IMPLEMENTATION OF MICROSERVICES ARCHITECTURE ON COMRADES BACKEND

Cahyanto Setya Budi[1], Adam Mukharil Bachtiar[2]

[1,2]Program Studi Teknik Informatika, Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116, Coblong, Bandung, Jawa Barat
E-mail : cahyantosetyabudi@gmail.com[1], adam@email.unikom.ac.id[2]

ABSTRACT

Comrades is a smart application as an information and education media about HIV / AIDS. The Comrades application helps people living with HIV / AIDS to exchange information and support and can connect with both the general public and fellow HIV / AIDS sufferers. In its development, Comrades also cooperated with related institutions, one of them was the community with HIV / AIDS in the surrounding area. Planned users from the Comrades application will increase due to the cooperation. To overcome this problem, it is necessary to increase the performance of the Comrades application. The purpose of this study is to improve performance and availability in the Comrades application. This can be achieved by applying the microservices architecture to Comrades web services from the previously monolithic ones. In this study the process of changing architecture in the Comrades application will use the Domain Driven Design approach to solving the service. While on the physical architecture side, it will also experience changes by applying several technologies currently available, including: Docker, Kubernetes and Gateway API.

Keywords : Microservices, Web Services, API, Domain Driven Design, Monolithic, Performance

1. INTRODUCTION

1.1 Background

Comrades is a smart application as an information and education media about HIV / AIDS. Comrades which is a smart application based on technology, helps people living with HIV / AIDS to exchange information and support and can connect with the general public and with fellow HIV / AIDS sufferers. There are several features in the Comrades application, such as: chat, information articles, news and events, positive tweets and stickers that can be sent as support for people with HIV / AIDS. In the process, Comrades also collaborates with the community with HIV / AIDS in the surrounding area. On the other hand, with the increasing number of communities working together in this application, it is planned that this will also increase the number of users over time. In addition, so that applications can continue to serve optimally there are several needs that must be met such as: performance, availability, and resilience when experiencing high requests from users [1]. Thus there is a type of architecture that is developing at this time that can meet these needs, namely, microservices.

In the early stages of the construction of the Comrades application using a type of monolithic architecture. This architecture applies the functional aspects of web services that use a uniform code base and technology in its application [2]. This happens because the application demands to be built quickly and with the limited number of available teams. As the business needs in application, monolithic architecture will become larger on the other hand monolithic architecture will also be difficult to develop [3]. Problems start to appear when more and more applications are functional, and traffic continues to increase. So that when a functional experiences an error it will have an impact on the overall web services. From the analysis of Blazemeter testing conducted on web services applications at Comrades, information is obtained that the application has not worked optimally because there are still deficiencies in the Comrades web services. This is evidenced by web services requiring at least 2.5 seconds in response.

Thus, based on the problems that occur as well as the literature studies that have been done before, a research is carried out in the form of changing the architecture of the Comrades application from monolithic to microservices. Microservices is an application architecture design that breaks applications into separate small services according to their functions [4]. This solution aims to create scalable, resilient and high-availability web services. The application of the microservices architecture is expected to handle the problems that occur in the Comrades application.

1.2 Identification of Problems

Based on the background described above, a problem can be formulated, namely how much influence the performance produced by applying the microservices architecture to Comrades web services.

1.3 Research Purpose

The purpose of this study is to improve the performance of the Comrades application.

1.4 Scope of Problem

The limitations of the problem in this study are as follows:

a. In this study the analysis focused on performance recorded in Quality of Services (QOS) Web Services.
b. The application of the microservices architecture in research only focuses on web services APIs on applications.
c. In the process of exchanging information between clients with the server using the JSON format with the REST mechanism, because this mechanism is lighter than the others [5].
d. In this study the process of breaking applications into microservices uses the Domain Driven Design approach.

2. RESEARCH CONTENT

2.1 Theory Based

2.1.1 Web Services

Web Service is a component of distributed systems based on Service Oriented Architecture (SOA) [6]. SOA is an application design architecture that uses components from pre-existing business needs. In this case the application built will be modular according to these components, this component is also called a service. Service built has an interface that functions to process and send messages in XML or JSON. Service created using the object oriented concept in the process of sending requests and responses.

2.1.2 Microservices

Microservices is an application architecture design that breaks applications into separate small services according to their functions. The service will be divided into more detail in terms of its functionality so that each function works independently [3]. In this case each service may be able to have a stack technology that is different from one another depending on service requirements. This means that there will be different technologies in one large application.

