

Search Engine Get Application Programming Interface

M. Fuadi Aziz Muri¹⁾, Hendrik Setyo Utomo²⁾, Rabini Sayyidati³⁾

¹⁾²⁾³⁾Jurusan Teknik Informatika, Politeknik Negeri Tanah Laut
Jl. A Yani Km 6 Pelaihari Tanah Laut Kalimantan Selatan
Telepon (0512) 21537; Faksimile (0512) 21537

¹⁾fuadimuri@gmail.com

²⁾hendrik.tomo@politala.ac.id

³⁾rabini.sayyidati@politala.ac.id

Abstrak

Application Programming Interface (API) adalah sebuah konsep fungsi yang dapat dipanggil oleh program lain. API berkerja menjadi penghubung yang mempersatukan berbagai aplikasi dari berbagai jenis *platform*, biasa dikenal dengan nama publik API. publik API sudah banyak tersebar, sedangkan penggunaanya yaitu *paraprogrammer* yang ingin mencari publik API harus menelusuri melalui berbagai cara seperti mesin pencari umum, dokumentasi repositori ataupun langsung di artikel *web*. Pengguna belum memiliki suatu sistem yang khusus untuk mengumpulkan publik-publik API, sehingga pengguna kesulitan dalam melakukan pencarian *link* publik API. Solusi dari permasalahan tersebut dapat diselesaikan dengan membangun sebuah kerangka *web* dengan antarmuka *search engine* yang menyediakan pencarian khusus publik-publik API tersebut, sehingga para pengguna dapat mencari publik API dengan lebih mudah. *Web Service* adalah API yang dibuat untuk mendukung interaksi antar dua atau lebih aplikasi yang berbeda melalui jaringan. *Representational State Transfer (ReST)* adalah salah satu aturannya.

Kata kunci: *Search Engine, Application Programming Interface, Publik API*

Abstract

Application Programming Interface (API) is a function concept that can be called by other programs. The API works as a link that unites various applications of various types of platforms, commonly known as API public names. The public API has been widely spread, while its users, programmers who want to search for public APIs, must browse through various methods such as general search engines, repository documentation or directly in web articles. The user does not yet have a system specifically for collecting public-public APIs, so that users have difficulty in performing API public link searches. The solution to these problems can be solved by building a web framework with a search engine interface that provides specific public-public searches for the API, so that users can search the API public more easily. *Web Service* is an API that is made to support the interaction between two or more different applications through a network. *Representational State Transfer (ReST)* is one of the rules.

Keywords: *Search Engine, Application Programming Interface, API Public*

1. PENDAHULUAN

Application Programming Interface (API) memberikan konsep fungsi antarmuka pemrograman aplikasi, yang menjadi salah satu cara agar suatu aplikasi dapat diakses dan dimanfaatkan oleh pihak lain dansaling komunikasi antar sistem meskipun berbeda *platform*. *Web Service* sebagai API yang berperan dalam memberikan akses tersebut dalam proses pengambilan data. Melalui arsitektur *Representational State Transfer (ReST)* yang

dioperasikan melalui *Hypertext Transfer Protocol* (HTTP), berisikan sebuah *file Java Script Object Notation* (JSON). *File* tersebut yang akan disajikan kepada para pengguna saat mengakses API (Warsito, Ananda, & Triyanjaya, 2017).

API yang tersedia menjadi *web service* biasa disebut sebagai publik API. *Programmer* yang ingin mencari publik API harus menelusuri melalui berbagai cara seperti mesin pencari umum, dokumentasi repositori ataupun langsung di artikel *web*. Pengguna belum memiliki suatu sistem yang khusus untuk mengumpulkan publik-publik API, sehingga pengguna kesulitan dalam melakukan pencarian *link* publik API. Solusi dari permasalahan tersebut dapat diselesaikan dengan membangun sebuah kerangka *web* dengan antarmuka *search engine* yang menyediakan pencarian khusus publik-publik API tersebut, sehingga para pengguna dapat mencari publik API dengan lebih mudah. Pengujian yang akan penulis untuk memastikan bahwa fungsional sistem berjalan dengan baik serta kesesuaian masukan dan keluaran, penulis akan menggunakan desain pengujian *black box*.

