ABSTRACT

The Telkom University Open Library Application is a website that was developed as a means of open information for every book, scientific work, and journal with several features such as search, catalog, and book reservation. Based on the data collection that the researcher conducted in the form of non-functional software testing in the form of load and performance testing, the Open Library application at the Telkom University Library requires improvements in terms of scalability, availability, and performance because the system cannot serve requests at a certain amount and time. The inability of a system to handle many requests goes hand in hand with the agency's loss of not being able to serve customers at any given time. The backend system currently built in the Telkom University Open Library Application uses a monolithic architecture where the entire system is built, designed, and maintained in one unit. The disadvantage of using a monolithic architecture is that it is difficult for software to scale both in terms of performance and development productivity. These problems will affect developer productivity in development, speed, and a system. By building an architecture with Microservices and using gRPC as Remote Procedure Invocation / Call the data exchange process on Microservices services is much faster and the architecture is developed over time. This study aims to implement a software architecture with a Microservices architectural style using gRPC as an exchange of data or information from services owned by the Telkom University Open Library application to improve or implement the Scalability, Availability, & Performance Pattern of a designed system. The development method used in this research is the Software Development Life Cycle (SDLC) with the Iterative SDLC model as the stages for developing the backend of the Telkom University Open Library Application. The technique used to evaluate this research is to test non-functional software to measure the limit of users who can access the same time on a system and the speed of the system in the form of response time. Based on the problem, the author migrates software with software starting by collecting each function and process from the Telkom University Open Library Application Library and separating each business process into an independent domain by applying the concept of domain driven design. After decomposing the business process, the *author migrates the monolithic software architecture to microservices by applying a strangulation pattern and development to the functional testing stage in the form of unit testing on each module and non-functional testing in the form of performance tests, load tests, stress tests, and surge tests with specified scenario. The results show that there are several problems of monolithic software architecture that can be categorized into scaling in the context of performance and productivity and development. As a result of migrating software architecture to microservices, organizations can improve system performance to choose the user's particular preferences and freedom in technology according to the problem at hand.*

Keywords— ***microservices, open library, gRPc***