



PROJECT REPORT
MICROSERVICE AND MONOLITH PERFORMANCE
COMPARISON IN TRANSACTION APPLICATION

ALEXANDER JASON LAUWREN
18.K1.0054

Faculty of Computer Science
Soegijapranata Catholic University
2022

HALAMAN PENGESAHAN



Judul Tugas Akhir: : Microservice and Monolith Performance Comparison in Transaction Application

Diajukan oleh : Alexander Jason Lauwren

NIM : 18.K1.0054

Tanggal disetujui : 06 Januari 2022

Telah setuju oleh

Pembimbing : Y.b. Dwi Setianto S.T., M.Cs.

Penguji 1 : Y.b. Dwi Setianto S.T., M.Cs.

Penguji 2 : Hironimus Leong S.Kom., M.Kom.

Penguji 3 : R. Setiawan Aji Nugroho S.T., MCompIT., Ph.D

Penguji 4 : Yonathan Purbo Santosa S.Kom., M.Sc

Penguji 5 : Yulianto Tejo Putranto S.T., M.T.

Penguji 6 : Rosita Herawati S.T., M.I.T.

Ketua Program Studi : Rosita Herawati S.T., M.I.T.

Dekan : Dr. Bernardinus Harnadi S.T., M.T.

Halaman ini merupakan halaman yang sah dan dapat diverifikasi melalui alamat di bawah ini.

sintak.unika.ac.id/skripsi/verifikasi/?id=18.K1.0054

DECLARATION OF AUTHORSHIP

I, the undersigned:

Name : Alexander Jason Lauwren

ID : 18.K1.0054

declare that this work, titled Microservice and Monolith Performance Comparison in Transaction Application, and the work presented in it is my own. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at Soegijapranata Catholic University
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
3. Where I have consulted the published work of others, this is always clearly attributed.
4. Where I have quoted from the work of others, the source is always given.
5. Except for such quotations, this work is entirely my own work.
6. I have acknowledged all main sources of help.
7. Where the work is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Semarang, Januari, 15, 2022



Alexander Jason Lauwren

18.K1.0054

HALAMAN PERNYATAAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan dibawah ini:

Nama : Alexander Jason Lauwren

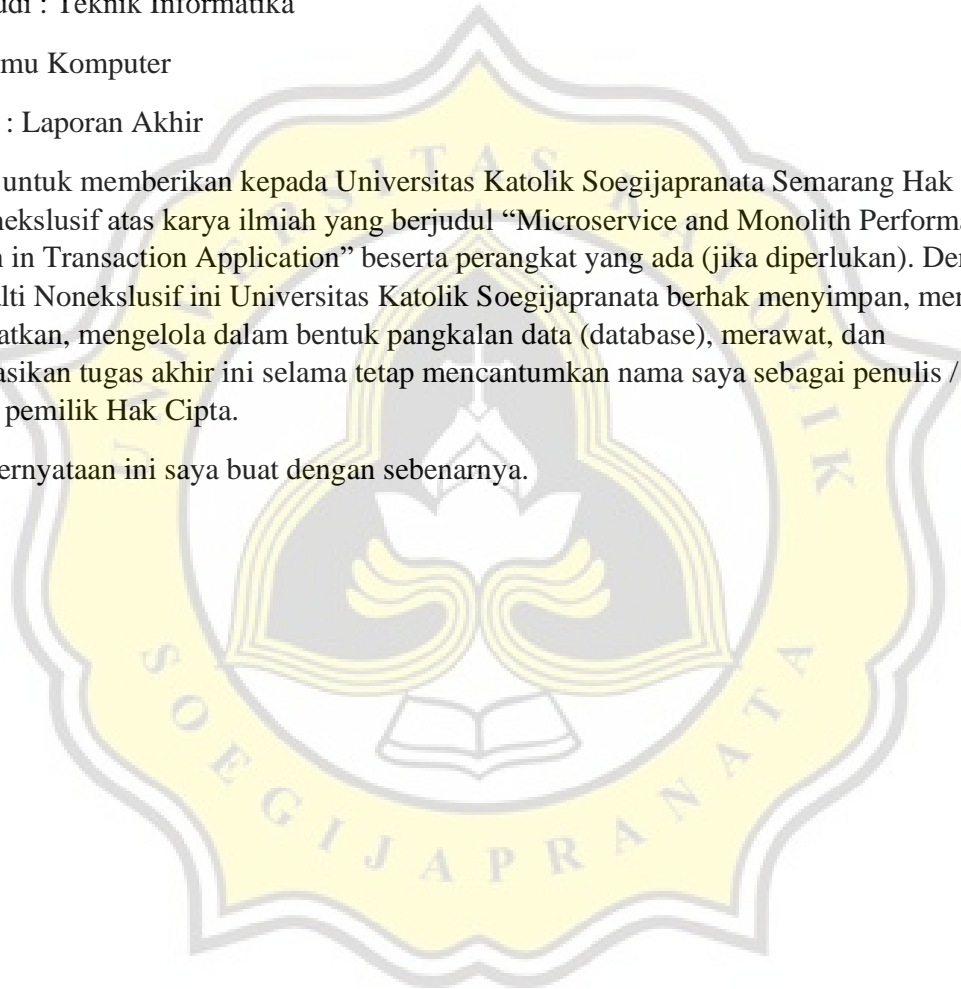
Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Jenis Karya : Laporan Akhir

