

Desain Prototype Frontend Auto Generator Based On REST API

Putra Prima Arhandi¹, Yushintia Pramitarini², Rizky Alviandra³

^{1,2,3}Teknologi Informasi, Teknik Informatika, Politeknik Negeri Malang,
¹putraprima@polinema.ac.id, ²yushintia@polinema.ac.id, ³rizkyalviandra@gmail.com

Abstrak— Frontend merupakan salah satu baik dari software development dimana frontend akan berpengaruh cukup besar terhadap minat dari pengguna ketika menggunakannya. Angular merupakan salah satu dari framework untuk membangun frontend yang dikembangkan oleh Google untuk saat ini. Pada umumnya, frontend developer membuat tampilan frontend yang sama dalam satu project dengan skala yang cukup besar. Ketika seorang developer mengulang-ulang baris kode yang sama secara terus menerus, rentan error dari baris kode tersebut akan semakin besar pula. Karena masalah ini muncul, maka developer perlu adanya code generator untuk mempercepat kinerjanya dan mengurangi rentan error karena pengulangan kode tersebut. Dari hasil implementasi, waktu yang digunakan untuk menulis baris kode akan terpotong sangat signifikan dibandingkan dengan menulis kode secara manual.

Kata kunci— Angular, Rest API, Express, Frontend

I. PENDAHULUAN

Perkembangan teknologi pada saat ini sangat pesat sehingga dapat memberikan manfaat yang lebih dalam kemajuan. Penggunaan teknologi dapat membantu manusia dalam membantu menyelesaikan masalah. Pada saat ini perkembangan teknologi sudah menggantikan aplikasi berbasis desktop dengan aplikasi berbasis website. Dalam pekerjaan *software engineering* terdapat dua istilah dalam pembuatan website yaitu *frontend* dan *backend*. Sering sekali *front-end developer* merasa cepat jenuh karena harus membuat baris kode yang sama dalam 1 project yang sama. Selain membuat *programmer* jenuh, penulisan kode yang sama secara berulang menyebabkan tingkat *error* yang cukup tinggi.

Pada penelitian sebelumnya terkait penelitian ini adalah penelitian milik Nikola Luburić, Goran Savić, Gordana Milosavljević, Milan Segedinac, Jelena Slivka yang dilakukan pada tahun 2016. Penelitian tersebut dapat *generate* front-end dengan basis REST. Front-end yang digunakan adalah angular dan REST yang digunakan adalah REST JSON. Penggunaan angular sangat membantu developer baru karena tingkat kesulitannya tidak tinggi. Angular juga dapat memberikan beberapa fungsi lebih terhadap tag html. Tingkat keberhasilan sisi *front-end* dari sistem *code generator* pada penelitian sebelumnya sebesar 65% berhasil generate dan tidak atau sedikit membutuhkan modifikasi serta 10% ketika diperlukan jumlah perubahan yang layak. Namun apabila

semakin seringnya perubahan versi pada angular, penelitian sebelumnya akan mengalami perubahan pula karena angular yang digunakan pada penelitian sebelumnya belum menggunakan angular typescript [1]. Peranan sekolah merupakan hal penting dalam kemajuan bangsa dan negara, karena sekolah salah satu sarana yang dapat meningkatkan kemampuan generasi penerus bangsa untuk menghadapi persaingan dengan negara lain.

Oleh karena itu dalam skripsi ini penulis akan membahas *Front-end Auto Generator Based on REST API* dengan menggunakan angular yang sudah menggunakan typescript karena typescript sudah memiliki standar penulisan yang cocok untuk digunakan *generator*. Selain itu, angular memiliki struktur data yang rapi dibandingkan dengan front-end library yang lainnya sehingga angular lebih cocok diimplementasikan sebagai code generator.

