

PENERAPAN FUNGSI TRANSFORMING DAN RATE LIMITING UNTUK MANAGEMENT API DI PERUSAHAAN

M. Ainurrahman^{1*}, Siswanto²

^{1,2} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta Selatan, Indonesia

Email: ^{1*}1611510015@student.budiluhur.ac.id, ²siswanto@budiluhur.ac.id
(* : corresponding author)

Abstrak-API telah menjadi fundamental dari ekosistem perangkat lunak, menyediakan berbagai macam metode untuk mendorong suatu kemajuan dan inovasi didalam maupun diluar instansi atau organisasi dalam menghadirkan suatu sistem yang terintegrasi. Pertama, penulis membangun dan mengkonfigurasi API menggunakan tools management API. Kemudian, penulis mengimplementasikan fungsi Transforming pada API untuk mengubah struktur data yang diterima dari aplikasi sumber ke format yang dibutuhkan oleh API tujuan. Hal ini memungkinkan integrasi yang lebih mudah antara aplikasi sumber dan API tujuan. Selain itu, penulis menerapkan *Rate Limiting* pada API untuk mengatur jumlah permintaan yang diterima dari aplikasi sumber. Dengan mengatur batas permintaan, kita dapat menghindari *overload* sistem dan mempertahankan kualitas layanan. penulis melakukan pengujian kinerja untuk memverifikasi efektifitas implementasi *Rate Limiting* dalam mengontrol lalu lintas API dengan . Hasil penelitian ini menunjukkan bahwa implementasi API Management pada fungsi *Transforming* dan *Rate Limiting* dapat berhasil dilakukan menggunakan *platform web* dan *mobile* dengan *Error Rate* 0%. Fungsi *Transforming* memungkinkan adaptasi data yang lebih fleksibel antara aplikasi sumber dan API tujuan. Sementara *Rate Limiting* membantu menjaga stabilitas dan kinerja sistem. Penelitian ini memiliki dampak penting dalam pengembangan aplikasi dan integrasi sistem, karena memberikan panduan praktis untuk mengelola API dengan efisien. Diharapkan bahwa hasil penelitian ini dapat membantu pengembang dan professional IT dalam mengimplementasikan management API yang lebih baik pada proyek-proyek di masa depan dan membantu perusahaan dalam hal ini adalah PT. Administrasi Medika yang mempunyai permasalahan dalam hal integrasi sistem dengan *client*, yaitu masih melakukan patching API secara manual ketika ada client yang melakukan integrasi ke fitur yang sudah ada tapi dengan struktur yang berbeda dan belum bisa mengontrol jumlah *request* API dari *client* sehingga potensi sistem mengalami *down* menjadi sangat besar, sehingga solusi *Transforming* dan *Rate Limiting* menjadi hal disarankan untuk sistem atau API dengan *load* tinggi atau dengan data yang sangat dinamis.

Kata Kunci: *Webservice, API Management, Transforming, Rate Limiting, Mobile, Web*

IMPLEMENTATION OF TRANSFORMING AND RATE LIMITING FUNCTIONS FOR API MANAGEMENT IN THE COMPANY

Abstract- API has become the fundamental of the software ecosystem, providing various methods to encourage progress and innovation within and outside the agency or organization in presenting an integrated system. First, the author builds and configures the API using the API management tools. Then, the author implements the Transforming function in the API to change the data structure received from the source application to the format required by the destination API. This allows for easier integration between source applications and destination APIs. In addition, the authors implement Rate Limiting on the API to regulate the number of requests received from the source application. By setting request limits, we can avoid system overload and maintain service quality. the author performs performance testing to verify the effectiveness of the implementation of Rate Limiting in controlling API traffic with . The results of this study indicate that the implementation of API Management in the Transforming and Rate Limiting functions can be successfully carried out using web and mobile platforms with an error rate of 0%. The Transforming function enables more flexible adaptation of data between source applications and destination APIs. Meanwhile Rate Limiting helps maintain system stability and performance. This research has an important impact on application development and system integration, as it provides a practical guide for efficiently managing APIs. It is hoped that the results of this research can help developers and IT professionals implement better API management on future projects and help companies in this case, namely PT. The Medical Administration has problems with system integration with clients, which is still patching the API manually when a client integrates with existing features but with a different structure and has not been able to control the number of API requests from the client so that the potential for the system to experience downtime becomes is very large, so Transforming and Rate Limiting solutions are recommended for systems or APIs with high loads or with very dynamic data.