2.1.3 Domain Driven Design

Domain - Driven Design (DDD) is a software development approach in modeling an application, DDD helps in the software development process to be more accurate in accordance with the business processes that will run on the application [7]. One of the problems that often arises in the process of designing applications is communication between classes or functions that along with the development of applications will be even greater [8]. This problem will cause a dead code, when a large application code is not dare to be touched by the developer for fear of causing errors. The longer the dead code will be bigger and even increase, this becomes something frightening for developers.

2.1.4 Performance for Web Service

In building a good web service, a standard is needed so that the web service that is built can work well in accordance with its purpose [9]. This standard must cover many aspects so that web service is not only in accordance with the objectives, but in terms of performance, cost, maintainability, etc. can be said to be optimal. The standards that must be met include [1]:

a. Response Time

Response time is the total time used by the service in responding to incoming requests.

b. Throughput

Throughput is the number of requests from the same service that can be served in a certain time unit.

c. Latency

Latency is the length of time it takes for a service to send a response from the server to the client.

2.2 Methods

The research methodology used in this study is a quantitative approach that uses a comparative type of research. This research aims to compare between two objects with certain variables. In this case the object that is compared is the level of web service performance in the Comrades application that uses the previous architecture and the architecture that will be applied, namely microservices. The following are the stages in this study which can be seen in Figure 1:

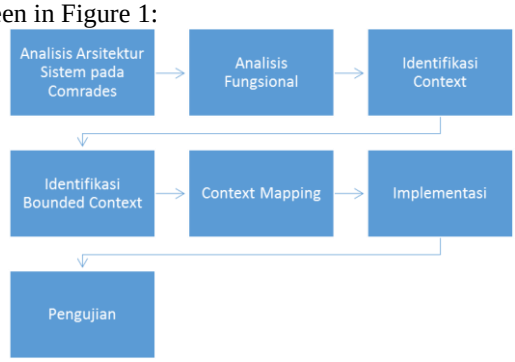


Figure 1 Research Methodology

Analysis of System Architecture at Comrades

At this stage an analysis of the architecture of the entire application is being carried out.

Functional Analysis

At this stage, the functional analysis of the system and its description will be carried out. This function explains what business processes are contained in the application.

Identification of Context

At this stage there are groupings of several functionalities that have been explained in the same context.

Identify Bounded Context
At this stage it is done to find out the boundaries that each context has.

Context Mapping
At this stage is done to describe the overall context and the type of relationship between each other.

Implementation
At this stage, web services will be implemented into the code, in accordance with the design of microservices that have been done before.

Testing
At this stage web service testing has been made. This test aims to find out whether the web services that have been made are appropriate. This test is also useful to find out whether the new architecture of web services has improved performance.

2.3 Result and Discussion

2.3.1 Analysis of System Architecture at Comrades

At this stage an analysis of the architecture of the entire application is being carried out. The architecture that is running on the Comrades application can be seen in Figure 2:

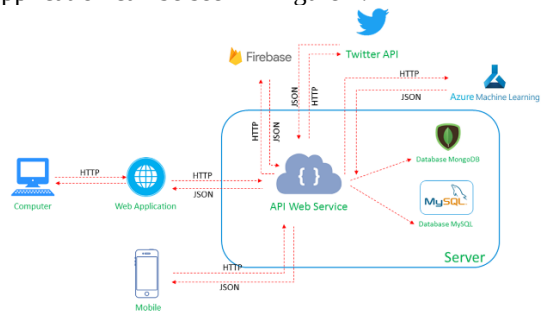


Figure 2 Comrades System Architecture

After knowing the overall architecture that is running, then in the analysis of any problems contained in the architecture.

2.3.2 Problem Analysis

At this time several web services are still long in response, this is evidenced through testing using Blazemeter tools. The results of the tests that have been carried out can be seen in Table 1:

Table 1 Results of Old Architecture Testing

Service	Avg. Respon (ms)	Minimum Response (ms)	Maximum Response (ms)
Admap All Posting	4932.926	4043	10036
Admapp All Artikel	3273.912	2483	4190
List All Artikel	3131.818	2135	5268
List All Berita	2721.784	2127	4005
Admapp Berita by Admin	1856.038	1455	3189
Ambil Tweet	1713.467	1603	2314
List 5 Artikel	1731.357	1032	4197

After finding out the problems found in the Comrades architecture, the architecture changes from monolithic to microservices. This change is expected to later improve the performance of the Comrades web service.

