

A System Integrator Model and Implementation Using Synchronous Communications Approach for PIS-PK System of the Ministry of Health, Republic of Indonesia

Faiq Zigo Fananie¹, Astie Darmayantie²

^{1,2}Department of Management Information System, Gunadarma University, Depok, West Java, Indonesia-16424

Abstract— *The PISPK (Program Indonesia Sehat dengan Pendekatan Keluarga) application is an electronic system that already exists in the ministry of health. PIS-PK is one part of the 5th Article in Nawa Cita Agenda, which is to improve the Quality of Life for Indonesian. In 2021, system development will be carried out by making additional modules based on the needs required by policy makers in the ministry of health, such as health services, health offices, and beneficiaries of the PISPK application. One of the challenges of the PISPK application is the large amounts of users and data. PISPK needs a system that can integrate all the results of family interviews in every province in Indonesia. The system shows a system integrator models can run properly with 0 errors and 100% passing the test.*

Keywords— *Backend, Endpoint, Database, Integrated System, REST API, Synchronous Process.*

I. INTRODUCTION

Health improvement has become a significant social priority since a moderately good human capital condition improves the abilities, efficiency, and quality of life of a workforce. Moreover, through its impact on the output in production and service, human capital accumulation is a primary determinant of economic development. Accordingly, health is an essential bridge to link human capital accumulation and economic growth [1]. The health of a country has a variety of effects on its economic development. When public health status improves, value is added to manufacturing and service by merging expert knowledge, capital equipment, technical skills, and medical science [2]. The quantity of investment, the social policy environment, and the quality of services are all factors that influence health outcomes. People who are wealthier will seek the best health services, and they will be willing to pay a greater price for higher-quality private healthcare services. Even if expenditure levels are low, health status would be a better outcome for a country where the government executes its social and health policies well [3]. The government's health expenditure is one of the essential factors with which to accumulate human capital as health status has been considered one of the unique elements that affect economic growth for a country [4].

According to the geographic position, health and economic development are vital in Asia countries. From the perspective of economic, Asian countries contributed ~60% of the world GDP per capita in 2017. On the other hand, investigating healthcare expenditure is important for the government to

improve the health system and health plan of a country, as a rapid increase of healthcare expenditure is a trend in major developing and developed countries [5][6]. According to the global health expenditure database in World Health Organization, healthcare expenditure as a percentage of GDP varied from 1.5 to 11.5 among Asian countries in 2017.

In Indonesia, its very wide area and separated from the sea is a factor that the region in this country does not get health services evenly, in addition to difficult access, and the lack of awareness of every citizen about health. This makes the development of digital transformation in health very important. During the Covid-19 pandemic, digital transformation in the health sector grew rapidly through digital health applications (HA). This momentum had indeed accelerated these transformations. Under the physical and social distancing policy, there was a migration of social activities from physical to virtual through many digital platforms. That made people's lives during the pandemic more comfortable even when their smartphone or laptop had no direct interaction. Because conventional healthcare Centers (HC) are the most vulnerable area of transmission of the Covid-19 between people. Therefore, virtual services become a very rational choice for people who need HC services without having to come to a HC like a clinic or hospital. It means that the innovation in medical sector, especially using the smartphone platform, are changing people's lives including the way they seek medical information or services [7].

The Ministry of Communication and Information Technology noted that the use of HA had increased rapidly by 443 percent since the Covid-19 pandemic. The HA users from various applications in Indonesia approached 4 million in the beginning, and during the pandemic users of various HA had exceeded 15 million people [7]. innovations that have been made such as peduli lindungi and PIS-PK.

To have a clear picture of how health conditions are in Indonesia, the Ministry of Health has developed a measuring instrument to calculate the national health index. This calculation is known as "Indeks Keluarga Sehat", which includes 12 indicators. The IKS calculation is carried out from the Puskesmas level through survey officers and is calculated so that it is at the national level. With this instrument, a more comprehensive portrait of public health can be obtained. To perform the IKS calculation, information technology is widely used, both from entering data, processing data, presenting data

and including performing data visualization. The output of IKS is used by the Ministry of Health to consider programs and policies in the field of public health. Since this data can also be obtained at the provincial and district/city levels, it can be used to design public health policies. The PISPK (Program Indonesia Sehat dengan Pendekatan Keluarga) application is an electronic system that already exists in the ministry of health. In 2021, system development will be carried out by making additional modules based on the needs required by policy makers in the ministry of health, such as health services, health offices, and beneficiaries of the PISPK application. One of the challenges of the PISPK application is the large amounts of users and data. To support PIS-PK which is one of the 5th Agenda of Nawa Cita, Improving the Quality of Life for Indonesian. PISPK is in the 2015-2019 National Medium-Term Development Plan, through the Decree of the Minister of Health, Republic of Indonesia. Number HK.02.02/Menkes/52/2015 about the Strategic Plan of the Ministry of Health for 2015-2019. The target of the PISPK is to increase the health and nutritional status of the community. through health efforts and community empowerment supported by fair distribution of health services.