Keywords: *Webservice, API Management, Transforming, Rate Limiting, Mobile, Web*

1. PENDAHULUAN

Pada era saat ini, teknologi informasi berkembang sangat pesat dan dapat di rasakan dari tahun ke tahun tumbuh kembangnya. Perkembangan ini dapat kita lihat di dunia; kesehatan, pertanian, pendidikan, dan masih banyak lagi jika ingin disebutkan satu persatu. tanpa kita sadari, teknologi informasi banyak membantu kehidupan manusia dalam menyelesaikan pekerjaan manusia itu sendiri [1]. Selain itu, teknologi informasi juga dapat menghemat waktu sehingga kita dapat mempersingkat proses penyelesaian dalam pengerjaan pekerjaan tersebut [2]. Tentu nya dalam perkembangan teknologi tersebut harus diimbangi dengan rasa tanggung jawab baik pembuat teknologi tersebut atau yang menggunakannya agar tidak menjadi sesuatu hal yang bersifat negatif dalam penggunaannya.

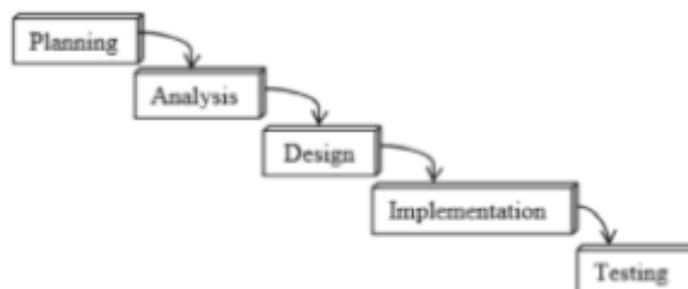
PT Administrasi Medika atau yang lebih dikenal dengan nama AdMedika didirikan pada tahun 2002, merupakan salah satu perusahaan yang memanfaatkan teknologi informasi bergerak di bidang asuransi yang menyediakan layanan administrasi pihak ketiga (*Third Party Administration/TPA*) menjadi penyedia layanan kesehatan menyeluruh [3]. Layanan kesehatan ini terbagi menjadi tiga kategori utama, yaitu Manajemen Klaim Kesehatan (*Health Claim Management*), aplikasi MyAdMedika, serta Layanan Manajemen Fasilitas Kesehatan (*Provider Management Service*). Masing-masing kategori bercabang menjadi berbagai layanan dengan satu tujuan memberikan akses layanan kesehatan bagi *client*. Tentu saja ini sangat membantu *client*, sehingga memudahkan *client* dalam mengakses layanan kesehatan serta dari mana pun dan kapan pun [4].

Seiring dengan berjalannya layanan kesehatan ini, beberapa permasalahan mulai muncul. Dimana saat *client* secara berkala atau terus menerus mengakses layanan kesehatan tersebut yang diberikan hak akses layanan secara penuh [5]. Maksud arti kata “penuh” yaitu, *client* bisa mendapatkan data diri *client* tanpa ada pembatasan atau limit untuk mengakses. Selain itu jika ada perubahan atau peningkatan hak akses data layanan, maka *client* akan mendapatkan hak akses tersebut tanpa ada nya konfirmasi atau kesepakatan terlebih dahulu ke pihak *client*. Hal ini membuat layanan tersebut menjadi *uncontrolled* dan *high traffic* dan agak lama dalam mendapatkan data diri untuk *client* itu sendiri. Ini di akibatkan karena banyak nya pemakaian *resource* (*bandwidth, memory, cpu*) dalam tiap jam nya serta *traffic* pengguna yang padat [6]. Tentu saja ini sangat mempengaruhi performa dalam pelayanan untuk pihak *client* maupun PT Administrasi Medika sebagai penyedia layanan kesehatan. Oleh karena itu, diperlukan suatu pembatasan atau limit akses yang berguna untuk mengurangi terjadi *down* dan mengurangi pemakaian *resource* yang berlebih. Karena PT Administrasi Medika selaku penyedia layanan memiliki tanggung jawab untuk menjaga performa akses layanan kesehatan tersebut.

