

**Implementasi *REST API* Dengan Arsitektur *Microservices* Terhadap
Pendataan Kegiatan *Workshop* Menggunakan *Framework Springboot***



**PERBANAS
INSTITUTE**

OUTLINE SKRIPSI

Disusun Untuk Memenuhi Sebagai Persyaratan Guna Memperoleh Sarjana
Komputer

Oleh:

Dwi Putra Ramadhan

NIM: 1914000025

Program studi: Teknik Informatika

Jenjang Pendidikan: Strata satu (S-1)

INSTITUT KEUANGAN PERBANKAN DAN INFORMATIKA ASIA

PERBANAS

JAKARTA

2023

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang sangat signifikan pada era evolusi 5.0 menyebabkan munculnya beragam revolusi dan inovasi terhadap perkembangan zaman khususnya pada suatu sistem yang berjalan. Tingginya pengguna teknologi informasi semakin banyak pengguna yang menemukan inovasi terbaru. Inovasi tersebut dikembangkan menjadi sedemikian rupa agar dapat berguna untuk ormawa HIMATIKA. Teknologi-teknologi yang beredar semakin mudah digunakan untuk membangun suatu sistem sehingga *developer* menjadi leluasa dalam menentukan bagaimana proses pembuatannya. Pada akhirnya penentuan sistem yang akan digunakan *developer* akan menentukan apakah sistem yang dibuatnya dapat lebih unggul dibanding dengan sistem lainnya.

Ormawa HIMATIKA merupakan organisasi yang bergerak dibidang IT (*Information and Technology*) dimana pada salah satu divisi yang bergerak dalam pengembangan sumber daya manusia memiliki *jobdesc* untuk mengembangkan *softskill* dan *hardskill* mahasiswa dalam kegiatan *workshop*. Kegiatan tersebut tentunya membutuhkan *management*, pengelolaan, serta persiapan secara baik dan matang oleh karena itu dibutuhkan sistem yang digunakan untuk meng-*input* data serta menyimpan arsip dan dokumentasi pada divisi tersebut. Maka dari itu, peneliti akan membuat sistem untuk meng-*input* data serta menyimpan arsip pada divisi tersebut yang menggunakan arsitektur *microservices* untuk pendataan kegiatan *workshop* dan *REST API* sebagai penghubung komunikasi antar masing-masing sistem.

REST API adalah arsitektur perangkat lunak yang memberlakukan syarat mengenai cara *API* bekerja. *REST API* pada awalnya dibuat sebagai panduan untuk mengelola komunikasi pada jaringan kompleks seperti internet. Peneliti dapat menggunakan arsitektur berbasis *REST* untuk mendukung komunikasi berperforma tinggi dan andal sesuai skala. *REST API* memiliki keunggulan dapat dengan mudah untuk menerapkan dan memodifikasinya, membawa visibilitas dan portabilitas lintas *platform* ke semua sistem *API*.

Microservices adalah membagi *service* ke kumpulan proses independen dan kecil yang berkomunikasi antara satu dengan lainnya untuk membentuk *service* kompleks yang

agnostik terhadap bahasa *API* apa pun. *Service-service* ini dibuat menjadi bagian-bagian terpisah yang dibangun sekomples mungkin hingga menjadi blok-blok kecil, dan fokus pada tugas-tugas ringan untuk memfasilitasi metode modular dalam pembangunan sistem. Arsitektur bergaya *microservices* mulai menjadi standar dalam pembangunan sistem yang dinamis dan konstan berkembang. Secara sederhana masih banyak sekali sistem yang berjalan menggunakan sistem pendekatan *monolithic*. Ketika membangun dan penerapannya ke suatu sistem menggunakan arsitektur *monolithic* berbagai masalah akan muncul dari waktu ke waktu. Pada akhirnya, saat ini developer-developer melirik arsitektur *microservices* untuk membangun suatu sistem, karena banyak sekali kekurangan yang dimiliki oleh arsitektur *monolithic* dapat diatasi oleh arsitektur *microservices*.