PISPK needs a system that can integrate all the results of family interviews in every province in Indonesia. its main purpose is to collect, categorize and analyze information based on data that is integrated across the region into one very large system. The information certainly has many benefits for the advancement of family health in Indonesia. Therefore, PISPK is required to have a system that is suitable and safe from the needs described above.

II. STATE OF THE ART

The concept of microservice architecture was first introduced by Lewis and Fowler back in 2014. The key idea behind microservices architecture is the system built as a collection of services that are distributed into many small parts with their respective functions, and communicate remotely with each other. By doing this, the goal of the microservices architecture is to improve the maintainability as well as the scalability of the developed applications. Each microservice is supposed to be an independent service that is responsible for a very specific task. Microservices can communicate through different methods and the functionality of the system is based on the cooperation of multiple microservices communicating with each other. Independency and loose coupling of the different services is the most crucial aspect and desired feature of the microservice architecture. This helps reduce the many dependencies between different parts of the system and is the basis for most of the benefits of this architectural style.

Microservices architecture uses the well-known single responsibility principle regarding different sub-systems of microservices. This means that each microservice must have one well-defined purpose. Changes made in microservices should be enforceable independently and do not require cascading changes in other microservices.

Different microservices expose potential to the outside through lightweight communication mechanisms like REST APIs. Each microservice includes logic to perform for incoming

or outgoing requests. therefore, the communication path that messages pass between microservices should not contain a lot of logic.

Research by Vijay Surwase, describes the connectivity and advantages in implementing service oriented architecture together with Web Service-based REST API as a way of communication. This research has an important contribution in selecting the REST API that goes hand in hand with SOA on the integration system to be built. REST API is considered suitable when run in conjunction with SOA when viewed from all its advantages. In this study, it is explained that the REST architecture provides all the advantages in using API as a communication line between services that exist in the system.

The above principles result in beneficial properties such as simplicity, scalability, modifiability (agility in the underlying system), reliability (Through failover mechanisms or ensuring high availability), visibility (Use of open interfaces), performance (Dynamic load balancing), and portability. Like SOA, REST is an architectural discipline defined by a set of design principles, and a set of architectural constraints are also imposed by REST. The resource-centric model used in REST is the opposite of the object-centric model. In REST, everything is related to a resource. When modeling a RESTful service, the service is encapsulated and exposed as a resource pool. ROA (Restful Oriented Architecture) can be considered as a strategy to align IT assets with business capabilities in a RESTful manner, the careful design of strategic components by making the best use of SOA and REST proves to be a very good investment. SOA's strong focus on sharing and reuse can optimize the utilization of IT assets and REST's focus on extensibility and secure means of communicating to external systems, together make for a great tool to implement into systems.

III. RESEARCH METHODOLOGY

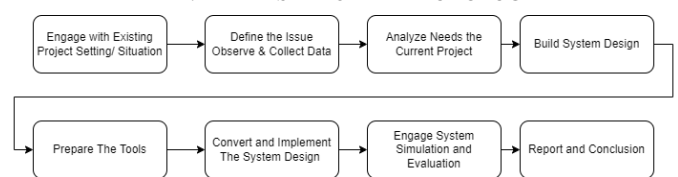


Fig. 1. Research Methodology

This section will be discussed about the steps and procedures of research activities carried out, from engage with existing project to report and discussion. This section starts from common research phases then, followed by explanations in each phase below as follows:

1. First step is engaging with the existing project setting and situation.
2. Then, defining the issue, observe and collect data.
3. The next step is analyzing needs the current project.
4. The fourth step is preparing the tools.
5. The fifth step is building system design
6. The sixth step is converting and implementing the system design.
7. Afterwards, engage system simulation and evaluation.
8. The final step is report and discussion.

