CHAPTER 6
CONCLUSION

The test results obtained by testing both monolith and microservice with several scenarios are quite varied. From the testing, it could be concluded that the architecture affects the performance and it shows that:

1. In Merchant service, the average latency gap between microservice and monolith is 662,34 ms, where monolith average latency is only half of microservice's. In terms of success rate, microservice slightly edge monolith with 8,45% or 515 more successful request on microservice side. By looking at the number gap. It is clear that monolith is better for merchant service
2. In User service, monolith is better with edging microservice on both average latency and success rate with 28,5 ms lower latency and 4,75% more success rate.
3. In Transaction service, unlike user service, microservice performs better with having 494 ms lower latency and better success rate at 2,96%.

By looking at the overall testing data, the average latency from the monolith is better than the microservice by a 71.61 ms gap. Meanwhile, in terms of success rate, the data shows that microservices average success rate edge monolith with a small gap of 2.22 percentage. The latency average number result for microservice is slightly worse because there are more success requests and taking more time. The usage of the API gateway also affects the latency because the request has to go through NGINX before arriving on the server.

By changing the testing design, it could be concluded that the testing design is affecting the service's performance and it shows that:

1. Merchant service have the best latency on design 1 but the best success rate average on design 3 with 15% gap with the first one. But the latency on the third testing design is 3 times higher than first testing design
2. User service performs best on design 3 since it has lowest latency, eventhough the average success rate is worse but the gap only 0,15%.
3. Transaction service performs best on design 1 like merchant service since it has the lowest latency average and highest success rate.

From the overall data it shows that separating JMETER from another device makes the latency worse by 804.06 ms. But separating NGINX to the third device also did not make the latency better. It is shown that it has 1537,33 ms higher average latency than the first testing design. The average latency become worse because the request took longer times since the server is not in the local environment anymore. Meanwhile, in terms of success rate, from the testing data, the success rate on design 2 is slightly worse than design 1 with a 9.48% gap. On the other hand, testing design 3 has a slightly better success rate average with a 4.48% gap from design 1.

From the data it appears that there is no absolute better architecture. It could be concluded that every architecture has its own advantages. But from overall if the service is more loaded, need high chance of success rate, microservice is a better choice. If the service is light weight and need faster response, monolith is the option. In further research, the future researcher could use a better device for the server and testing. Also, the microservice could be made more complex by adding workers and increasing the number of features. It is also possible to use various database machines outside PostgreSQL with using various RDBMS for microservice.