**ARSITEKTUR *MICROSERVICE* DALAM RANCANG BANGUN APLIKASI SISTEM INFORMASI DEPARTEMEN ILMU KOMPUTER UNIVERSITAS DIPONEGORO**

****

**PROPOSAL PENELITIAN**

**Disusun sebagai salah satu syarat**

**untuk memenuhi Ujian Tengah Semester Genap**

**mata kuliah Metodologi dan Penulisan Ilmiah yang**

**diampu oleh Dr. Retno Kusumaningrum, S.Si, M.Kom**

**Disusun oleh:**

**Ananda Prabu Tritya Vijaya**

**24060117130048**

**DEPARTEMEN ILMU KOMPUTER/ INFORMATIKA**

**FAKULTAS SAINS DAN MATEMATIKA**

**UNIVERSITAS DIPONEGORO**

**2020**

# ABSTRAK

Sistem Informasi adalah salah satu aspek penting dalam proses teknis dan administratif perkuliahan. Beberapa sistem informasi tersebut menggunakan aplikasi untuk memudahkan prosesnya. Tetapi terdapat beberapa kekurangan yang ada pada aplikasi sistem infromasi yang terdapat di Departemen Ilmu Komputer / Informatika Fakultas Sains dan Matematika Universitas Diponegoro yaitu tidak adanya integrasi satu sama lain yang menyebabkan tidak menjamin data yang ada sinkron, dan juga untuk pengembangan kedepannya saat sistem semakin besar maka akan susah diatur dan dikembangkan. Oleh karena itu penelitian ini dilakukan dengan tujuan untuk membangun ulang aplikasi sistem informasi dasar sebagai akar dari seluruh aplikasi lainnya dengan menggunakan arsitektur *Microservice* dan untuk mengukur seberapa efisien penggunaan arsitektur ini dibandingkan dengan arsitektur *Monolithic* dalam pengembangan aplikasi berikutnya dan *maintenance*. Diharapkan dengan penggunaan arsitektur *Microservice* ini aplikasi sistem informasi yang ada di Departemen Ilmu Komputer / Informatika Fakultas Sains dan Matematika Universitas Diponegoro dapat dirawat dan dikembangkan dengan mudah, serta dapat memberikan data tentang seberapa efisien pengembangan aplikasi sistem informasi menggunakan arsitektur *Microservice* dibandingkan menggunakan arsitektur *Monolithic*.

**Kata Kunci:** *Software Development,* Sistem Informasi, *Microservice Architecture, Monolithic Architecture*

# DAFTAR ISI

[ABSTRAK i](#_Toc37905394)

[DAFTAR ISI ii](#_Toc37905395)

[DAFTAR GAMBAR iii](#_Toc37905396)

[DAFTAR TABEL iv](#_Toc37905397)

[BAB I PENDAHULUAN 5](#_Toc37905398)

[1.1. Latar Belakang 5](#_Toc37905399)

[1.2. Rumusan Masalah 6](#_Toc37905400)

[1.3. Tujuan dan Manfaat 6](#_Toc37905401)

[1.4. Ruang Lingkup 6](#_Toc37905402)

[BAB II LANDASAN TEORI 7](#_Toc37905403)

[2.1. *State of The Art* 7](#_Toc37905406)

[2.2. Arsitektur *Monolithic* 8](#_Toc37905407)

[2.3. Arsitektur *Microservice* 9](#_Toc37905408)

[2.4. *Application Programming Interface* (API) 9](#_Toc37905409)

[2.5. *Waterfall Development* 10](#_Toc37905410)

[BAB III METODOLOGI PENELITIAN 12](#_Toc37905411)

[3.1. Bahan Penelitian 12](#_Toc37905415)

[3.2. Alat Penelitian 12](#_Toc37905416)

[3.3. Metode Penelitian 12](#_Toc37905417)

[BAB IV RENCANA JADWAL PENELITIAN 14](#_Toc37905418)

[4.1. Jadwal Penelitian 14](#_Toc37905422)

[DAFTAR PUSTAKA 15](#_Toc37905423)

# DAFTAR GAMBAR

[Gambar 2.2.1 Ilustrasi Arsitektur Monolithic 8](#_Toc37905424)