A. Engaging with The Existing Project Setting and Situation.

The subject of this study is Ministry of Health Republic of Indonesia, PISPK System Integration. PISPK is a very large system which is divided into several sub-systems. Data integration is one of the PISPK sub-systems which has an important role in integrating all family data from each province and collected in one database. The goal is to collect all IKS data that has been collected in each province and become the initiation of data in the database.

B. Defining The Issue, Observe and Collect Data

The problem that is currently being faced in building the PISPK data integration system is that it allows this system with two-way integration. An integration system that allows the sender to receive a response such as a description of whether the data is accepted or rejected. hence, the most suitable system is needed to solve this problem. Data collection is done by conduction the literature reviews to find and study the references for the research carried out by related documents and paper through the internet.

C. Analyzing Needs the Current Project

Import database is one of the possible solutions. only with database files, very large data will be easily sent. But it does not answer the problem of two-way integration. In this way, the sender can only send data without knowing whether the data is accepted or rejected. Likewise with import excel, data may be sent according to the sender, but he does not know whether the data was actually sent.

The most likely way to solve this problem is to use an API as a data input gateway with logging mechanism, login mechanism, and API documentation. The sender can make a request and then receive a response from the recipient of the data.

D. System Design

To meet the needs of the required system, it is necessary to describe the system that will be made first. This System design will explain how the integration of this system will work. This section will explain the system as a whole, such as data flow, data structure, data verification rules and the endpoint design that will be used.

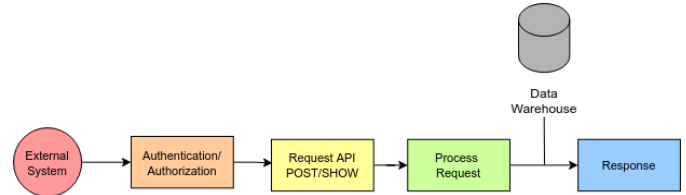


Fig. 2. Integrated System Design

The Integration System in Figure 2 is a service that is used to integrate the health system from the provincial level to submit data into the Pispk system. This system is useful in enabling collaboration in data collection from each region without compromising regional creativity.

PISPK data integration system uses JSON RAW format to receive data from external systems (Access from provincial level). the external system must have been validated and allowed to send requests to the PISPK system. at the authentication/authorization stage, each of these external systems is validated. the goal is that only a few institutions that are approved to be granted access can make requests to prevent unidentified parties from accessing them. After the access is validated, the external system can make requests both supplying data or displaying data.

The desired request is processed first by verifying the request according to the standards set by the integration system. After the request is verified, the data can be saved or displayed. The data will be stored in a database that will integrate all data from each province. After the request is processed, then the response can be given to the external system, whether it is accepted or rejected because of it does not pass the data verification process.

E. Data Structure

In deciding which database to use, there is an important role in paying attention to the data structure first. this is done to determine which database matches the type of data structure to be created. The data structure this time describes the contents of each table that will be needed and their relationship with other tables.

There are 14 tables used: survei, survei_individu_detail, survei_rumah_tangga, provinsi, kota, kecamatan, kelurahan, ref_agama, ref_pekerjaan, ref_pendidikan, ref_puskesmas, tutup_buku, iks_inti, user.

TABLE I. Endpoint Design.

Method	Uri	Target Table	Parameter	Status OK	Status Failed
GET	/api/survei	survei,survei_rumah_tangga, survei_individu_detail	id	200	400
POST	/api/survei	survei,survei_rumah_tangga, survei_individu_detail	json raw	200	400
GET	/api/provinsi	provinsi	id	200	400
POST	/api/provinsi	provinsi	json raw	200	400
GET	/api/kota	kota	id	200	400
POST	/api/kota	kota	json raw	200	400
GET	/api/ref-kecamatan	ref_kecamatan	id	200	400
POST	/api/ref-kecamatan	ref_kecamatan	json raw	200	400
GET	/api/kelurahan	kelurahan	id	200	400
POST	/api/kelurahan	kelurahan	json raw	200	400
GET	/api/puskesmas	puskesmas	id	200	400
POST	/api/puskesmas	puskesmas	json raw	200	400
GET	/api/iks-inti	iks_inti	id	200	400
POST	/api/iks-inti	iks_inti	json raw	200	400

F. API Endpoint Design

To solve the problems above, API endpoint design is needed to provide a comprehensive picture start from the parameters required to the response desired by each requester.