II. LANDASAN TEORI

A. Representational State Transfer

Representational State Transfer (REST) adalah suatu gaya arsitektur perangkat lunak untuk pendistribusian *hypermedia*. Istilah ini sering digunakan dengan longgar untuk mendeskripsikan semua antarmuka sederhana yang menyampaikan data dalam domain spesifik seperti HTTP. Sistem yang mengikuti prinsip REST Fielding sering disebut RESTful. REST yang digunakan berupa JSON karena pada server side, Response yang digunakan berupa JSON sehingga REST yang digunakan berupa REST JSON [2].

B. Angular

Menurut Lutfi Gani dalam buku Menguasai Angular JS untuk Membuat Website Dinamis, Angular merupakan front-end framework dengan basis code javascript yang dikembangkan pertama kali oleh sebuah perusahaan besar di dunia web yaitu Google. Angular dikembangkan konsep arsitektur MVC (Model View Controller) dan MVVM (Model View View Model). Angular dapat memberikan fungsi yang lebih pada tag-tag HTML sehingga menghasilkan lingkungan yang ekspresif dan mudah untuk dikembangkan. Angular juga dapat melakukan manipulasi DOM dengan baik, sehingga

mempercepat proses pembuatan aplikasi web. Penulisan baris kode secara clean code akan membantu programmer ketika akan berkolaborasi. Clean code adalah code dalam software dengan format yang benar dan disusun dengan rapi dan baik sehingga programmer lain dapat membaca dan memodifikasi code tersebut tanpa harus menanyakan terlebih dahulu kepada programmer sebelumnya. Selain itu, dibedakan juga smart component dan dump component dalam software. Smart component adalah component yang memikirkan bagaimana aplikasi bekerja. Sedangkan dump component adalah component yang bekerja untuk menampilkan sesuatu [3]. Angular adalah framework yang sudah cukup menata struktur komponennya sehingga dalam implementasi system ini angular lebih cocok dibandingkan dengan framework yang lainnya. Angular sudah bisa mengimplementasikan smart component dan dump component dalam struktur folder yang berbeda sehingga dalam proses di system ini, perubahan pada component dapat berjalan sesuai dengan alur yang sudah terbuat seperti one-way data flow. Dibandingkan dengan framework lainnya, angular telah memiliki rule yang sangat membantu untuk membuat system ini berjalan dan dapat digunakan dengan cepat.

C. Node JS

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi client atau browser saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. [4]. Node.js memiliki banyak kebutuhan yang dapat digunakan dalam pengembangan system ini seperti ExpressJS yang digunakan sebagai REST API Dummy, Angular-cli yang digunakan dalam membuat frontend template pada system ini, dan banyak library pendukung lainnya sehingga system ini menjadi suatu system yang diharapkan. Tanpa adanya Node.js, system ini tidak dapat dikembangkan dengan baik karena Node.js ini adalah package terpenting dalam pengembangan aplikasi ini.

D. Typescript

Typescript adalah zaman modern bahasa pengembangan Javascript. Ini adalah bahasa yang statis dikompilasi untuk menulis kode Javascript yang jelas dan sederhana. Hal ini dapat dijalankan pada NodeJS atau browser yang mendukung ECMAScript atau versi yang lebih baru. Typescript menyediakan opsional statis menetik, kelas, dan antarmuka [5]. Typescript tidak hanya membantu developer dalam mengembangkan aplikasi melalui angular saja. Typescript mulai digunakan di framework maupun library lainnya selain angular. Typescript sangat membantu pengembangan system ini karena Typescript dapat menentukan kesalahan-kesalahan yang umum dilakukan oleh javascript developer karena kebebasan penulisan di kode javascript. Typescript dapat

membatasi kebebasan dalam penulisan kode tersebut sehingga developer dapat mengurangi kesalahan-kesalahan kecil yang berdampak besar pada project yang sedang dikerjakan. Dalam system ini, Typescript pada angular telah membuat batasan penulisan kode yang berdampak sangat besar karena dengan menggunakan Typescript, angular akan memberikan batasan yang berkesinambungan dengan arsitektur project yang dilakukan.