[Gambar 2.3.1 Ilustrasi Arsitektur Microservice 9](#_Toc37905425)

[Gambar 2.4.1 Ilustrasi Application Programming Interface 9](#_Toc37905426)

[Gambar 3.1.1 Metodologi Waterfall 10](#_Toc37905427)

# DAFTAR TABEL

[Tabel 4.1.1 Jadwal Penelitian 14](#_Toc37905428)

# BAB I PENDAHULUAN

## Latar Belakang

Sistem Informasi adalah salah satu aspek penting dalam proses teknis dan administratif perkuliahan. Oleh karena itu ada beberapa sistem informasi yang berbasis aplikasi di Departemen Ilmu Komputer / Informatika Fakultas Sains dan Matematika Universitas Diponegoro, seperti aplikasi sistem informasi yang berhubungan dengan Praktek Kerja Lapangan (PKL), dan juga ada aplikasi yang diurus oleh Himpunan Mahasiswa Informatika seperti "Bukulink” untuk mencatat seluruh identitas mahasiswa dan dosen yang ada di Departemen Ilmu Komputer / Informatika. Sayangnya, aplikasi-aplikasi sistem informasi yang ada tidak terintegrasi satu sama lain sehingga data-data dan sistem tidak tersinkronisasi yang akan rentan mengakibatkan perbedaan data.

Tidak adanya integrasi antar aplikasi satu sama lain, akan mengakibatkan beberapa masalah, diantaranya adalah jika ada sistem informasi baru yang ingin dibuat maka *development* / pengembangan akan memakan waktu yang relatif lama, dan juga aplikasi akan susah jika ingin dikembangkan (Sani et al., 2019). Oleh karena itu, dengan menggunakan arsitektur *microservice* dalam pembangunan aplikasi sistem informasi akan membuat semua data dan sistem terintegrasi satu sama lain, dan diharapkan untuk pengembangan selanjutnya akan menjadi lebih mudah (Roca et al., 2020). Akan tetapi, ada beberapa masalah yang dihadapi dengan menggunakan arsitektur *microservice* ini, salah satunya adalah kerumitan dari arsitektur ini yang mengharuskan ada perencanaan yang sangat matang sebelum membangun sistem menggunakan arsitektur ini.

Oleh karena itu, untuk menutupi kelemahan dari arsitektur *microservice* tersebut, penelitian ini menggunakan metodologi *Waterfall Development* yang diharapkan dapat mengatur alur dari pengembangan sistem ini dari awal hingga ke akhir secara rinci dan tepat setiap *life cycle* nya sehingga dapat meminimalisir adanya kesalahan di tengah-tengah waktu pengembangan sistem.

## Rumusan Masalah

* + 1. Seberapa efisien implementasi aristektur *microservice* dalam pengembangan aplikasi sistem informasi di Departemen Ilmu Komputer / Informatika Fakultas Sains dan Matematika Universitas Diponegoro dibandingkan arsitektur *monolithic?*

## Tujuan dan Manfaat

* + 1. Tujuan
       1. Mengetahui seberapa efisiensi implementasi arsitektur *microservice* dalam pengembangan aplikasi sistem informasi di Departemen Ilmu Komputer / Informatika Fakultas Sains dan Matematika Universitas Diponegoro.
    2. Manfaat
       1. Untuk Departemen, pengembangan aplikasi selanjutnya akan lebih terstruktur dan lebih mudah
       2. Untuk Mahasiswa, menambah ilmu tentang arsitektur *microservice* dan mendapat nilai

## Ruang Lingkup

* + 1. Permasalahan yang dibahas

Permasalahan yang akan dibahas dalam penelitian ini adalah rancang bangun aplikasi sistem informasi utama yang akan menjadi dasar atau akar (*root*) dari aplikasi aplikasi sistem informasi lainnya dan pembahasan tentang seberapa efisien arsitektur *microservice* dalam pembangunan aplikasi sistem informasi ini.