Metode *Rate Limiting* dan *Transforming* dibuat untuk mengatasi masalah tersebut. *Rate Limiting* digunakan untuk membatasi hak akses layanan untuk *client*. Jika *client* telah menggunakan hak akses layanan sudah dibatas ketentuan, maka *client* tidak akan bisa mengakses layanan tersebut pada waktu tersebut dan akan dapat mengakses kembali layanan kesehatan setelah adanya kesepakatan kembali. *Tranforming* digunakan untuk menampilkan data yang diperlukan oleh *client* dengan ketentuan yang telah disepakati oleh kedua belah pihak. Sehingga *client* disuguhkan dengan data sesuai dengan apa yang dibutuhkan dan disepakati antara pihak PT Administrasi Medika dan *client*.

2. METODE PENELITIAN

Dalam mengembangkan penelitian ini, penulis menggunakan Metode *Waterfall* dengan berdasarkan kesesuaian model yang di pakai. Setiap tahap – tahap memiliki keterkaitan dalam proses pengembangannya [7]. Metode *Waterfall* dimulai dari tahap *Planning, Analysis, Design, Implementation, Testing* dan *Maintenance*. Pada Gambar 1 berikut ini :



Gambar 1. Metode *Waterfall*

- a. *Planning*
Tahapan *Planning* digunakan untuk pendefinisian tujuan, melakukan uji kelayakan teknis yang berupa ketersediaan *hardware* dan *software*. Hardware yang digunakan yaitu ; *Processor 2 GB Single Core, Memory 2 GB, SSD 50 GB* dan *Smartphone (Android / IOS)*. Sedangkan *software* yang digunakan yaitu ; OS Centos 8, *Webserver (Nginx)*, *Postgresql, NodeJs, Flutter (Dart)* dan *DNS*.
- b. *Analysis*
Tahapan *Analysis* dilakukan untuk menganalisis kebutuhan-kebutuhan yang diperlukan untuk membangun metode *Transforming* dan *Rate Limiting* di PT. Administrasi Medika yang didapatkan berupa saran dan masukan dalam pengembangan aplikasi layanan kesehatan tersebut.
- c. *Design*
Tahapan *Design* dilakukan untuk menerjemahkan kebutuhan yang sudah dianalisa dalam pengembangan aplikasi layanan kesehatan. Tahap ini meliputi perancangan struktur data, struktur sistem, masukan dan keluaran.
- d. *Implementation*
Tahap *Implementation* merupakan tahapan secara nyata dalam mengerjakan suatu aplikasi yang berguna untuk menerjemahkan desain yang telah dibuat kedalam Bahasa pemrograman. Bahasa pemrograman Dart dan NodeJs serta PostgreSQL sebagai database, merupakan salah satu yang dapat dimengerti oleh komputer.
- e. *Testing*
Tahap akhir dilakukan adalah *Testing* untuk memastikan aplikasi dapat bekerja sesuai apa yang telah direncanakan dan digunakan untuk menemukan kesalahan dan memastikan sistem akan memberikan hasil yang diinginkan. Dalam tahap ini, peneliti menggunakan aplikasi tersebut untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan apakah masih terdapat kesalahan atau tidak, dengan dilakukan penggabungan modul-modul yang telah dibuat.

2.1 Pengumpulan Data

Pengumpulan data merupakan suatu proses untuk mengumpulkan data yang dibutuhkan dalam pengembangan aplikasi ini agar hasil yang di dapatkan sesuai dengan apa yang di harapkan dan yang di butuhkan. Berikut tahap-tahap yang di lakukan adalah :

- a. Wawancara di lakukan berupa aktifitas tanya jawab kepada pihak terkait yang berguna untuk mendapatkan informasi tentang kebutuhan untuk mengembangkan aplikasi layanan kesehatan di PT. Administrasi Medika.
- b. Observasi merupakan aktifitas mengumpulkan data serta mengamati langsung apa yang terjadi di PT. Administrasi Medika yang berguna sebagai referensi atau masukan untuk mengembangkan aplikasi layanan kesehatan tersebut.
- c. Studi Pustaka dilakukan berupa aktifitas dengan cara membaca jurnal yang berkaitan dengan metode *Transforming* dan *Rate Limiting* yang di gunakan dalam pengembangan aplikasi layanan kesehatan di PT. Administrasi Medika.