2. TINJAUAN PUSTAKA

2.1 *Search Engine*

Search engine didesain untuk membantu pengguna untuk mencari informasi yang tersimpan di situs-situs lain. *Search engine* ibarat kendaraan yang digunakan pengguna untuk menjelajahi jagad raya dunia maya (Santoso, Sundawa, & Azhari, 2016). *Search engine* atau mesin pencari adalah *website* yang mengumpulkan dan mengorganisir konten dari seluruh bagian internet. Pengguna memasukkan apa yang ingin dicari dan mesin pencari akan memberikan berbagai *link* ke konten yang sesuai dengan apa yang dicari. Mesin-mesin pencari menggunakan program yang sering disebut dengan istilah *spiders*, *robots*, atau *crawlers* untuk mencari konten dari seluruh sudut internet. *Search Engine* yang penulis bangun berbeda, karena tidak menggunakan program *spiders*, *robots* ataupun *crawlers*, melainkan menggunakan DBMS *MySQL*.

2.2 *Application Programming Interface*(API)

Application Programming Interface (API) adalah konsep fungsi antarmuka pemrograman aplikasi, yang menjadi salah satu cara agar suatu aplikasi dapat diakses dan dimanfaatkan oleh pihak lain tanpa mengubah struktur kode utama maupun *database* sistem, serta memudahkan komunikasi antar sistem meskipun berbeda *platform* (Wijonarko & Mulya, 2018). *Web Service* adalah API yang berperan dalam memberikan akses pengguna dalam proses pengambilan data. Melalui arsitektur *Representational State Transfer* (ReST) yang dioperasikan melalui *Hypertext Transfer Protocol* (HTTP), berisikan sebuah *file Javascript Object Notation* (JSON), *file* tersebut yang akan disajikan kepada para pengguna saat mengakses API (Warsito et al., 2017).

2.3 *Database*

Database atau basis data adalah tempat menyimpan koleksi data yang terorganisir yang terdiri dari skema, tabel, *view*, *query*, *storeprocedure* dan objek-objek lainnya (Chan, 2017). *Database* atau basis data bisa disebut sebagai sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. *Database* sebagai media untuk menyimpan data agar dapat diakses dengan mudah dan cepat (Sukamto & Shalahuddin, 2016). *Database* juga menjadi sebuah tempat untuk menyimpan data yang jenisnya beraneka ragam. Keuntungan menyimpan data di *database* adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk tabel (Winarno, Zaki, & SmitDev Community, 2014).

2.4 *Entity Relationship Diagram*(ERD)

Entity Relationship diagram (ERD) merupakan pemodelan awal basis data yang paling banyak digunakan. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika (Sukamto & Shalahuddin, 2016). ERD digunakan untuk pemodelan basis data relasional.

2.5 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual, juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. *Use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sukamto & Shalahuddin, 2016).

2.6 Black Box

Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Suatu kasus pengujian yang baik adalah apabila pengujian tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap. Salah satu dari jenis pengujian yang ada adalah pengujian Black Box. Desain Pengujian Black Box dilaksanakan untuk memastikan bahwa fungsional sistem berjalan dengan baik serta kesesuaian masukan dan keluaran (Mustaqbal, Firdaus, & Rahmadi, 2015).

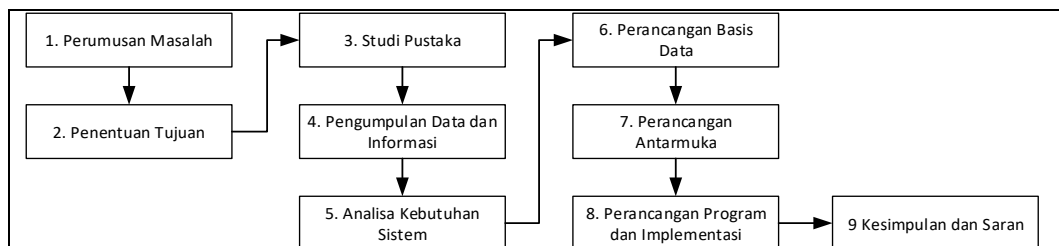
2.7 Response Time

Pengujian *response time* adalah waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output/deliverables* tertentu. *Response time* juga bisa disebut sebagai *delay* rata-rata antara transaksi dan *respon* dari standar transaksi tertentu (Mumpuni & Dewa, 2017).