The design is built using 14 API Endpoints. This endpoint serves to send data and view data according to the access that each account has. There are two methods used in each table, namely GET and PUT. The parameters used, the successful status used and the failed status used can be seen in

G. Prepare The Tools

This step will explain what technologies are used in the construction of the PISPK integration system. The selection of the wrong technology will affect performance and development of the system going forward. In this research, author uses Linux Operating System, Go Programming Language, MySQL, Swagger, and JWT Token. That also secondary tools to help author, like Visual Studio Code and Postman.

H. Implementing System Design

The choice of technology used in the system is one of the most important factors to achieve system development with high reliability and scalability. The selection of the wrong technology will affect performance and development of the system going forward. This step will be explained in detail in section IV about Results and Discussions.

I. Engage System Simulation and Evaluation

This step is a series of tests used to determine whether an application can run properly. This step will be explained in detail in section IV about Results and Discussions.

J. Report and Conclusion

This step contains a description of the conclusions obtained during the writing and suggestions given to interested parties to carry out further development. This step will be explained in detail in section V about Conclusions and Suggestions.

IV. RESULTS AND DISCUSSIONS

The stages are divided into 2 sub-stages. This stage focuses on technology used and required environment. The first stage are framework implementation, database implementation, endpoint implementation, authentication implementation, log implementation and swagger implementation for API documentation. The second stage focuses on simulation and evaluation.

A. Framework Implementation

According to the official website, The Go programming language is an open source project to make programmers more productive. Go is expressive, concise, clean, and efficient. Its concurrency capabilities make it simple to develop programmes that take use of multicore and networked machines, while its innovative type system makes programme construction flexible and modular. Go compiles swiftly to machine code while also providing the benefits of garbage collection and run-time introspection. It's a compiled, fast, statically typed language that has the feel of a dynamically typed, interpreted language.

Most importantly, based on the book “Cloud Native Programming with Golang”, Go is said to be perfect if it is developed for microservices, which we will see a lot in the future. Microservices Architecture divides its functionality into several small services that only focus on one specific task. The services can communicate with each other to get the information they need to produce.

The Go language is needed as a programming language for this system. This language needs to be installed first before it can be used further. Following are stages of the installation process:

1. Extract the archive downloaded into /usr/local, creating a Go tree in /usr/local/go. For example, run the following as root or through sudo:


```
rm -rf /usr/local/go && tar -C /usr/local -xzf go1.17.6.linux-amd64.tar.gz
```
2. Add /usr/local/go/bin to the PATH environment variable. Do this by adding the following line to \$HOME/.profile or /etc/profile (for a system-wide installation):


```
export PATH=$PATH:/usr/local/go/bin
```
3. Verify that you've installed Go by opening a command prompt and typing the following command:


```
$ go version
```

 - Confirm that the command prints the installed version of Go.
4. Download and extract echo-gorm
5. run the server


```
$ go run main.go
```

B. Database Implementation

In developing the PISPK Integration System, MySQL database is used. The developed database is placed in one database, where the database is grouped based on its function. Table relations that occur in the database are described using the Entity Relational Database (ERD) as shown in Figure 3

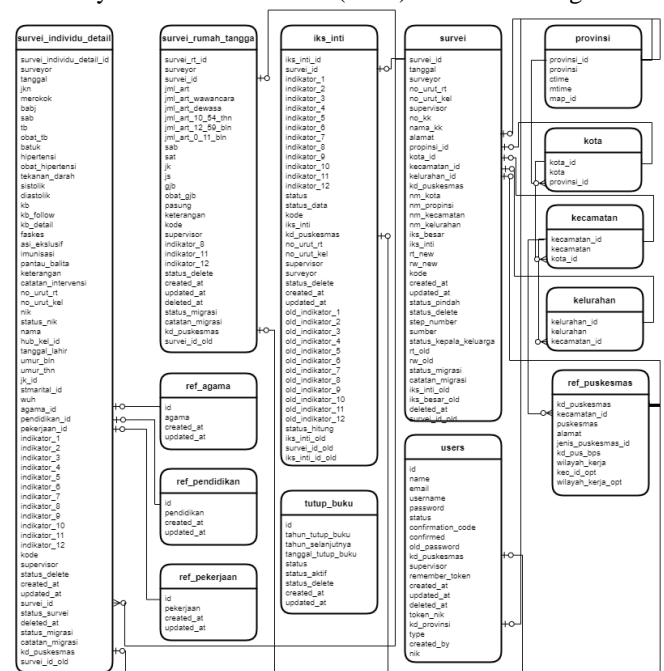


Fig. 3. ERD Database Design

C. Endpoint Implementation

In this case, the API being sampled is API GET provinsi. Here is the process:

1. Create a provinsi struct model. struct is created based on the table structure used in API GET provinsi.
2. Create an API GET provinsi controller. This controller contains all the logic and conditions that exist in an API.
3. Create an API GET provinsi route. This route serves as a set of addresses to go to if you want to make a request to API GET provinsi.