Menyetujui untuk memberikan kepada Universitas Katolik Soegijapranata Semarang Hak Bebas Royalti Noneksklusif atas karya ilmiah yang berjudul “Microservice and Monolith Performance Comparison in Transaction Application” beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Katolik Soegijapranata berhak menyimpan, mengalihkan media/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir ini selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.



Semarang, 15 Januari 2022

Alexander Jason Lauwren

ACKNOWLEDGMENT

First of all, I'd like to thank Jesus Christ because if not for His blessings, this final project is impossible to be this successful. The final project is a mandatory requirement to be a Bachelor of Computer Science in the Informatic Engineering Study Program at Soegijapranata Catholic University Semarang.

Along the journey and preparation of making this final project, I am blessed with the support and encouragement of people around me. Especially I'd like to express my gratitude to :

1. My Father, Mother, Sister, and my big family always pour down their support and prayer for me while building this final project.
2. Y.b.Dwi Setianto is a very kindful Supervising Lecturer who always kindly provides guidance, suggestion, and input so that this final project could be finished properly
3. My Friends, Bernardinus Hendra, Alfonsus Ferdiaz, and Samuel Kurniawan who always be there for me when I need help and keep on supporting me when I'm down.
4. Mr. Febrian Wicaksono and Mr. Widana give me the inspiration for the topic for this final project and help me when facing errors.
5. And all other persons that I can't mention one by one and have provided me myriad support.

Semarang, 15 Januari 2022



Alexander Jason Lauwren

ABSTRACT

When it comes to building or developing an online system, developers need to choose what kind of architecture will be used for the system. When facing the challenge, a developer needs the most suitable architecture that most fit the case whether uses microservices or monolithic architecture. Both architectures offer different benefits. Microservice recently become popular because many large companies start migrating from monolith to microservices but on the other hand, many organizations are still unfamiliar with microservices. Despite microservice providing many benefits, it also has challenges. With that being said, it is common that many organizations choose to stick with monolithic architecture since it was easier to maintain, develop, and deploy.

To find out the better architecture performance-wise, API provided by both application need to be tested. The test was conducted by hitting the API several times with many threads concurrently. The test result is latency and request time, success rate needs to be monitored as well since the error occurred during load testing. With the data collected it could be shown which architecture performs better.

The test results obtained by testing both monolith and microservice with several scenarios are quite unexpected. From the data, it turns out that for most features, average latency from the monolith is better than microservice. Meanwhile in many other scenarios, microservice edge monolith on the close gap in terms of success rate. The latency average number result for microservice is slightly worse because there are more success requests and taking more time.