3. METODE PENELITIAN

3.1 Kerangka Penelitian

Kerangka Penelitian yang dilakukan, sebagai berikut:



Gambar 1. Kerangka Penelitian

Gambar 1 adalah gambaran kerangka penelitian. Tahapan-tahapan kerangka penelitian, sebagai berikut:

1. Perumusan Masalah
Tahap pertama adalah merumuskan masalah dari pembuatan *Search Engine GetApplication Programming Interface*.
2. Penentuan Tujuan
Tahap kedua setelah merumuskan masalah adalah menentukan tujuan dibuatnya *Search Engine GetApplication Programming Interface*.
3. Studi Pustaka
Penulis mempelajari dan melakukan studi literatur dengan cara membaca buku, jurnal, artikel dan situs-situs yang berkaitan dengan pengumpulan data untuk membangun *Search Engine GetApplication Programming Interface*.

4. Pengumpulan Data dan Informasi
Setelah penulis mempelajari dan melakukan studi literatur, dilakukan pengumpulan data dan informasi yang berkaitan dengan pembuatan *Search Engine GetApplication Programming Interface*.
5. Analisa Kebutuhan Sistem
Tahap selanjutnya adalah analisa kebutuhan sistem, apa-apa saja komponen yang diperlukan untuk membangun *Search Engine GetApplication Programming Interface*, terdiri dari perangkat keras, perangkat lunak.
6. Perancangan Basis Data
Tahap berikutnya adalah merancang basis data yang diperlukan untuk membangun *Search Engine GetApplication Programming Interface*.
7. Perancangan Antarmuka
Setelah merancang basis data, penulis melakukan perancangan antarmuka *Search Engine Get Application Programming Interface* menggunakan *Microsoft Visio 2019*.
8. Perancangan Program dan Implementasi
Tahap perancangan terakhir adalah merancang program dan implementasi dari *Search Engine Get Application Programming Interface*.
9. Kesimpulan dan Saran
Tahapan terakhir dari alur penelitian adalah membuat kesimpulan dan saran dari hasil pembangunan *Search Engine Get Application Programming Interface*.

3.2 Tahap Pengumpulan Data

Pengumpulan data terdiri dari beberapa metode, sebagai berikut:

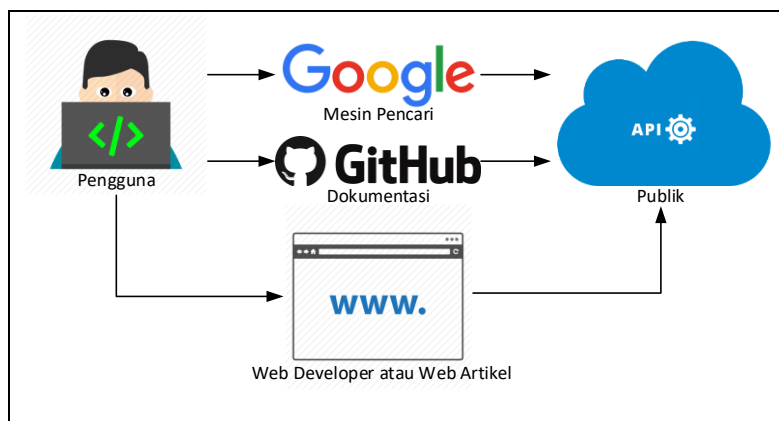
1. Metode Pustaka
Penulis mempelajari dan melakukan studi literatur dengan cara membaca buku, jurnal, artikel dan situs-situs yang berkaitan dengan pengumpulan data untuk membangun *Search Engine GetApplication Programming Interface*.
2. Metode Kuisisioner
Penulis membuat sebuah *formonline*, hasilnya berupa pengguna dapat memberikan masukan berupa penilaian saran dan kritik agar *Search Engine GetApplication Programming Interface* semakin berkembang.

4. PEMBAHASAN

Berikut adalah Pembahasan analisis sistem yang berjalan sampai dengan implementasi aplikasi.

4.1 Analisis Sistem yang Berjalan

Analisis sistem yang berjalan dapat dilihat pada Gambar 2.