Kelebihan dari arsitektur *microservices* ini adalah ketika ada perubahan yang diperlukan pada bagian tertentu, arsitektur ini hanya mengubah *service* atau fitur terkait yang perlu dimodifikasi sehingga kita tidak perlu menerapkannya ke seluruh fitur atau *service* yang sedang berjalan. Kemudian, mudah untuk di-*upgrade* dan diatur sesuai dengan kebutuhan pengguna (*scalable*). Pada bagian terpenting adalah jika satu *microservices* gagal, *service* yang lainnya akan tetap terus berjalan. Namun, dari aspek kelebihan metode ini memiliki kekurangan. Kekurangan arsitektur ini adalah proses *deployment microservices* yang umumnya tersebar, pada akhirnya proses testing bisa menjadi sedikit lebih rumit dan membutuhkan banyak proses. Serta sistem yang dimiliki *microservice* lebih rumit dikarenakan arsitektur *microservices* memiliki kompleksitas yang tinggi karena developer harus mengurangi adanya *error*, latensi jaringan, dan lain-lain.

Hasil dari penelitian ini adalah dapat membantu ormawa HIMATIKA dalam meng-*input* perencanaan *workshop*, meng-*input* berkas selama kegiatan berlangsung, meng-*input* absensi untuk peserta dalam kegiatan *workshop*, dan menyimpan dokumentasi selama kegiatan *workshop* berlangsung.

1.2 Masalah Penelitian

1.2.1 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka peneliti mengidentifikasi suatu masalah yaitu “Pada saat ini ormawa HIMATIKA khususnya pada divisi pengembangan sumber daya manusia menggunakan penyimpanan untuk pendataan melalui google

drive yang masih kurang efektif. Dikarenakan proses tersebut harus membuat folder baru di setiap kegiatannya, sehingga dalam proses pendataan kurang efektif dan efisien.”

1.2.2 Batasan Masalah

Batasan masalah dari penelitian ini yaitu:

1. Penelitian ini dilakukan sampai sistem agar dapat meng-*input* perencanaan *workshop*, meng-*input* berkas selama kegiatan berlangsung, meng-*input* data absensi untuk peserta serta menyimpan arsip pada divisi tersebut yang menggunakan arsitektur *microservices* untuk pendataan kegiatan *workshop*;
2. Penerapan arsitektur *microservices* pada penelitian hanya fokus kepada API yang dapat menghubungkan antara sistem dengan *database*;
3. Sistem yang dibangun untuk diterapkan pada aplikasi berbasis web;
4. Sistem yang dibangun tidak menghasilkan *report*.

1.2.3 Rumusan Masalah

Berikut rumusan masalah berdasarkan latar belakang yang tertera diatas:

1. Bagaimana mengimplementasikan REST API menggunakan arsitektur *microservices* untuk membantu ormawa HIMATIKA untuk pendataan kegiatan *workshop*?
2. Layanan apa saja yang akan dibuat untuk memenuhi pendataan kegiatan *workshop* di ormawa HIMATIKA?

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan dari penelitian

Berikut rumusan masalah berdasarkan latar belakang yang tertera diatas:

1. Tujuan dari penelitian ini berguna untuk membantu anggota divisi yang akan melakukan pendataan dalam kegiatan *workshop*.
2. Dapat membuat sebuah sistem dengan konsep *microservice* yang dihubungkan dengan *REST API*.

1.3.2 Manfaat Penelitian

Manfaat dari penelitian ini untuk membantu ormawa HIMATIKA dalam pendataan kegiatan *workshop* dalam meng-*input* perencanaan *workshop*, meng-*input* berkas selama kegiatan berlangsung, meng-*input* data absensi untuk peserta serta menyimpan

arsip pada divisi tersebut, serta dapat mempermudah developer dalam mengembangkan sebuah aplikasi yang *high performance* dan memudahkan *developer* untuk mengembangkan setiap *service* tanpa mengganggu *service* lain.

1.4 Sistematika Penulisan

Proses yang digunakan penulisan dalam penelitian ini untuk menyusun skripsi dalam bentuk yang sistematis dan logis dibagi menjadi lima bagian, yaitu:

BAB I: PENDAHULUAN

Pada bab ini berisi latar belakang, identifikasi masalah, batasan masalah, rumusan masalah, tujuan penelitian serta manfaat penelitian yang memberikan informasi sebagai pendukung terkait penelitian yang dibahas.