* + 1. Batasan Penyelesaian

Penelitian akan dikatakan selesai jika aplikasi sistem informasi ‘*root’* ini sudah diselesaikan dan telah membandingkan efisiensi arsitektur *microservice* dengan *monolithic* yang selama ini digunakan untuk membuat aplikasi sistem informasi di Departemen Ilmu Komputer / Informatika Fakultas Sains dan Matematika Universitas Diponegoro dari berbagai aspek.

* + 1. Data yang dikumpulkan

Data yang dikumpulkan berupa *source code* dari aplikasi sistem informasi ‘*root*’ dan juga data statistik tentang efisiensi arsitektur *microservice* dibandingkan dengan arsitektur *monolithic*.

# BAB II LANDASAN TEORI

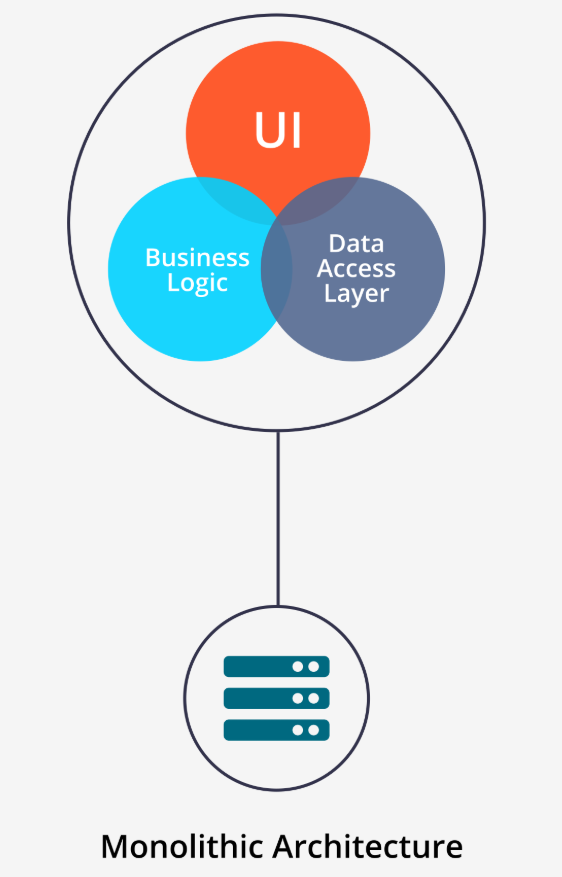


## *State of The Art*

Penelitian tentang pengembangan aplikasi di bidang *microservice* sudah pernah dilakukan sebelumnya. Adapun tujuan dari beberapa penelitian tersebut diantaranya adalah untuk meningkatkan kualitas dari aplikasi, untuk membuat suatu *chatbot*, dan sebagainya. Beberapa jurnal yang melakukan penelitian tentang *microservice* diantaranya:

* + 1. Penelitian dengan judul “*Development of Microservice Based Application E-Inkubator Incubation and Investment Service Provider for SMEs*”dengan peneliti yang bernama Nisfu Asrul Sani\*, Wildan Azka Fillah, Aris Tjahyanto, dan Hatma Suryotrisongko. Penelitian ini bertujuan untuk mengembangkan aplikasi yang sudah ada dimana aplikasi itu awalnya dibuat dengan menggunakan arsitektur *monolithic* dan dikembangkan dengan menggunakan arsitektur *microservice.* Tujuan dari aplikasi ini adalah untuk membuat suatu aplikasi sistem informasi yang lebih fleksibel dan juga membandingkan kecepatan aplikasi dengan menggunakan arsitektur *monolothic* dan *microservice.*
    2. Penelitian dengan judul “*Microservice chatbot architecture for chronic patient support*” yang diteliti oleh Surya Roca⁎, Jorge Sancho, José García, dan Álvaro Alesanco. Penelitian ini berisi tentang pemanfaatan kualitas kecepatan dari arsitektur *Microservice* dalam pembuatan *chatbot* dimana tiap *service* memiliki tugas sendiri sendiri yang bertujuan untuk memperingan komputasi dari setiap *service*. Peneliti juga mengharapkan dengan menggunakan arsitektur ini kedepannya *chatbot* ini dapat dikembangkan lebih jauh lagi untuk membantu pasien.