2.2 API (Application Programming Interface)

API (Application Programming Interface) adalah sekumpulan instruksi program dan protokol yang digunakan untuk membangun aplikasi perangkat lunak. API berperan sebagai pembawa pesan yang menerima permintaan pengguna dan memberitahu sistem apa yang harus dilakukan, lalu memberikan respons yang sesuai untuk permintaan tersebut [8]. Prinsip yang sama juga berlaku ketika terjadi interaksi antara database aplikasi dengan pegawai atau komputer, masing-masing sistem tersebut memiliki API yang memungkinkan untuk saling bertukar informasi. Dengan kata lain API adalah apa yang membawa informasi antara database aplikasi dan device pengguna untuk menyampaikan data dan menciptakan konektivitas antar system.

2.3 Transforming

Transforming atau Transformasi adalah sebuah proses perubahan secara berangsur-angsur sehingga sampai pada tahap ultimate, perubahan yang dilakukan dengan cara memberi respon terhadap pengaruh unsur eksternal dan internal yang akan mengarahkan perubahan dari bentuk yang sudah dikenal sebelumnya melalui proses menggandakan secara berulang-ulang atau melipatgandakan [9]. *Transforming* bersifat Tipologikal (geometri) bentuk geometri yang berubah dengan komponen pembentuk dan fungsi ruang yang sama. *Transforming* bersifat gramatikal hiasan (ornamental) dilakukan dengan menggeser, memutar, mencerminkan, menjungkirbalikkan, melipat dan lain – lain. *Transforming* bersifat refersal (kebalikan) pembalikan citra pada figur objek yang akan ditransformasi dimana citra objek dirubah menjadi citra sebaliknya. *Transforming* bersifat distortion (merancukan) kebebasan perancang dalam beraktifitas.

2.4 Rate Limiting

Rate Limiting ini dapat digunakan untuk menjadikan sistem lebih aman. Ini adalah praktik untuk mengoptimalkan dan untuk membatasi permintaan yang terlalu banyak [10]. Ini memastikan bahwa tidak ada satu pun pengguna API yang menggunakan API secara berlebihan dan aksesibilitas API seragam. Itu membuat API tersedia untuk semua orang dalam batas. Karena kebutuhan organisasi berbeda satu sama lain, tidak bijaksana untuk menerapkan strategi tunggal untuk pembatasan tarif.

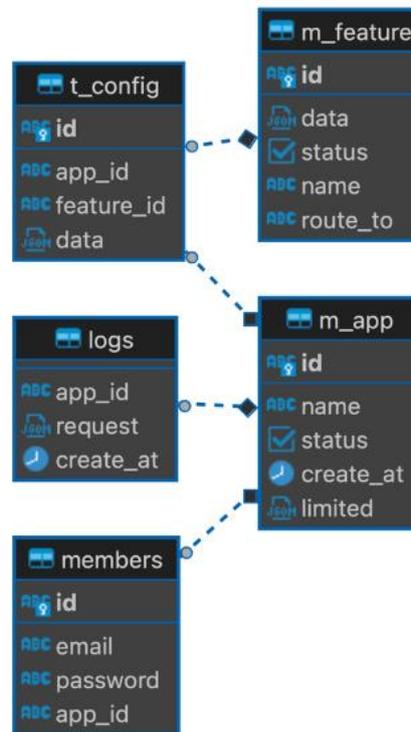
3. HASIL DAN PEMBAHASAN

Pada tahap ini merupakan implemntasi dari metode terhadap aplikasi yang sudah dibuat dan juga pengujian metode yang diterapkan. Serta melakukan uji coba pada aplikasi yang telah dibangun, apakah telah sesuai dengan perancangan dan hasil seperti yang diharapkan.

3.1 Rancangan Basis Data

3.1.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) di bawah ini adalah rancangan database yang akan dibangun.sepereti pada gambar 2 di bawah ini :



Gambar 2. Entity Relationship Diagram

3.1.2 Logical Record Structured (LRS)

Setelah transformasi *ERD (Entity Relationship Diagram)* ke dalam *LRS (Logical Record Structured)* selesai, maka dapat dibuat *LRS (Logical Record Structured)* pada gambar 3 dengan tabel relasi untuk membuat spesifikasi basis data nya.