2.3.3 Analysis of the New Comrades System Architecture

This stage aims to explain the system architecture that will be applied to Comrades web services. The overall picture of the architecture to be applied can be seen in Figure 3:

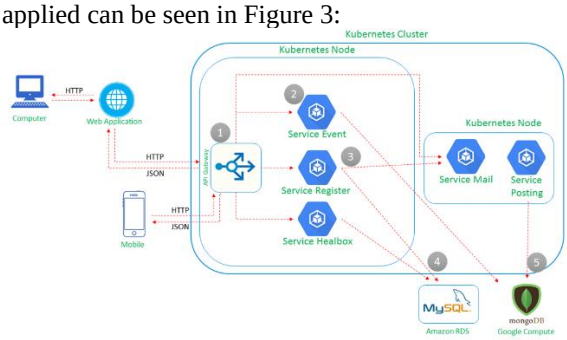


Figure 3 Microservices' overall architecture

The following is a description of the new architecture that will be applied to Comrades web services:

a. On the API side the Gateway will use Ambassadors, useful ambassadors handle incoming requests and proceed to the destination services contained in the internet service. The Ambassador is also equipped with a load balancer that will divide the request if it is considered to exceed capacity.
b. On the service side, each service represents the service that is on the web service. In the service it may contain several pods, it depends on the service needs in serving requests from users.
c. On the service side, each service makes it possible to communicate with each other if it is needed. This means that it happens when a microservices communicate with each other with microservices. Communication can not only be done inside the node but can also be done outside the node. The mechanism used is using http and JSON formats. For this reason, Service

Discovery is needed, service discovery is a mechanism that is performed to find available services and their addresses [10].

d. In the MySQL database, all existing services will go directly to the database used.
e. In the MongoDB database, all existing services will go directly to MongoDB that is used.

2.3.4 Functional Analysis

At this stage, the functional analysis of the system and its description will be carried out. This functional explains what business processes are contained in the application [11]. In functional analysis will use the use case diagram modeling. From the results of this analysis obtained 23 functionalities contained in the Comrades application. The following are some of the functionalities found in the Comrades application which can be seen in Table 2:

Table 2 Functional in the Comrades Application

Use Case	Description
Register	This functionality is used by general users to register themselves into the system
View Support Tweets	This functionality is used to find out tweets of support for PLWHA
See the location and service of the drug	This functionality is used to find out the place of service and medicine for the closest PLWHA
Recommend a place	This functionality is used to recommend places that PLHIV can use to get servants about HIV / AIDS

Once the functional in the Comrades application is known, then the same functional-functional will be grouped into context.

2.3.5 Identification of Context

At this stage grouping aims to find out which functionals are similar. The context contained in Comrades web services can be seen in Table 3:

Table 3 Context list in the Comrades Application

No.	Nama Context
1.	Event
2.	Friends
3.	Healbox
4.	Lokasi Obat
5.	Notifikasi
6.	Posting
7.	SMS
8.	Twitter
9.	User
10.	Email
11.	AES
12.	Sticker

From the results of the identification of the context that has been done, 12 contexts are found in the Comrades application.

2.3.6 Identify Bounded Context

The characteristics possessed by the microservices architecture are found in the autonomy of each service, for this reason an analysis is bounded context. This analysis functions to find out what limitations are contained in the context. The list of bounded context contained in the Comrades context event can be seen in Table 4:

Table 4 Bounded Context in the Comrades Application

Bounded Context	Description
Event	Manage event data on web services
Notifikasi	Send new event notifications on the mobile application

From the total identification results, there are 21 bounded contexts scattered in the Comrades application.

2.3.7 Context Mapping

At this stage all identified bounded contexts will be described in their entirety. The context map pieces contained in the Comrades application can be seen in Figure 4:

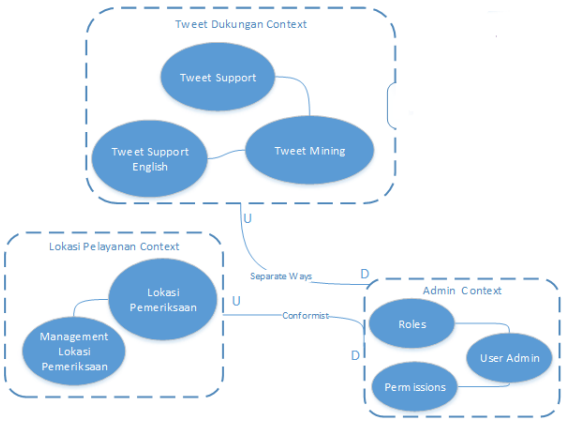


Figure 4 Context Map pada Aplikasi Comrades

Based on the results of the context map that has been described, then microservices will be implemented in the Comrades application.