Once the API is created, there are two result conditions based on the request being sent. the result is success or error. Additional tools like Postman will help in testing the API

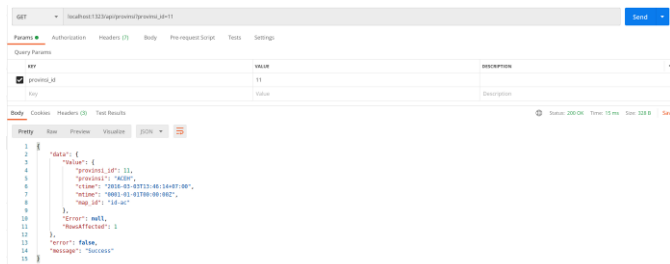


Fig. 4. Success Response



Fig. 5. Failed Response

D. Authentication Implementation

There are many ways to do the authentication. All methods have their own advantages and disadvantages. In PISPCK project, the most suitable login mechanism to use is the JWT Token. This is because the level of security is secure and difficult to hack, along with the use of this JWT (JSON Web Token) which is very suitable when used in conjunction with REST, where REST is the type of architecture used in API creation of this project.

The token method algorithm used in this system is HS256. JWT is divided into 3 parts, header, payload and signature. The basic information is encoded Base64Url and then stored in the payload section of JWT. Header section contains information on the algorithm used. Signature section contains the system-defined private key. The output is three Base64-URL strings separated by dots that can be easily passed in HTML and HTTP environments. This token will be a sign that the account is authenticated to send requests.

The installation process is carried out by implementing several functions and middleware for the checking process. This function will generate the JWT code that required by each account access. Here is the process:

1. The application of authentication to the system begins with integration by creating a function to generate JWT according to the data information provided. This function will be called on API login logic process.

2. For each API that has been created, it must go through the JWT validation process. this validation process is applied to a middleware that will be called right before entering the API endpoint.

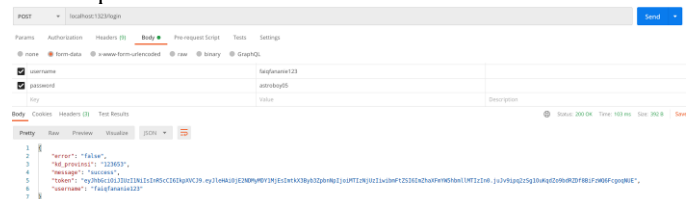


Fig. 6. Login with JWT

E. Log Implementation

Logging is the process of recording all API transactions that occur on a system. Logging has many benefits, including facilitating development, making it easier to find bugs, making history tracing easier, etc. Of course, the role of logging in developing technology has become mandatory. Logging in this system provides data in the form of recording requests that are accepted or rejected by the API. Logging here contains some information such as API request time, method used, HTTP status, level in the form of info or error, as well as messages filled in causing the request to be rejected.

In this system, logging mechanism uses logrus library in golang. every API transaction will be recorded whether it is successful or failed because it was rejected. everything is collected in one file named "info.log".

log level info:

```
{ "latency_ns": 1877580, "level": "info", "method": "GET", "msg": "", "path": "/api/survei", "status": 200, "time": "2022-01-24T21:44:33+07:00" }
```

log level error:

```
{ "latency_ns": 297, "level": "error", "method": "GET", "msg": "Record is not exist", "path": "/api/survei", "status": 400, "time": "2022-01-24T21:44:33+07:00" }
```

F. API Documentation

Api documentation is used to make it easier for clients to read the API structure. API Documentation Interface is designed to be easier to understand and use. This contains all the information needed to make a request to the endpoint that has been created, such as the method used, uri, required parameters, required authentication/authorization and the response provided. one API documentation contains all the APIs created by the developer. large selection of scattered API documentation.

In this project, the API documentation used is swagger. The advantage of swagger is open source, lightweight and there is a library that supports the use of swagger in language that used, golang. The swaggo/echo-swagger library is used in this project to support automatic API Documentation creation with swagger.