E. Express JS

ExpressJS adalah sebuah framework dari javascript yang digunakan untuk membangun sebuah server pada website. ExpressJS menyediakan banyak fitur untuk membangun web application maupun RESTful API. Salah satu fitur yang dimiliki oleh expressJS adalah mengembangkan RESTful API dengan cepat dan mudah [6]. Fitur express ini menunjang dalam pengembangan system ini karena express dapat membuat RESTful API dengan cepat dan mudah. Selain dapat membuat RESTful API dengan mudah, express juga dapat membuat routing dengan cukup mudah. Routing adalah menentukan bagaimana suatu aplikasi merespon request client ke endpoint tertentu, yang merupakan path atau request HTTP Method tertentu seperti GET,POST,PUT, dan sebagainya [8]. Pada system ini, expressJS digunakan untuk membangun RESTful API sebagai sumber uji coba dari generator.

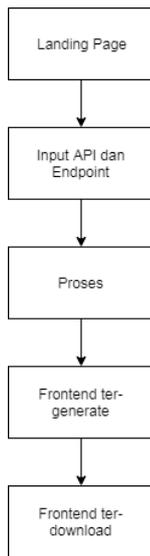
F. MYSQL

MySQL adalah basis data manajemen sistem SQL yang bersifat *open source* sehingga membuat MySQL ini terkenal. MySQL ini dikembangkan oleh Oracle Corporation, lalu didistribusikan. Basis data yang ada di MySQL bersifat *relational* atau relasi dimana penyimpanan data dapat diletakkan pada tabel yang berbeda daripada meletakkan data di satu tabel yang besar. Diantara data yang berbeda terdapat peraturan relasi yang dapat dibuat seperti *one-to-one*, *one-to-many*, unik, diperlukan atau opsional, dan petunjuk antara tabel yang berbeda. Sehingga basis data yang dibuat memiliki rancangan yang baik dan data yang ada tidak duplikat, tidak hilang dan selalu *update* [10]. MySQL database pada system ini berperan sebagai wadah dari data yang akan mendapatkan request dari REST API untuk template pada angular. MySQL database cocok dalam pengembangan software secara local karena tidak membutuhkan internet untuk mendapatkan data yang ada di database.

^aIdentify applicable sponsor/s here. If no sponsors, delete this text box (sponsors).

III. ANALISIS DAN PERANCANGAN

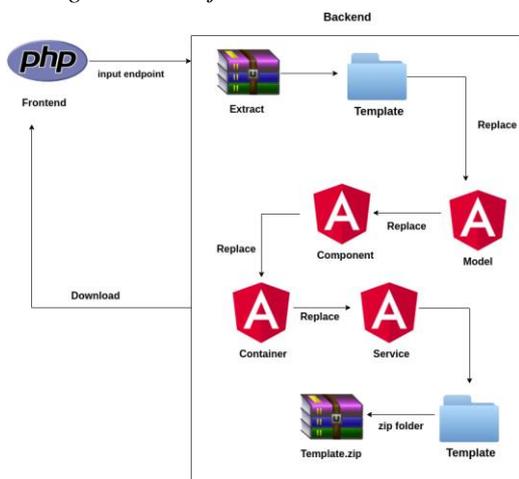
A. Perancangan User Flow



GAMBAR 1 USERFLOW SISTEM

Pada gambar 1 menunjukkan User Flow dari Frontend Auto Generator Based on REST API. Alur dimulai ketika pengguna membuka halaman generator. Saat pengguna membuka halaman generator, pengguna akan mendapatkan tampilan berupa *input text* untuk API, *input text* untuk *Endpoint*, dan *button submit* untuk memulai proses. Saat proses pada sistem selesai, maka frontend yang sudah di generate oleh system akan ter-download oleh system dan pengguna dapat menggunakannya secara langsung karena file frontend hasil dari system sudah disesuaikan dengan kebutuhan pengguna melalui REST API yang sudah disediakan pengguna sebelumnya.