Gambar 2 Sistem yang Berjalan

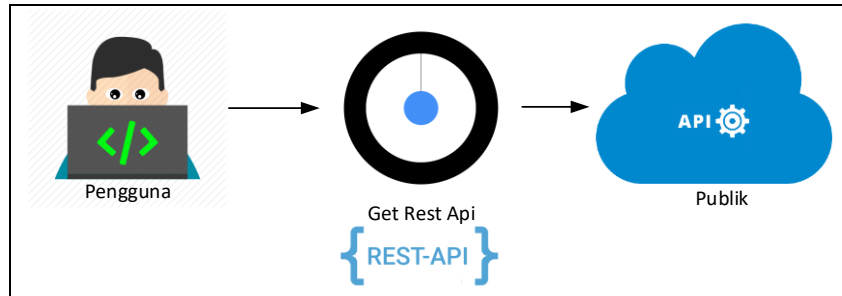
Gambar 2 adalah sistem yang sedang berjalan, pengguna mencari Publik API melalui tiga cara yang berbeda-beda, diantaranya:

1. Mesin Pencari, contohnya : *Google, Bing*.
2. Dokumentasi Repositori, contohnya : *GitHub*.
3. *Web Developer* atau *Web Artikel*.

Pengguna belum memiliki suatu sistem yang khusus untuk mengumpulkan publik-publik API, sehingga pengguna kesulitan dalam melakukan pencarian *link* publik API

4.2 Analisis Sistem yang Diusulkan

Analisis sistem yang diusulkan dapat dilihat pada Gambar 3.

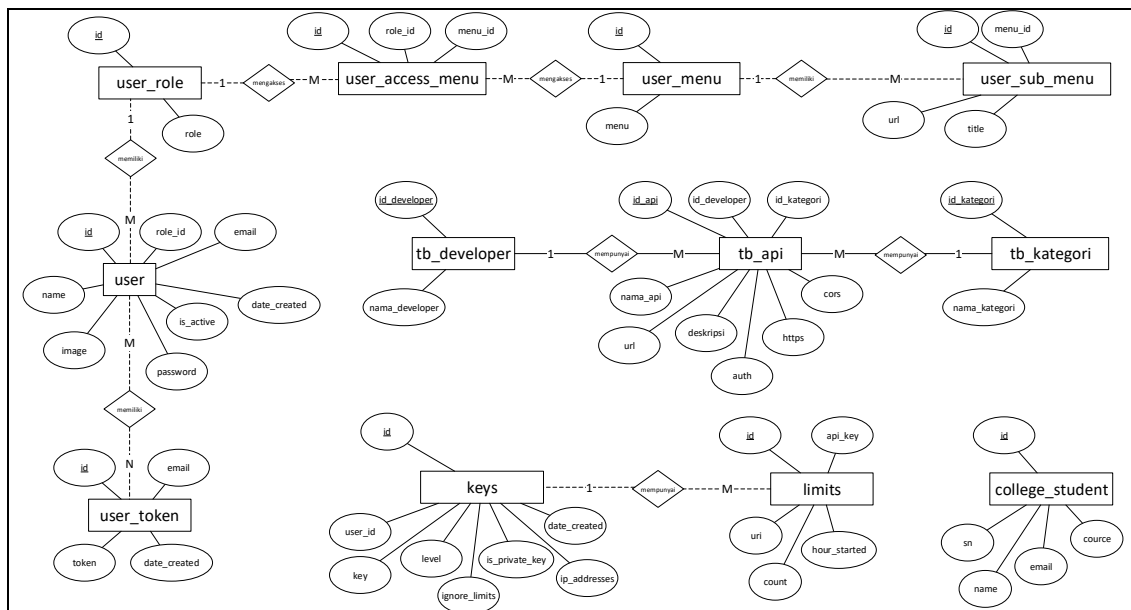


Gambar 3 Sistem yang Diusulkan

Gambar 3 adalah sistem yang diusulkan oleh penulis, pengguna mencari Publik API cukup melalui *Search Engine GetApplication Programming Interface*, atau *Get Rest API*. Pengguna terbagi menjadi tiga jenis, *Free user*, *member* dan *administrator*. *Free user* yang yang berbagung kedalam *Get Rest API* akan menjadi *member*. *Member* dapat melakukan uji coba ReSTfull API yang telah disediakan oleh *administrator*

4.3 Entity Relationship Diagram (ERD)

Berikut adalah rancangan ERD pada *Search Engine Get Application Programming Interface*, dapat dilihat pada gambar 4.