Keyword: microservice, monolith, latency, performance

TABLE OF CONTENTS

COVER	i
APPROVAL AND RATIFICATION PAGE (Heading plain)	ii
DECLARATION OF AUTHORSHIP	iii
HALAMAN PERNYATAAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	iv
ACKNOWLEDGMENT	v
TABLE OF CONTENTS	vii
LIST OF FIGURE	ix
LIST OF TABLE	xiii
CHAPTER 1 INTRODUCTION	14
1.1. Background.....	14
1.2. Problem Formulation.....	15
1.3. Scope.....	15
1.4. Objective.....	15
CHAPTER 2 LITERATURE STUDY	16
CHAPTER 3 RESEARCH METHODOLOGY	20
3.1. Literature Study.....	20
3.2. Building Monolith Architecture Web Server.....	20
3.3. Building Microservice Architecture Web Server.....	20
3.4. Setting Up NGINX Server.....	20
3.5. Testing.....	21
3.6. Analyze.....	21
CHAPTER 4 ANALYSIS AND DESIGN	22
4.1. Analysis.....	22
4.2. Design.....	23

CHAPTER 5 IMPLEMENTATION AND RESULTS.....	32
5.1. Implementation.....	32
5.2. Results.....	42
CHAPTER 6 CONCLUSION.....	60
REFERENCES	62
APPENDIX	a



LIST OF FIGURE

Figure 4.1 Monolith Architecture Design	23
Figure 4.2 Flowchart Monolith Application Handling Single HTTP Request	24
Figure 4.3 Microservice Application Architecture.....	25
Figure 4.4 Microservice Flowchart Handling Single HTTP Request	26
Figure 4.5 Flowchart JMeter	28
Figure 4.6 Testing Design Where JMeter, Services, Database, and Proxy Server (NGINX) Deployed On a Single Device.....	29
Figure 4.7 Testing Design Where JMeter Separated From Main Server.....	29
Figure 4.8 Testing Design Where Services Server, JMeter, NGINX Are Separated	30
Figure 5.1 Monolith Versus Microservice Merchant Get Service Performance.....	42
Figure 5.2 Monolith Versus Microservice Insert Merchant Performance Chart.....	44
Figure 5.3 Monolith Versus Microservice Update Merchant Performance Chart	45
Figure 5.4 Monolith Versus Microservice Register User Performance Chart	46
Figure 5.5 Monolith Versus Microservice Login User Performance Chart.....	47
Figure 5.6 Monolith Versus Microservice Insert Bulk Transaction Performance Chart..	49
Figure 5.7 Error Message From JMeter Failed Thread	50
Figure 5.8 Design Test 1 VS Design Test 2 Get Merchant Performance Chart	51
Figure 5.9 Design 1 Versus Design 2 Login User Performance Chart.....	52
Figure 5.10 Monolith Versus Microservice Insert Bulk Transaction Performance Chart	53
Figure 5.11 Design Test 1 VS Design Test 2 and Design Test 3 Get Merchant Performance Chart.....	54
Figure 5.12 Design 1 Versus Design 2 and Design 3 Login User Performance Chart	55

Figure 5.13 Design 1 Versus Design 2 and Design 3 Insert Bulk Transaction Performance Chart.....	56
Figure 5.14 Overall Monolith Versus Microservice Average Latency Chart.....	57
Figure 5.15 Overall Monolith Versus Microservice Average Success Rate Chart	58
Figure 5.16 Testing Design Latency Comparison Chart	59
Figure 5.17 Testing Design Success Rate Comparison Chart.....	59
Figure 6.1 HTTP Header Manager JMeter	n
Figure 6.2 Loop Controller	o
Figure 6.3 Counter.....	o
Figure 6.4 Transaction Testing.....	p
Figure 6.5 Merchant Testing.....	p
Figure 6.6 User Testing.....	q
Figure 6.7 Microservice 100 – Get Merchant	q
Figure 6.8 Microservice 1000 – Get Merchant	q
Figure 6.9 Microservice 5000 – Get Merchant	q
Figure 6.10 Monolith 100 – Get Merchant	q
Figure 6.11 Monolith 1000 – Get Merchant	q
Figure 6.12 Monolith 5000 – Get Merchant	r
Figure 6.13 Monolith 100 - Insert Merchant.....	r
Figure 6.14 Monolith 1000 – Insert Merchant	r
Figure 6.15 Monolith 5000 – Insert Merchant	r
Figure 6.16 Microservice 100 – Insert Merchant	r
Figure 6.17 Microservice 1000 – Insert Merchant	r
Figure 6.18 Microservice 5000 – Insert Merchant	r