Gambar 3. Logical Record Structured

3.2 Implementasi Transforming dan Rate Limiting

Adapun langkah-langkah implementasi metode yang dilakukan adalah sebagai berikut:

3.2.1 Integrasi Dengan API Management

Integrasi dilakukan agar terjadi sinkronisasi data ketika *client* memanggil API via API Management dan diteruskan ke Core API, pada tabel 1 API yang akan dilakukan integrasi dengan API Management:

Tabel 1. API Core dan Endpoint

Nama	Endpoint URL
API Member	v2/getMember
API Plan	v2/getPlan
API Benefit	v2/getBenefit
API History Claim	v2/getClaim

3.2.2 Mapping field

Mapping field Core API kedalam API Management dilakukan ketika untuk memberikan *treatment* yang berbeda terhadap *client* pada response API yang mereka akan dapatkan sesuai dengan kesepakatan kedua belah pihak, pada tabel 2 adalah field yang perlu di mapping terhadap API Management.

Tabel 2. API Core field

API	Field (case & format sensitive)
API Member	nik,vip,name,bpjsid,gender,remark,card_no,corp_code,help_line,member_id,policy_id,payor_code,payor_name,active_flag,agenc_code,dateofbirth,employee_id,identity_no,member_type,bank_account,dependent_id,faskes_level,member_since,type_of_room,corporate_info,payor_member_id,cancellation_date
API Plan	remark,benefit,coverage,emp_name,end_date,excess_by,max_value,plan_code,plan_type,limit_type,start_date,employee_id,member_type,dependent_id,current_limit,record_status,referral_flag,frequency_desc,family_over_limit,member_over_limit
API Benefit	card_no,mstatus,emp_name,end_date,loa_desc,available,corp_code,member_id,plan_code,payor_code,start_date,benefit_code,employee_id,max_benefit,benefit_code,dependent_id,coverage_code,condition_desc,frequency_code,frequency_desc,multiple_condition,amt_approve_asses
API History Claim	Excess,benefit,incured,remark,approved,provider,claims_id,diagnosis,claim_type,payment_date,claims_status,admission_date,dischargeable_date,refereed_op_form_no,claims_process_status

3.2.2 Membuat *Matrix* pembatasan akses

Matrix pembatasan dibuat agar admin bisa membedakan pada masing-masing tipe, yaitu pada tabel 3 sebagai berikut:

Tabel 3. *Matrix* pembatasan akses

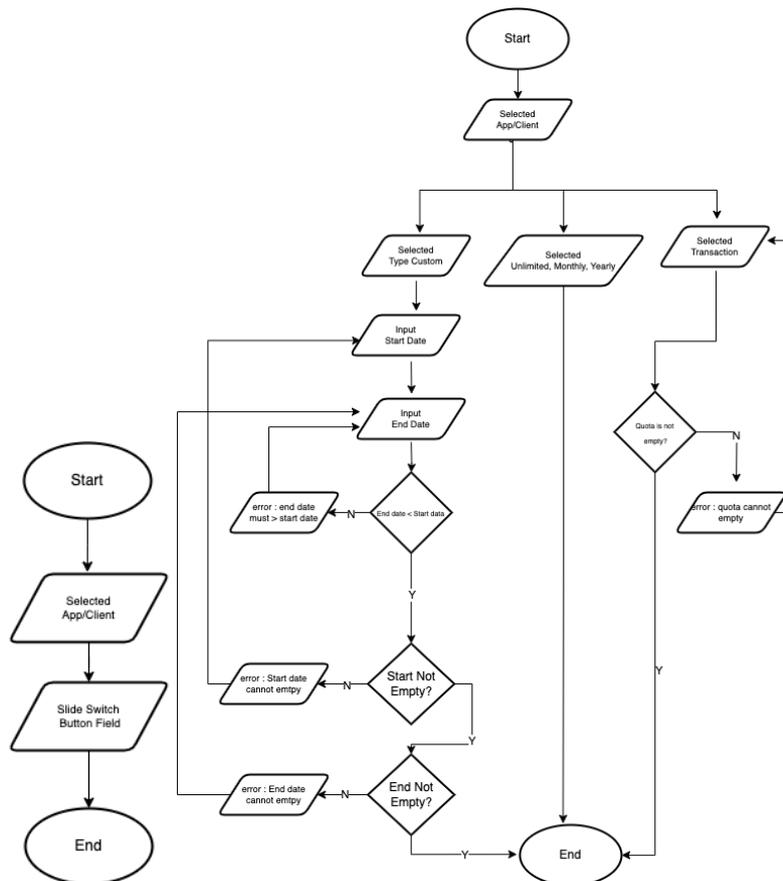
Kategori	Requirement
Unlimited	Start & End Date (Auto)
Monthly	Start & End Date (Auto)
Yearly	Start & End Date (Auto)
Transaction	Provide Quota
Custom	Start & End Date (Manual)