## Arsitektur *Monolithic*

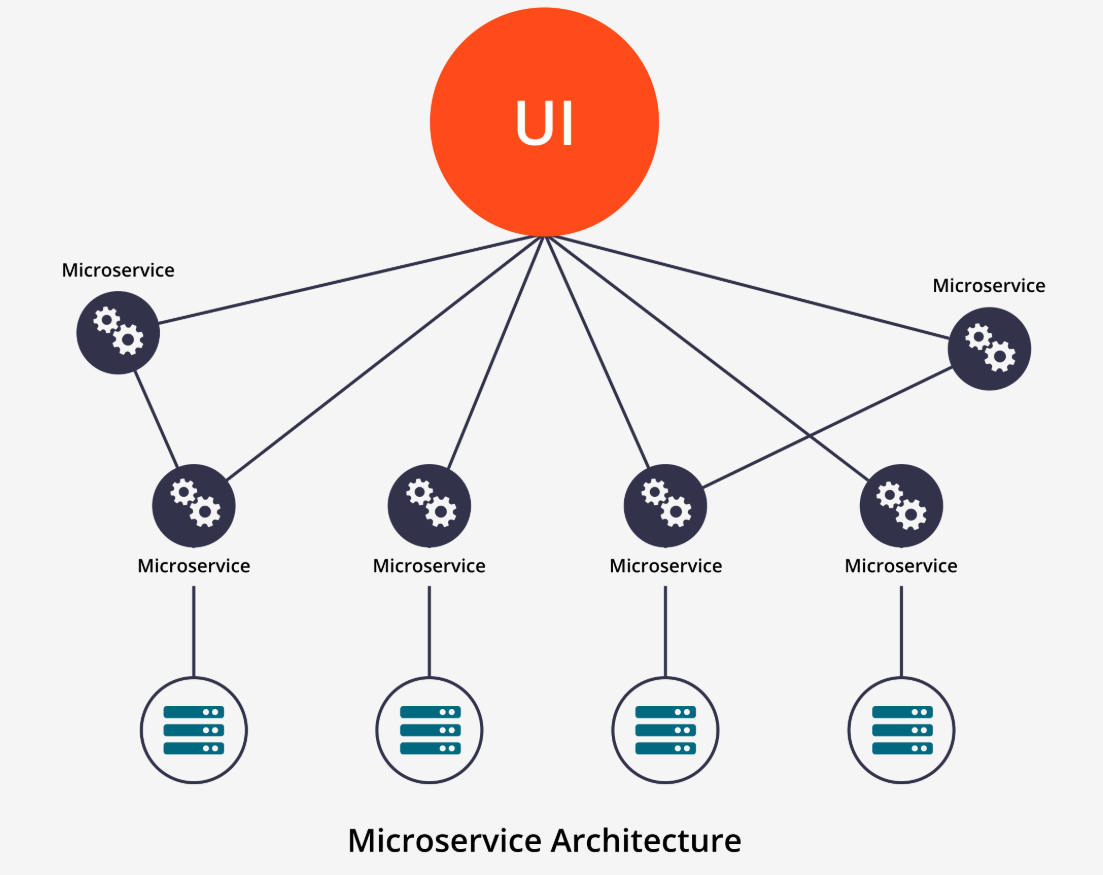


Gambar 2.2.1 Ilustrasi Arsitektur Monolithic

Arsitektur *Monolithic* adalah arsitektur dimana pembuatan aplikasi / sistem dibuat dengan cara dimana semua komponen aplikasi dibuat, didesain, diimplementasi, dan dirawat di dalam satu kode unit (Sani et al., 2019). Aplikasi atau sistem yang dibuat menggunakan arsitektur *Monolithic* bisa saja berjalan pada proses-proses yang berbeda yang didistribusikan ke beberapa *CPU* tetapi tetap saja ini masih masuk ke dalam satu sistem operasi dan perangkat keras (Götz et al., 2018).

Kelemahan dari arsitektur ini adalah disaat ada pengembangan berkelanjutan akan membuat tim pengembang akan mengalami kesulitan karena saat ada suatu komponen yang di *upgrade,* komponen lain yang ada di dalam satu kode unit tersebut harus di *upgrade* juga (Wan et al., 2018). Terdapat kekurangan lain dimana sekecil apapun perubahan atau *update* dari suatu aplikasi diharuskan untuk *testing* dan *deploy* ulang keseluruhan kode unit (Abdullah et al., 2019).