Figure 6.19 Microservice 100 – Update Merchant.....	r
Figure 6.20 Microservice 1000 – Update Merchant.....	r
Figure 6.21 Microservice 5000 – Update Merchant.....	s
Figure 6.22 Monolith 100 – Update Merchant.....	s
Figure 6.23 Monolith 1000 – Update Merchant.....	s
Figure 6.24 Monolith 5000 – Update Merchant.....	s
Figure 6.25 Monolith 100 – Register User	s
Figure 6.26 Monolith 1000 – Register User.....	s
Figure 6.27 Microservice 100 – Register User	s
Figure 6.28 Microservice 1000 – Register User.....	s
Figure 6.29 Microservice 100 – Login Request.....	s
Figure 6.30 Microservice 1000 – Login Request.....	s
Figure 6.31 Monolith 100 – Login Request.....	t
Figure 6.32 Monolith 1000 – Login Request.....	t
Figure 6.33 Monolith 100 – Insert Trx	t
Figure 6.34 Monolith 1000 – Insert Trx	t
Figure 6.35 Monolith 5000 – Insert Trx	t
Figure 6.36 Microservice 100 – Insert Trx	t
Figure 6.37 Microservice 1000 – Insert trx.....	t
Figure 6.38 Microservice 5000 – Insert trx.....	t
Figure 6.39 100 monolith - Get merchant.....	t
Figure 6.40 1000 monolith - Get Merchant	u
Figure 6.41 5000 monolith - Get Merchant	u
Figure 6.42 100 microservice - Get Merchant	u

Figure 6.43 1000 microservice - Get Merchant	u
Figure 6.44 5000 microservice - Get Merchant	u
Figure 6.45 100 microservice - Login User	u
Figure 6.46 1000 microservice - Login User	u
Figure 6.47 100 monolith - Login User	u
Figure 6.48 1000 monolith - Login User	u
Figure 6.49 100 monolith - Insert trx.....	v
Figure 6.50 1000 monolith - Insert trx.....	v
Figure 6.51 5000 monolith - Insert trx.....	v
Figure 6.52 100 microservice - Insert trx.....	v
Figure 6.53 1000 microservice - insert trx.....	v
Figure 6.54 5000 microservice - insert trx.....	v
Figure 6.55 100 Design 3 - insert trx.....	v
Figure 6.56 1000 Design 3 - insert trx.....	v
Figure 6.57 5000 Design 3 - insert trx.....	w
Figure 6.58 100 Design 3 - Login User	w
Figure 6.59 1000 Design 3 - Login User	w
Figure 6.60 100 Design 3 - Merchant Get	w
Figure 6.61 1000 Design 3 - Merchant Get	w
Figure 6.62 5000 Design 3 - Merchant Get	w

LIST OF TABLE

Table 5.1. Detailed Get Merchant Performance Data on Both Architecture	43
Table 5.2. Detailed Insert Merchant Performance Data on Both Architecture	44
Table 5.3. Detailed Update Merchant Performance Data on Both Architecture	45
Table 5.4. Detailed Register User Performance Data on Both Architecture.....	47
Table 5.5. Detailed Login User Performance Data on Both Architecture	48
Table 5.6. Detailed Insert Bulk Transaction Performance Data on Both Architecture	49
Table 5.7. Detailed Get Merchant Performance Data on Both Architecture	51
Table 5.8. Detailed Login User Performance Data on Both Architecture	52
Table 5.9. Detailed Insert Bulk Transaction Performance Data on Both Architecture	53
Table 5.10. Detailed Get Merchant Performance Data on Design Test 2 and Design Test 3	54
Table 5.11. Detailed Design 2 and Design 3 Login User Performance	55
Table 5.12. Detailed Design 2 and Design 3 Insert Bulk Transaction Performance	56
Table 5.13. Overall Average Services Latency	57
Table 5.14. Overall Average Latency and Success Rate Every Test Design.....	58