B. Perancangan Alur Kerja Sistem



GAMBAR 2 ALUR KERJA SISTEM

Pada gambar 2, menunjukkan alur kerja dari sistem. Pada saat pengguna membuka aplikasi, maka frontend dari sistem akan meminta endpoint yang dibutuhkan user. Sistem akan melakukan ekstraksi file template yang akan digunakan. Pada

hasil dari ekstrak zip file, maka folder bernama generator-dashboard akan muncul pada folder template dan akan dipindahkan ke folder result untuk diproses dan disesuaikan dengan kebutuhan pengguna. Setelah folder dipindahkan, file yang akan di replace pertama kali adalah file model. File model seperti constructor dimana setiap value yang dibutuhkan akan diberikan inisiasi terlebih dahulu sebelum digunakan. File yang akan di *replace* setelah model adalah file component pada folder component. File *component* diubah terlebih dahulu karena file *component* sebagai *presentational component* akan menampilkan component ke browser. setelah file component diubah, selanjutnya adalah file component pada folder container karena pada smart component, *presentational component* akan diberikan fungsi agar dapat mengirim dan menerima data dari *service*. selain itu, *presentational component* akan ditata di folder *container* ini. setelah itu, file *service* akan diatur agar dapat memberikan fungsi-fungsi pada *container* sehingga pada *presentational component* dapat menjalankan aksi-aksi yang dimiliki *service*.

C. Kebutuhan Non-Fungsional

1) Kebutuhan Perangkat Keras

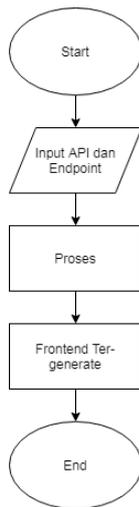
- a) Processor Intel
- b) Memory 4 gb
- c) HDD 500 gb

2) Kebutuhan Perangkat Lunak

Saat menuliskan nama penulis, tidak perlu ditunjukkan nama jabatan (contoh: Dosen Pembimbing), gelar akademik (contoh: Dr), atau keanggotaan dari setiap organisasi profesional (contoh: Senior Member IEEE). Dokumen *template* ini didesain agar penulisan afiliasi tidak ditulis berulang untuk beberapa penulis yang afiliasinya sama. Afiliasi ditulis singkat mungkin (contoh: tidak perlu membedakan afiliasi dari jurusan yang berbeda dalam universitas yang sama). *Template* ini didesain untuk dua afiliasi.

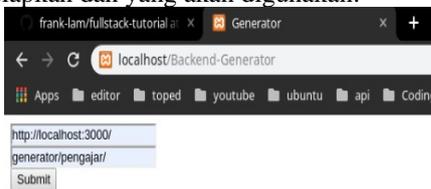
IV. IMPLEMENTASI

A. Implementasi Input API dan endpoint



GAMBAR 3 FLOWCHART INPUT API DAN ENDPOINT

Pada gambar 3, Flowchart dari Input API dan Endpoint dimulai dengan ketika pengguna membuka halaman generator, lalu pengguna akan mengisi API dan Endpoint pada input text yang sudah disediakan. Setelah itu, system akan melanjutkan proses berikutnya hingga frontend ter-generate dan pengguna hanya tinggal menggunakannya pada project angular yang telah pengguna siapkan dan yang akan digunakan.



GAMBAR 4 IMPLEMENTASI INPUT API DAN ENDPOINT

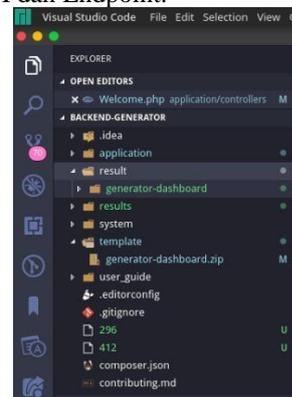
Tampilan seperti pada gambar diatas. Tampilan pada landing page terlihat sederhana karena pada gambar 5.11 tidak ada atribut untuk css sehingga *style* pada tampilan menjadi sederhana.

B. Implementasi Unzip File Template



GAMBAR 5 FLOWCHART UNZIP TEMPLATE

Pada gambar diatas, alur dari unzip file dimulai dengan pengguna membuka halaman generator lalu pengguna memasukkan API dan Endpoint.