## Arsitektur *Microservice*

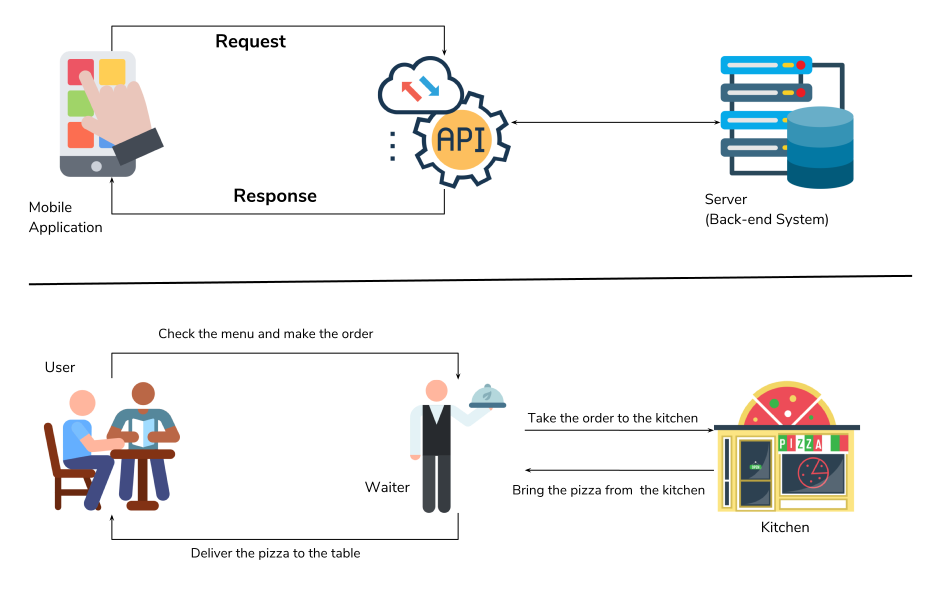


Gambar 2.3.1 Ilustrasi Arsitektur Microservice

Arsitektur *Microservice* adalah arsitektur dimana pembuatan aplikasi / sistem dipecah menjadi beberapa *service* tersendiri dimana *services* tersebut tidak terlalu mengikat satu sama lain yang mengakibatkan setiap *service* dapat berjalan sendiri sehingga dalam pe(Abdullah et al., 2019; Roca et al., 2020; Sani et al., 2019).

Beberapa alasan mengapa orang menggunakan arsitektur ini adalah skalabilitas, kecepatan, dapat menggunakan teknologi yang berbeda-beda, dan juga kemudahan saat ada *upgrade* dan *maintenance* (Huang et al., 2019; Madnani et al., 2018; Wu et al., 2018).

## *Application Programming Interface* (API)

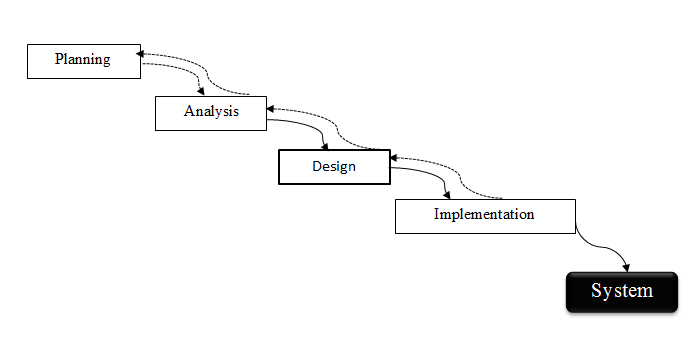


Gambar 2.4.1 Ilustrasi Application Programming Interface

*Application Programming Interface* atau yang sering disingkat API adalah *service* dan *function* yang tersedia di suatu program atau aplikasi yang bisa diakses secara otomatis maupun manual, yang mendeskripsikan apa saja *service* yang tersedia, *input* atau *parameter* yang dimasukkan, output seperti apa yang akan dihasilkan, dan protokol-protokol dalam menggunakan *service* tersebut. Adapun keuntungan dari penggunaan API adalah kemudahan dalam mengakses dan bertukar data antar sistem / program / aplikasi (Selby et al., 2019).