Fig. 7. API Documentation with Swagger

G. Testing

In this case, the test case what is done is to test PISPK Integration System. This test case will describe the inputs entered, actions, expected results, desired results obtained, validity, and description. Test is done by testing conditions where the input given is correct and the input given is wrong. If the output given is in accordance with the expected value, it can be said the API have passed the test with the description "Pass".

The example test is done on the login API. The test is carried out with three conditions, where the first condition is with the correct input of information, the second is with input of information with the wrong username and last with input of information with the wrong password. As a result, the login API is categorized as passing the test with these three conditions.

TABLE II. API Test.

Step	Details	Input	Output	Expected Output	Test Result
1	Test Login with Correct Username and Password	username: faiqfananie123, password: astroboy05	" { \"error\": \"false\", \"kd_provinsi\": \"123653\", \"message\": \"success\", \"token\": \"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2NDMyODc5MDgsImtkX3Byb3ZpbmNpIjoiaWJlMTIzNjUzIiwibmFtZSI6ImZhaXFmYW5hbmlMTlZln0. gmWw6VUz5nn3IR71pxQX TxFsgUCDSqRD2efxmybOT2Q\", \"username\": \"faiqfananie123\" }"	Success	Pass
2	Test Login with Incorrect Username	username: faiqfananie1234, password: astroboy05	" { \"error\": true, \"message\": \"Record is not exist\", \"status\": 400 }"	Record is not exist	Pass
3	Test Login with Incorrect password	username: faiqfananie1234, password: 12345	" { \"error\": true, \"message\": \"Record is not exist\", \"status\": 400 }"	Record is not exist	Pass

The second test is using JMeter, the system will be accessed by 100 users done in 1 second.

TABLE III. API Load Test with JMeter.

Name	Success Rate	Latency ms	Throughput/sec	Received KB/sec	Sent KB/Sec
survei	100%	160	323.0	1128.59	44.47
provinsi	100%	49	1003.4	334.13	143.06
iks_inti	100%	57	807.2	545.47	112.72
kota	100%	57	928.0	250.13	126.88
kecamatan	100%	53	974	268.32	144.63
kelurahan	100%	51	1074.4	313.71	163.67
puskesmas	100%	46	1147.7	502.11	179.33

V. CONCLUSIONS AND SUGGESTIONS

This research successfully analyzed and implemented System Integration Using Synchronous Process. By implementing the REST architecture, problems like two-way communication that exist in the integration system can be solved. With the help of each API which has its own function and purpose, this system is able to integrate all existing data in each province into one main database.

The following are some suggestions that can be done for further research with synchronous process to build a system.

- Identify the problem first before deciding what technology to apply to the system
- Conduct continuous research on new technology, because its development is very rapid.
- Every decision method made always has its drawbacks, therefore it is very important to choose the appropriate technology not the most sophisticated one.

REFERENCES

- [1] Eggoh, Jude, Hilaire Houeninvo, and Gilles-Armand Sossou. "Education, health and economic growth in African countries." *Journal of Economic Development*, vol. 40, issue 1, pp. 93, 2015.
- [2] Morand, Olivier F. "Economic growth, health, and longevity in the very long term: facts and mechanisms." *Health and economic growth: findings and policy implications*, vol. 74, issue .2, 2005.
- [3] Balaji, B. "Causal nexus between public health expenditure and economic growth in four southern Indian states." *IUP Journal of Public Finance*, vol. 9, issue 3, pp.7, 2011.
- [4] Bedir, Serap. "Healthcare expenditure and economic growth in developing countries." *Advances in Economics and Business*, vol. 4, issue 2, pp.76-78, 2016.
- [5] Dieleman, Joseph L., et al. "Trends in future health financing and coverage: future health spending and universal health coverage in 188 countries, 2016–40." *The Lancet*, vol. 391, issue 10132, pp.1783-1798, 2018.
- [6] Yu, Tiffany Hui-Kuang, David Han-Min Wang, and Li-Ying Chang. "Examining the heterogeneous effect of healthcare expenditure determinants." *International Journal of Behavioural and Healthcare Research*, vol. 2, issue 3, pp.205-213, 2011.
- [7] Pitaloka, Ardanawati Ayu, and Arief Priyo Nugroho. "Digital Transformation in Indonesian Healthcare." *STI Policy and Management Journal*, vol. 6, issue 1, 2021.
- [8] M. Fowler and J. Lewis. "Microservices a definition of this new architectural term", 2014.
- [9] Surwase, Vijay. "REST API modeling languages-a developer's perspective." *Int. J. Sci. Technol. Eng.*, vol. 2, issue 10, pp.634-637, 2016.