3.3 Flowchart

Berikut ini adalah *Flowchart* yang digunakan untuk mengetahui alur proses program pada fungsi Transforming dan Rate Limiting

3.3.1 Flowchart Rate Limiting dan Transforming

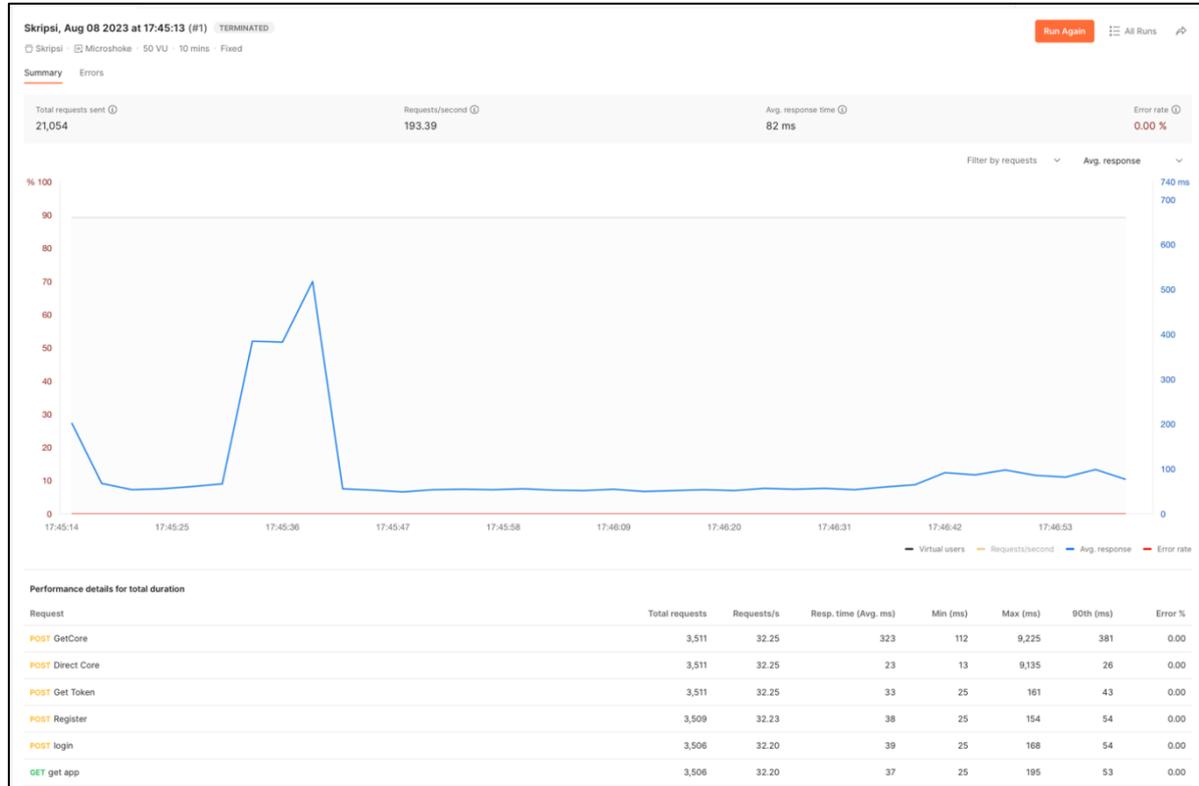
Flowchart ini merupakan proses dari konfigurasi *Rate limiting* dan Transforming terhadap masing-masing app/client yang bekerja sama dengan PT. Administrasi Medika sebagai upaya untuk mengontrol jumlah permintaan terhadap API pada gambar 4 sebagai berikut:



Gambar 4. *Flowchart* Transforming dan Rate Limiting

3.4.3 Pengujian Iterasi API

Pengujian ini dilakukan untuk mengetahui setiap data yang diintegrasikan ke aplikasi *client* berupa pengujian API *Register*, API *Update Token*, API *Login*, API *Get Clinet/App* dan Pengujian API *Get Core* seperti gambar 7 di bawah ini:



Gambar 7. Pengujian Iterasi API

4. KESIMPULAN

Setelah melakukan proses perancangan dan pengembangan aplikasi layanan kesehatan serta berdasarkan permasalahan yang telah di sebutkan pada pembahasan sebelumnya maka dapat di simpulkan bahwa metode *Rate Limiting* dan *Transforming* telah memberikan solusi rekomendasi terhadap PT. Administrasi Medika agar lebih *aware* terhadap Digitalisasi *Management API* serta menjamin *Integrasi* data atau sistem dengan *client* berjalan dengan baik setelah mendapatkan hasil test yaitu : *Error Rate* = 0%, *Success Rate* = 100%, *Virtual user* = 50 dengan durasi test selama 10 menit. Sehingga untuk sistem atau *API* dengan *load* tinggi serta data yang dinamis maka metode *Transforming* dan *Rate Limiting* bisa pakai.

DAFTAR PUSTAKA

- [1] C. A. Cholikh, "PERKEMBANGAN TEKNOLOGI INFORMASI KOMUNIKASI / ICT DALAM BERBAGAI BIDANG," *J. Fak. Tek.*, vol. 2, no. 2, pp. 39–46, 2021.
- [2] I. A. D. Astuti, R. A. Sumarni, and D. L. Saraswati, "Pengembangan Media Pembelajaran Fisika Mobile Learning berbasis Android," *J. Penelit. Pengemb. Pendidik. Fis.*, vol. 3, no. 1, p. 57, 2017, doi: 10.21009/1.03108.
- [3] D. Rahmawati, "Analisis Faktor Faktor yang Berpengaruh Terhadap Pemanfaatan Teknologi Informasi," *J. Ekon. dan Pendidik.*, vol. 5, no. 1, pp. 107–118, 2012, doi: 10.21831/jep.v5i1.606.
- [4] J. Kuswanto and F. Radiansah, "Media Pembelajaran Berbasis Android Pada Mata Pelajaran Sistem Operasi Jaringan Kelas XI," *J. Media Infotama*, vol. 14, no. 1, 2018, doi: 10.37676/jmi.v14i1.467.
- [5] I. A. Kaniya Pradnya Paramitha, D. M. Wiharta, and I. M. Arsa Suyadnya, "Perancangan Dan Implementasi Restful Api Pada Sistem Informasi Manajemen Dosen Universitas Udayana," *J. SPEKTRUM*, vol. 9, no. 3, p. 15, 2022, doi: 10.24843/spektrum.2022.v09.i03.p3.

- [6] A. Tedyyana, M. Fauzi, and F. Ratnawati, “Revamp Keamanan Web Service Milik PT XYZ Menggunakan REST API,” *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 12, no. 1, pp. 1–10, 2021, doi: 10.31849/digitalzone.v12i1.6378.
- [7] W. Sulistiani and W. Sulisty, “Implementasi Web Service dengan Metode REST Berbasis Golang pada Layanan Google Cloud Platform di PT Sumber Alfaria Trijaya , Tbk . Artikel Ilmiah Implementasi Web Service dengan Metode REST Berbasis Golang pada Layanan Google Cloud Platform di PT Sumber ,” no. 672015229, 2020.
- [8] T. R. Al Fath and I. H. Al Amin, “Implementasi Arsitektur Microservices menggunakan RESTful API untuk Website Online Course Esploor,” *J. Tek. Inform. Unika ST. Thomas*, vol. 07, pp. 2657–1501, 2022.
- [9] Y. Svetashova, S. Schmid, and Y. Sure-Vetter, “New facets of semantic interoperability: Adding JSON - JSON-LD transformation functionality to the BIG IoT API,” *CEUR Workshop Proc.*, vol. 1963, pp. 1–4, 2017.
- [10] D. J. Riyanto, P. Pizaini, N. S. H., and M. Affandes, “Implementasi Service Choreography Pattern Arsitektur Microservice Classroom Akademik Menggunakan Docker,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 3, pp. 768–779, 2022, doi: 10.29100/jipi.v7i3.3126.