Penggunaan API sangatlah erat dengan arsitektur *Microservice* karena *Microservice* menggunakan API untuk saling berkomunikasi dan bertukar data antara *service*nya (Zimmermann et al., 2020).

## *Waterfall Development*



Gambar 3.1.1 Metodologi Waterfall

*Waterfall Development* adalah metodologi dimana dalam mengembangkan aplikasi / program memiliki 5 fase, yaitu *Planning, Analysis, Design,* dan *Implementation* dimana setiap fasenya diselesaikan dulu 100% sebelum menuju fase berikutnya . Keuntungan dari metodologi ini adalah rencana yang akan dieksekusi akan matang dan meminimalisir perubahan rencana kedepannya (Dennis et al., 2012). Dennis et al., (2012) menerangkan bahwa berikut adalah penjelasan dari tiap fase:

* + 1. Planning

Fase ini adalah fase yang fundamental karena disini tim pengembang memikirkan dan memahami kenapa sistem harus dibuat dan kemana arah dari proyek ini.

* + 1. Analysis

Fase ini menjawab pertanyaan siapa yang menggunakan sistem, apa yang akan sistem lakukan, kapan dan di mana sistem akan digunakan.

* + 1. Design

Fase ini akan menentukan bagaimana sistem akan beroperasi, mulai dari perangkat keras, perangkat lunak, dan infrastrukturnya.

* + 1. Implementation

Di fase ini terdapat berbagai kegiatan seperti *system construction, testing (unit, integration, system,* dan *user acceptance), user documentation, system documentation,* dan juga *installation.*

* + 1. System

*Maintenance* dan *modification* dilakukan di fase ini.

# BAB III METODOLOGI PENELITIAN



## Bahan Penelitian

Data yang dibutuhkan dalam menyelesaikan penelitian ini adalah tata cara kerja sistem informasi yang ada di Departemen Ilmu Komputer / Informatika Fakultas Sains Dan Matematika Universitas Diponegoro yang akan didapatkan melalui wawancara langsung ke pihak departemen, dan juga data berupa *timeline* dan *logbook* hasil pembuatan aplikasi sistem informasi yang telah dibuat sebelumnya di Departemen Ilmu Komputer / Informatika Fakultas Sains Dan Matematika Universitas Diponegoro dengan tujuan untuk membandingkan efisiensi dari kedua arsitektur.

## Alat Penelitian

*Tools* yang digunakan untuk penelitian ini diantaranya:

* + 1. *Container*

*Container* digunakan karena isolasi setiap *service* yang lebih baik, eksekusi yang lebih cepat, dan juga isolasi eksekusi dilakukan di tingkat sistem operasi yang dapat mengurangi *overhead* dan juga komponen aplikasi terbebas dari komputasi yang berlebih.

* + 1. *API Gateaway*

*API Gateaway* digunakan untuk setiap *service* dapat melakukan komunikasi data.

* + 1. Bahasa pemrograman: Golang

## Metode Penelitian

Metode penelitian yang digunakan adalah *Waterfall*. Metode ini digunakan untuk memaksimalkan analisis di bagian infrastruktur yang nantinya berguna untuk meminimalisir kesalahan di pertengahan fase pengembangan. Hal ini dilakukan karena arsitektur *Microservice* memiliki kekurangan dimana analisis dan infrastruktur dari arsitektur ini rumit.

* + 1. Planning

Di fase ini dibuat beberapa perencanaan seperti *project charter, cost, time,* dan membuat perencanaan kerja.

* + 1. Analysis

Di fase ini dilakukannya *Requirement gathering* dimana peneliti mewawancarai *stackholder* (pihak departemen) yang nanti hasilnya adalah analisis fungsional dan non fungsional, dan juga di fase ini akan menghasilkan *Business Process Modeling*.

* + 1. Design

Di dalam fase ini, peneliti akan menentukan bagaimana arsitektur dari program ini, mulai dari desain antarmuka, desain sistem, dan desain data.