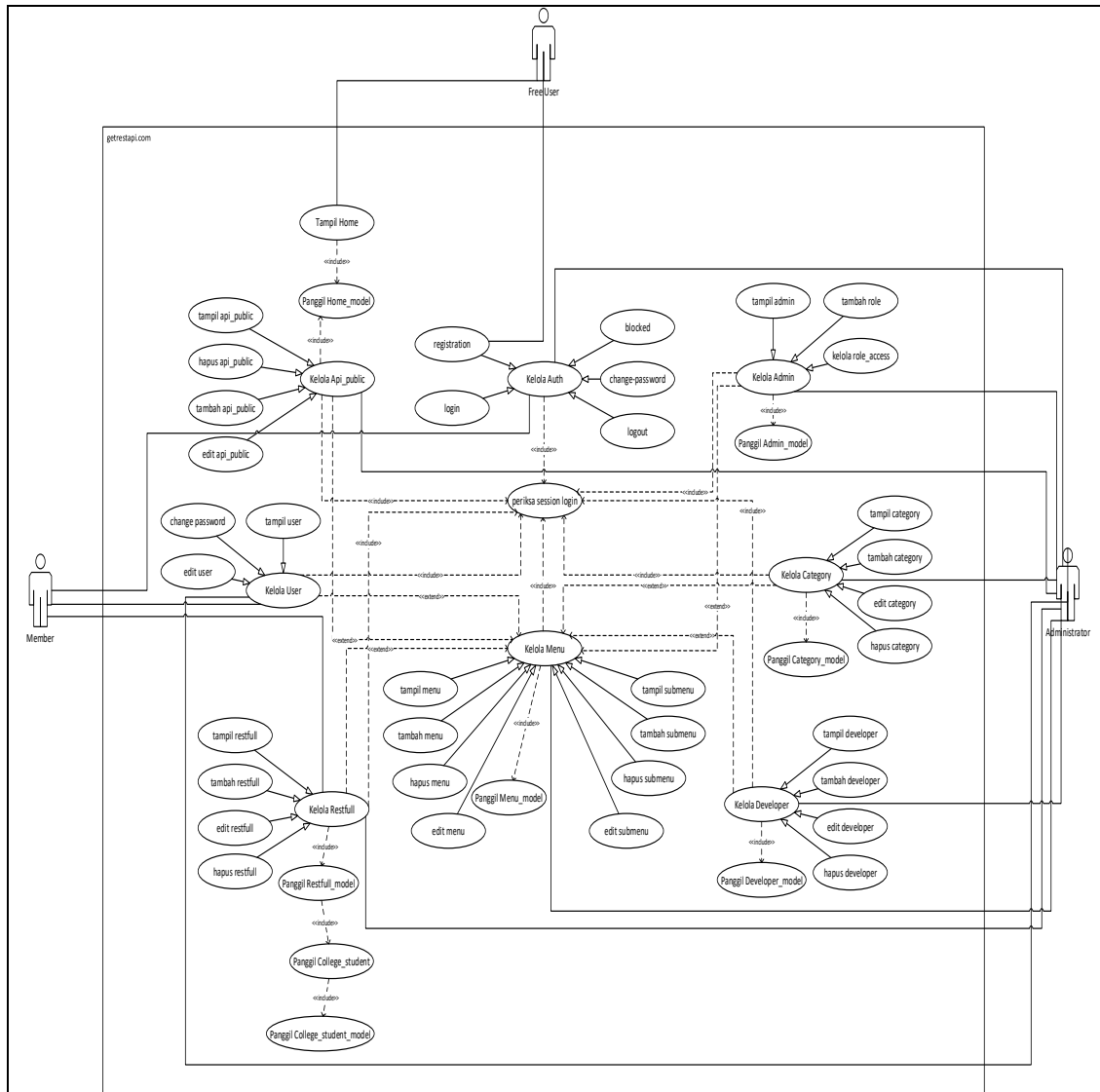


Gambar 4 ERD

Gambar 4 adalah alur *database* yang terdiri atas dua belas entitas, setiap entitas mewakili nama tabel yang ada di *databaseSearch Engine Get Application Programming Interface*. Setiap atribut yang ada mewakili kolom-kolom yang ada pada *database*. Relasi yang menghubungkan entitas-entitas sesuai peranannya, bobot peranan entitas dapat dilihat pada kardinalitasnya.