2.3.8 Implementation

At this stage the application of microservices will be carried out on the Comrades application. Architecture is based on the context identified earlier

in the Context Identification section. As for some of the results of the implementation of microservices that have been done can be seen in Figure 5:

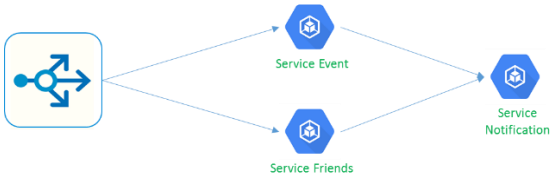


Figure 5 Microservices implementation on Web Services Comrades

2.3.9 Testing

System testing is useful to see and find deficiencies that are owned by the new system that has been implemented, so that it can help system development in the future. In addition, system testing is also carried out to find out if the new system is better than before.

3. TERMINATION

3.1 Conclusion

Based on the test results, the following conclusions are obtained:

a. In testing the integration of the new system architecture it can be concluded that the functional-function web services are already running in accordance with their respective tasks.
b. In testing the performance of web services with the same number of samples, web services architecture using microservices proved to be superior to monolithic. The results of the tests that have been carried out can be seen in Table 5 and Table 6.

Table 5 Test Results on Monolithic Architecture

Functional	Avg. Response Time (ms)	Avg. Latency (ms)	Avg, Throughput (second)
Admapp Artikel	3273	2419	0.051
Admapp Berita	2720	1995	0.05
Admapp Berita by Admin	1856	1425	0.05
Artikel	1713	1506	0.052
Berita	1504	1296	0.051
Detail Lokasi Obat	1299	1091	0.05

Table 6 Test Results on Microservices Architecture

Functional	Avg. Response Time (ms)	Avg. Latency (ms)	Avg, Throughput (second)
Admapp Artikel	654.78	483.99	0.01
Admapp Berita	544.00	399.14	0.01
Admapp Berita by Admin	371.21	285.03	0.01
Artikel	300.85	259.27	0.01
Berita	259.87	218.36	0.01
Detail Lokasi Obat	87.53	87.51	0.01

3.2 Suggestion

In research the level of web services performance was measured based on Quality of Web Services (QOS) as a reference. But from all the points contained in QOS, only a few points were applied in this study. Thus suggestions that use the same research topic are as follows:

a. On QOS as a reference, points that will be used in comparison are added.
b. The application of microservices architecture not only focuses on web services, but also from the other side.

REFERENCES

[1] T. Rajendran dan P. Balasubramanie, "Analysis on the Study of QoS-Aware Web Services Discovery," J. Comput., vol. 1, no. 1, hal. 2151–9617, 2009.
[2] A. Messina, R. Rizzo, P. Storniolo, A. Urso, dan I. Cnr, "A Simplified Database Pattern for the Microservice Architecture," no. c, hal. 35–40, 2016.
[3] K. Katuwal, "Microservices : A Flexible Architecture for the Digital Age Version 1 . 1," vol. 3, no. 4, hal. 23–28, 2016.
[4] B. Familiar, Microservice Architecture. 2015.
[5] A. Balalaie, A. Heydarnoori, dan P. Jamshidi, "Microservices Migration Patterns," no. 1, hal. 1–21, 2015.
[6] W3C.org, "Web Services Architecture." [Daring]. Tersedia pada: https://www.w3.org/TR/ws-arch/#whatis. [Diakses: 25-Agu-2018].
[7] V. Vernon, Domain-Driven Design Distilled. 2015.
[8] T. N. Millett Scott, Patterns, Principles and Practices of Domain-Driven Design. 2015.
[9] M. S. Sharmila dan E. Ramadevi, "Analysis of Performance Testing on Web

Applications," Int. J. Adv. Res. Comput. Commun. Eng., vol. 3, no. 3, hal. 5258–5260, 2014.
[10] F. Montesi dan J. Weber, "Circuit Breakers, Discovery, and API Gateways in Microservices," no. September, 2016.
[11] A. Suhendra dan A. M. Bachtiar, "Migration Code Pada Backend Crimezone Dari Php Ke Scala," vol. 12, no. 2, hal. 74–81, 2016.