* + 1. Implementation

Di fase ini, peneliti akan melakukan *testing,* membuat dokumentasi, dan melakukan instalasi program.

* + 1. System

Di fase ini, peneliti akan melakukan *maintenance* jika terjadi kesalahan atau error terhadap program yang telah selesai dibuat

* + 1. Benchmarking

Di fase ini, peneliti akan melakukan beberapa analisis perbedaan antara program yang dibuat dengan menggunakan arsitektur *monolithic* dan *microservice*

# BAB IV RENCANA JADWAL PENELITIAN



## Jadwal Penelitian

Penelitian ini akan dilakukan selama 21 minggu dengan rincian:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | Minggu | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 1 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | *Planning* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | *Analysis* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | *Design* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | *Implementation* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | *System* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | *Benchmarking* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Penyusunan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Tabel 4.1.1 Jadwal Penelitian

# DAFTAR PUSTAKA

Abdullah, M., Iqbal, W., & Erradi, A. (2019). Unsupervised learning approach for web application auto-decomposition into microservices. *Journal of Systems and Software*, *151*, 243–257. https://doi.org/10.1016/j.jss.2019.02.031

Dennis, A., Wixom, B. H., & Roth, R. M. (2012). System Analysis and Design 5th Edition. In *John Wiley & Sons, Inc.*

Götz, B., Schel, D., Bauer, D., Henkel, C., Einberger, P., & Bauernhansl, T. (2018). Challenges of Production Microservices. *Procedia CIRP*, *67*, 167–172. https://doi.org/10.1016/j.procir.2017.12.194

Huang, L., Zhang, C., & Zeng, Z. (2019). Design of a public services platform for university management based on microservice architecture. *Microsystem Technologies*, *4*(Loukides 2012). https://doi.org/10.1007/s00542-019-04474-4

Madnani, N., Cahill, A., Blanchard, D., Andreyev, S., Napolitano, D., Gyawali, B., Heilman, M., Lee, C. M., Leong, C. W., Mulholland, M., & Riordan, B. (2018). A Robust Microservice Architecture for Scaling Automated Scoring Applications. *ETS Research Report Series*, *2018*(1). https://doi.org/10.1002/ets2.12202

Roca, S., Sancho, J., García, J., & Alesanco, Á. (2020). Microservice chatbot architecture for chronic patient support. *Journal of Biomedical Informatics*, *102*(September 2019), 103305. https://doi.org/10.1016/j.jbi.2019.103305

Sani, N. A., Fillah, W. A., Tjahyanto, A., & Suryotrisongko, H. (2019). Development of microservice based application e-inkubator: Incubation and investment service provider for SMEs. *Procedia Computer Science*, *161*, 1064–1071. https://doi.org/10.1016/j.procs.2019.11.217

Selby, P., Abbeloos, R., Backlund, J. E., Basterrechea Salido, M., Bauchet, G., Benites-Alfaro, O. E., Birkett, C., Calaminos, V. C., Carceller, P., Cornut, G., Vasques Costa, B., Edwards, J. D., Finkers, R., Yanxin Gao, S., Ghaffar, M., Glaser, P., Guignon, V., Hok, P., Kilian, A., … Wren, J. (2019). BrAPI - An application programming interface for plant breeding applications. *Bioinformatics*, *35*(20), 4147–4155. https://doi.org/10.1093/bioinformatics/btz190

Wan, X., Guan, X., Wang, T., Bai, G., & Choi, B. Y. (2018). Application deployment using Microservice and Docker containers: Framework and optimization. *Journal of Network and Computer Applications*, *119*(July), 97–109. https://doi.org/10.1016/j.jnca.2018.07.003

Wu, S., Wang, X., & Zhu, Q. (2018). Design of WeChat Service System Based on Microservice Architecture. *Journal of Physics: Conference Series*, *1069*(1), 0–8. https://doi.org/10.1088/1742-6596/1069/1/012087

Zimmermann, O., Stocker, M., Lübke, D., Pautasso, C., & Zdun, U. (2020). Introduction to microservice API patterns (MAP). *OpenAccess Series in Informatics*, *78*(4), 1–4. https://doi.org/10.4230/OASIcs.Microservices.2017-2019.4