4.4 Unified Modeling Language (UML)

Berikut adalah rancangan sistem UML *Use Case* pada *Search Engine Get Application Programming Interface*, dapat dilihat pada gambar 5.



Gambar 5 UML Use Case

Gambar 5 adalah UML *Use Case* yaitu pondasi dalam pembuatan UML. *Search Engine Get Application Programming Interface* memiliki tiga jenis pengguna yaitu *administrator*, *member* dan *free user*, masing-masing memiliki jumlah aliran data yang berbeda.

4.5 Implementasi Aplikasi

Berikut adalah implementasi dari *Search Engine Get Application Programming Interface*, dapat dilihat pada gambar 6berikut.



Gambar 6 Implementasi Antarmuka: a. Home, b. Detail Home, c. Sign In, d. Sign Up, e. Admin, f. User, g. API Public, h. Detail API Public, i. Insert API Public, j. Update API Public

4.5.1 Antarmuka Home

Gambar 6 a. adalah antarmuka Home pengguna yang mengakses halaman tersebut adalah *free user*. Terdapat form pencarian untuk mencari publik API baik berdasarkan nama, kategori maupun *developer*-nya, pada bagian atas halaman, dan pada bagian bawah terdapat card yang masing-masingnya adalah sebuah publik API.

4.5.2 Antarmuka Detail Home

Gambar 6 b. adalah antarmuka Detail Home. Jika salah satu card pada halaman awal sebelumnya dipilih tombol detail, maka halaman detail Home akan terbuka. Terdapat card yang berisikan detail suatu Publik API, diantaranya informasi mengenai *name*, *developer*, *description*, *auth*, *https*, *cors* dan *category*. Tombol *Get This API* jika dipilih akan mengarahkan pengguna menuju link publik API bersangkutan.

4.5.3 Antarmuka Sign In

Gambar 6 c. adalah antarmuka *Sign In*, pada halaman tersebut administrator ataupun member dapat melakukan login untuk masuk kedalam *Get Rest API* yang memiliki fitur lebih dibanding *free user*. Cukup memasukkan *email* dan *password* pada *form* kemudian pilih tombol *login*.

4.5.4 Antarmuka Sign Up

Gambar 6 d. adalah antarmuka *Sign Up*, pada halaman tersebut *free user* melakukan registrasi untuk menjadi member. Caranya dengan memasukkan data pada *form* berupa Nama lengkap, *email* aktif, dan *password* minimal delapan karakter dua kali, kemudian pilih tombol *registration*. *GetRestAPI* akan mengirimkan *email* verifikasi kepada *email* yang telah dimasukkan tadi. *Freeuser* yang melakukan *verify* sudah menjadi member.

4.5.5 Antarmuka Admin

Gambar 6 e. adalah antarmuka *Admin*, pengguna yang dapat mengakses halaman tersebut hanyalah *administrator*. Terdapat perhitungan jumlah publik *API* dan jumlah member yang sudah bergabung pada *Get Rest API*.

4.5.6 Antarmuka User

Gambar 6 f. adalah antarmuka *User*. *Administrator* dan member berhak mengakses halaman tersebut sesuai profilnya masing-masing. Isi *profile* terdiri atas Nama lengkap, *email*, foto *profile* dan tanggal saat registrasi.

4.5.7 Antarmuka API Public

Gambar 6 g. adalah antarmuka *API Public*. Pengguna yang dapat mengakses halaman tersebut hanya *administrator*. *Administrator* dapat mengelola publik *API* yang tampil pada halaman Home/index melalui halaman tersebut. Diantaranya melakukan pencarian, melihat detail sesuai nama publik *API* atau melakukan *insert*.

4.5.8 Antarmuka Detail API Public

Gambar 6 h. adalah antarmuka Detail *API Public*. Jika salah satu detail pada *Api_public/index* sebelumnya dipilih, maka halaman detail Publik *API* akan terbuka. Terdapat *card* yang berisikan detail suatu Publik *API*, diantaranya informasi mengenai *name*, *developer*, *description*, *auth*, *https*, *cors* dan *category*. Pada halaman tersebut terdapat tombol *Delete* dan *Update*.

