, fungsi dari API tersebut, struktur dokumen yang diperlukan oleh API tersebut, dan berbagai respon dari API tersebut setelah menerima data baik respon sukses, serta respon gagal atau respon *error*.

Di dalam dokumentasi juga terdapat skema data yang berfungsi sebagai alat

untuk memberikan informasi lebih tentang jenis data yang diperlukan oleh API agar bisa berfungsi dengan benar.



Gambar 3. Tampilan Mendetil dari Tiap Dokumentasi API

Dengan *platform* ini, selanjutnya saat seorang pengembang dari *back-end* ingin mengembangkan sebuah fitur API baru, dapat langsung membuat dokumentasi mendetail tentang API tersebut, dan pengembang *front-end* saat ingin menggunakan API tersebut bisa langsung mengetahui apa yang diperlukan untuk menggunakan API tersebut.

Simpulan

Ustertip Private Limited adalah salah satu perusahaan SaaS Singapura yang mengalami penambahan pengembang yang bekerja di dalam perusahaan yang memaksa perubahan dalam metode pengembangan dari tradisional menjadi metodologi *Agile*. Namun, hal ini menimbulkan sebuah masalah di mana para pengembang banyak menghabiskan waktu untuk saling mengkomunikasikan cara penggunaan fitur yang baru di kembangkan.

Dengan menggunakan alat Swagger dan standarisasi OpenAPI, hal ini bisa menjadi solusi langsung, dimana setiap kali sebuah fitur baru dikembangkan, para pengembang bisa langsung mendapatkan dokumentasi mendetail tentang fitur API tersebut, mulai dari cara penggunaannya, struktur data yang diperlukan, hingga respon yang diberikan API saat berhasil ataupun gagal.

Saran

Platform yang dikembangkan dilaporan ini memfokuskan penggunaan untuk para pengembang internal, namun sebenarnya aplikasi ini juga bisa dimanfaatkan sebagai alat yang bisa menjadi tempat informasi API publik yang dikembangkan oleh perusahaan lain jika perusahaan tersebut memiliki API yang memang memiliki tujuan dipakai oleh publik. Jadi, *platform* ini tidak terbatas untuk penggunaan internal saja, tapi juga penggunaan komersial jika diinginkan.

Daftar Pustaka

- [1] T. Kusnandar, "Arsitektur Software As a Service (Saas) Pada Layanan Rumah Sakit," *J. Comput. Bisnis*, vol. 6, no. 1, pp. 17–26, 2012, [Online]. Available: <http://jurnal.stmik-mi.ac.id/index.php/jcb/article/view/82>.
- [2] C. Ploder, R. Bernsteiner, S. Schlögl, and C. Gschliesser, "The Future Use of LowCode/NoCode Platforms by Knowledge Workers – An Acceptance Study," in L. Uden, IH. Ting, and J. Corchado (eds.), *Knowledge Management in Organizations. KMO 2019, Communications in Computer and Information Science*, vol. 1027, Springer, Cham, 2019, pp. 493-504. https://doi.org/10.1007/978-3-030-21451-7_38
- [3] C. Chastro and E. Darmawan, "Perbandingan Pengembangan Front End Menggunakan Blade Template dan Vue Js," *J. Strateg. Maranatha*, vol. 2, no. 2, pp. 302–313, 2020.
- [4] E. Daruqutni, "Rancang Bangun Back End Aplikasi Data Produk pada TOKO PERTANIAN PUJI BULUH MADIUN Berbasis Web," Undergraduate thesis, Universitas Dinamika, 2022.
- [5] V. Venkatesh, J. Y. L. Thong, F. K. Y. Chan, H. Hoehle, and K. Spohrer, "How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills," *Inf. Syst. J.*, vol. 30, no. 4, pp. 733–761, 2020, doi: 10.1111/isj.12282.
- [6] S. A. Nathania, L. Abubakar, and T. Handayani, "OPEN APPLICATION PROGRAMMING INTERFACE DENGAN KETENTUAN PERBANKAN DIGITAL LEGAL IMPLICATION OF USING THE OPEN APPLICATION PROGRAMMING INTERFACE (API) FOR BANKING SERVICES ASSOCIATED WITH keuangan perbankan . Salah satu faktor munculnya digitalisasi sela," pp. 244–259, 2023.