GAMBAR 6 IMPLEMENTASI UNZIP TEMPLATE

Folder `generator-dashboard` akan muncul pada folder `result` yang nantinya akan diproses dan di *compress* lagi menjadi file `result`. `Generator-dashboard` adalah folder utama yang menyimpan beberapa folder dan file penting seperti folder `component` untuk presentational component, folder `component` untuk smart component, folder `model` sebagai constructor dari project, file `module` untuk menyimpan module-module yang digunakan dalam project, file `service` digunakan untuk menghubungkan antara frontend dengan backend. File `service` sebagai jembatan dari frontend dan backend untuk bertukar data.

V. PENGUJIAN

Untuk menguji kesiapan sistem generator ini, dilakukan pengujian blackbox

A. Pengujian Blackbox

Pengujian blackbox adalah pengujian yang dilakukan ketika 1 fitur selesai dibuat. Pengujian ini akan dilakukan untuk memeriksa apakah fungsi sudah berjalan dengan benar atau belum.

VI. KESIMPULAN DAN SARAN

Setelah revisi teks telah usai, konten makalah Anda siap dimasukkan ke dalam *template* ini. Simpan dokumen dengan aturan penamaan dokumen: [nama penulis pertama]-[afiliasi]-[judul makalah]-[siap].doc. Contoh nama dokumen: Arie Rachmad Syulistyo-Pengenalan Iris Mata-Polinema-siap.doc.

A. Kesimpulan

Kesimpulan yang diperoleh dari Desain Prototype Frontend Auto Generator Based on REST API adalah dengan menggunakan generator, hasil dari generator dapat membantu developer bekerja lebih cepat dibandingkan dengan ketika developer harus menuliskan kode tersebut secara manual. Tingkat error dari kesalahan penulisan juga dapat diminimalisir dengan menggunakan generator dibandingkan dengan ketika developer menuliskan kode secara manual. Pada generator ini, memiliki satu kekurangan yaitu ketika API dan Endpoint masih kosong, generator akan memunculkan error dan proses generate akan menjadi salah. Ketika API dan Endpoint diisi, generator dapat menjalankan semua fungsinya tanpa ada kendala

B. Saran

Berdasarkan penelitian yang dilakukan dapat diajukan saran, yaitu dapat menggunakan full javascript agar lebih mempermudah pengguna ketika ingin menggunakannya dan menjadikan generator ini menjadi sebuah library untuk Angular untuk mempermudah instalasi setelah template selesai diproses. Selain itu, proses pada generator dibuat lebih rapi sehingga generator terbuka tanpa error ketika API dan Endpoint masih kosong.

DAFTAR PUSTAKA

- [1] Luburic, nikola., savic, goran., milosavljevic, gordana., seldinac, gilana., silvika, jelena. 2016. A code generator for building front-end tier of REST-based rich client web applications.
- [2] M. Kowal, Bryan., R. Schreier, Travis., T. Dauer, Joseph., Helikar, Tomas. 2016. Programmatic access to logical models in the Cell Collective modeling environment via a REST API.
- [3] Gani, Lutfi, Menguasai AngularJS Untuk Membuat Website Dinamis, Lokomedia, 2017.
- [4] NodeJS "About NodeJS" [Online]. Available: <https://nodejs.org/en/about/>
- [5] Typescript "Typescript" [Online]. Available: <https://typescriptlang.org/>
- [6] ExpressJS (2017) "Express" [Online]. Available: <https://expressjs.com/>
- [7] ExpressJs Routing "Basic Routing" [Online]. Available: <https://expressjs.com/en/starter/basic-routing.html>

- [8] ExpressJS Routing "Routing" [Online]. Available: <https://expressjs.com/en/guide/routing.html>
- [9] MySQL (2019) "What is MySQL?" [Online]. Available: <https://intip.in/JLwR>.
- [10] MySQL "Mysql" [Online]. Available: <https://www.openhub.net/p/mysql>