4.5.9 Antarmuka Insert API Public

Gambar 6 i. adalah antarmuka *Insert API Public*, terdapat *form* pengisian Publik *API* baru yang dapat dimasukkan oleh *administrator*. Isinya berupa nama *API*, *developer*, kategori, *auth*, *https*, *cors*, deskripsi dan *urlAPI*. Setelah selesai pilih tombol *Insert* untuk menyimpan perubahan.

4.5.10 Antarmuka Update API Public

Gambar 6 j. adalah antarmuka *Update API Public*, terdapat *form* pengedit-an Publik *API* lama yang dapat diubah oleh *administrator*. Isinya berupa nama *API*, *developer*, kategori, *auth*, *https*, *cors*, deskripsi dan *urlAPI*. Setelah selesai pilih tombol *Update* untuk menyimpan perubahan.

4.6 Pengujian Black Box

Waktu : 25/06/2019 9:59 WITA s/d 26/06/2019 6:52 WITA
Alat bantu pengujian : *Google Form*
Alamat : <https://forms.gle/r5ZAJ1W8q1AAg7yv5>

Tabel 1. Pengujian *black box*

No.	Nama Pengujian	Hasil	Keterangan
1	Fitur pencarian <i>Public API</i>	Berhasil	Fungsional fitur pencarian dan <i>get link</i> berhasil
2	Isi dari <i>menu Learn</i>	Berhasil	Fungsional fitur pada <i>menu learn</i> berhasil
3	Fitur <i>Disqus</i>	Berhasil	Fungsional fitur mengirim komentar, saran, kritik berhasil
4	<i>Sign Up</i>	Berhasil	Fungsional fitur registrasi melalui validasi <i>email</i> berhasil
5	<i>Sign In</i>	Berhasil	Fungsional fitur <i>login</i> berhasil
6	<i>Forgot Password</i>	Berhasil	Fungsional fitur lupa <i>password</i> melalui validasi <i>email</i> berhasil
7	Fitur isi <i>menu User</i>	Berhasil	Fungsional dari fitur CRUD pada <i>menu User</i> berhasil
8	Fitur isi <i>menu API Restfull</i>	Berhasil	Fungsional dari <i>menu API Restfull</i> berhasil

4.7 Pengujian *Response Time*

1. *Desktop*

Spesifikasi *Device* : Asus A450CC (RAM 4GB)

Spesifikasi *Browser* : *Google Chrome* 72.0.3626.109 64 bit

Alat bantu pengujian : *Google PageSpeed Insights*

Alamat : <https://developers.google.com/speed/pagespeed/insights/>

2. *Mobile*

Spesifikasi *Device* : *Smartphone* Xiaomi Redmi 2 (RAM 1GB)

Spesifikasi *Browser* : *Google Chrome* 75.0.3770.101

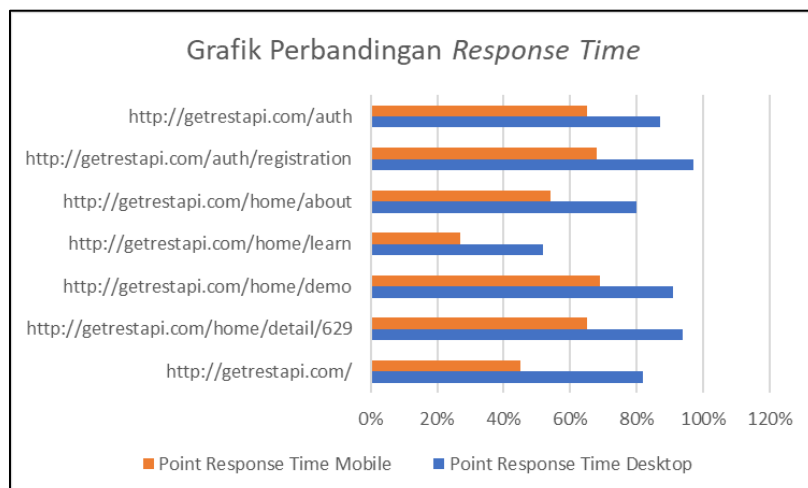
Alat bantu pengujian : *Google PageSpeed Insights*

Alamat : <https://developers.google.com/speed/pagespeed/insights/>

Tabel 2. Pengujian *response time*

No.	URL	<i>Point Response Time</i>	
		<i>Desktop</i>	<i>Mobile</i>
1	http://getrestapi.com/	82%	45%
2	http://getrestapi.com/home/detail/629	94%	65%
3	http://getrestapi.com/home/demo	91%	69%
4	http://getrestapi.com/home/learn	52%	27%
5	http://getrestapi.com/home/about	80%	54%
6	http://getrestapi.com/auth/registration	97%	68%
7	http://getrestapi.com/auth	87%	65%

Grafik perbandingan *response time* dapat dilihat pada Gambar 7 berikut.



Gambar 7 Grafik Perbandingan *Response Time*

5. KESIMPULAN

Kesimpulan yang dapat penulis kemukakan, sebagai berikut:

- Telah dibangunnya *Search Engine Get Application Programming Interface* yang dapat bermanfaat dan mudah digunakan oleh penggunanya yaitu para *programmer*, dalam mencari alamat publik API.
- Tercapainya pengujian *Search Engine Get Application Programming Interface* menggunakan *black box* dan *response time*.

DAFTAR PUSTAKA

- Chan, S. (2017). *Membuat Aplikasi Database dengan PowerBuilder 12.6 dan MySQL*. Jakarta: PT. Elex Media Kumputindo.
- Mumpuni, I. D., & Dewa, W. A. (2017). Analisis Dan Pengembangan Sistem Self Services Terminal (SST) Dengan Pendekatan PIECES Pada STMIK Pradnya Paramita Malang. *Matics*, 9(1), 12. <https://doi.org/10.18860/mat.v9i1.4127>
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *JITTER (Jurnal Ilmiah Teknologi Informasi Terapan)*, 1(3), 34. <https://doi.org/ISSN : 2407 - 3911>
- Santoso, B. W., Sundawa, F., & Azhari, M. (2016). Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Engine) Berbasis Web Pada Database. *Jurnal Sisfotek Global*, 6(1), 1–8.
- Sukanto, R. A., & Shalahuddin, M. (2016). *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: INFORMATIKA.
- Warsito, A. B., Ananda, A., & Triyanjaya, D. (2017). Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service. *Technomedia Journal*, 2(1), 26–36. <https://doi.org/10.33050/tmj.v2i1.313>
- Wijonarko, D., & Mulya, B. W. R. (2018). Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang. *SMATIKA JURNAL*. <https://doi.org/10.32664/smatika.v8i02.202>
- Winarno, E., Zaki, A., & SmitDev Community. (2014). *24 Jam Belajar PHP*. Jakarta: Elex Media Komputindo.

Biodata Penulis



M. Fuadi Aziz Muri, kelahiran Tanah Laut 22 Oktober 1998. Saat artikel ini ditulis, penulis berstatus mahasiswa di Politeknik Negeri Tanah Laut. Penulis memulai perkuliahan di jurusan Teknik Informatika sejak tahun 2016 dan meraih gelar Ahli Madya (A.Md) pada tahun 2019.



Hendrik Setyo Utomo, meraih gelar Sarjana Teknik (S.T) Jurusan Teknik Informatika dari Universitas Gunadarma dan menyelesaikan Magister Manajemen Sistem Informasi (M.M.S.I) pada Program Studi Manajemen Sistem Informasi dari Universitas Gunadarma. Menjadi dosen Program Studi Teknik Informatika di Politeknik Negeri Tanah Laut sejak 2012 sampai dengan sekarang (2019).



Rabini Sayyidati, lahir di Banjarmasin 5 Mei 1989. Penulis memperoleh gelar S.Pd dalam bidang pendidikan sejarah di FKIP Universitas Lambung Mangkurat pada tahun 2013, kemudian melanjutkan S2 di jurusan pendidikan IPS di FKIP Universitas Lambung Mangkurat dan memperoleh gelar M.Pd pada tahun 2015. Penulis kemudian menjadi dosen di Jurusan Teknik Informatika Politeknik Negeri Tanah Laut sejak 2015 sampai dengan sekarang (2019).