BAB II: LANDASAN TEORI DAN KERANGKA KONSEP

Pada bab ini berisi landasan teori yang membahas informasi dasar sistem atau aplikasi yang digunakan oleh peneliti, serta tinjauan literatur penelitian terdahulu sebagai pendukung penelitian ini dibuat.

BAB III: METODE PENELITIAN

Pada bab ini berisi tentang metode-metode yang dilakukan dalam implementasi REST API dengan arsitektur microservice menggunakan framework springboot.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini berisikan uraian tentang hasil implementasi sistem pemanfaatan rest api dan microservice dalam sistem pendataan kegiatan workshop serta melakukan pengujian sistem agar sesuai dengan fungsinya

BAB V : PENUTUP

Pada bab ini berisikan kesimpulan dari hasil penelitian yang dilakukan dan saran-saran terhadap kekurangan dari penelitian tersebut.

BAB II

LANDASAN TEORI DAN KERANGKA KONSEP

Pada bab ini akan membahas tentang teori penunjang dan penelitian sebelumnya yang berhubungan dengan implementasi REST API menggunakan arsitektur *microservice* dan beberapa komponen yang digunakan untuk membangun sistem pencarian pengobatan alternatif.

2.1 REST API

REST API (*Representational State Transfer Application Programming Interface*) adalah sebuah standar untuk mengembangkan aplikasi web yang memungkinkan untuk berkomunikasi dengan layanan lain melalui internet. Implementasi REST API memungkinkan sistem untuk berkomunikasi dengan layanan lain dan memanfaatkan informasi yang tersedia untuk menyediakan layanan yang lebih baik bagi pengguna.

2.2 Arsitektur *Microservice*

Arsitektur *microservice* adalah sebuah pendekatan untuk mengembangkan aplikasi yang terdiri dari beberapa layanan kecil yang beroperasi secara independen dan dapat digabungkan untuk membentuk sebuah aplikasi utuh. Arsitektur ini memungkinkan untuk mengembangkan dan memperbaharui setiap layanan secara independen, sehingga mempermudah proses pemeliharaan dan perbaikan sistem.

2.3 Metode *Software Development Live Cycle* (SDLC)

Metode SDLC (*Software Development Live Cycle*) adalah sebuah metode yang digunakan untuk membangun atau merancang sebuah aplikasi/*software* karena memiliki tahapan yang jelas serta fleksibel dari tahapan awal hingga tahapan akhir, tahapannya terdiri dari rencana(*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*)

2.4 StarUML

StarUML adalah perangkat lunak pemodelan visual yang digunakan oleh para pengembang perangkat lunak untuk merancang dan memodelkan aplikasi atau sistem. StarUML mendukung berbagai jenis diagram seperti diagram kelas, diagram use case, diagram aktivitas, dan banyak lagi, yang membantu para pengembang dalam merencanakan, merancang, dan membangun aplikasi. StarUML juga dapat digunakan untuk menghasilkan

kode program dari model yang dibuat serta mendukung beberapa bahasa pemrograman seperti Java, C++, dan PHP.

2.5 IntelliJ IDEA

IntelliJ IDEA adalah aplikasi *Integrated Development Environment* (IDE) untuk pengembangan *software* yang berbasis Java. IntelliJ IDEA memiliki fitur yang memudahkan pengembangan *software*, seperti *code completion*, *refactoring*, dan *debugging*. Aplikasi ini juga memiliki fitur-fitur canggih seperti analisis kode, dukungan untuk banyak bahasa pemrograman (seperti Java, Kotlin, Groovy, dan lain-lain), integrasi dengan *framework* dan library populer (seperti Spring, Hibernate, dan lain-lain), serta banyak fitur lain yang membantu pengembang untuk menciptakan *software* dengan efisiensi dan kualitas yang tinggi. Dalam hal penyimpanan proyek, IntelliJ IDEA memiliki integrasi dengan banyak sistem pengontrol versi (seperti Git, Subversion, dan lain-lain), sehingga pengembang dapat bekerja dengan mudah dalam tim.

2.6 MySql Workbench

MySQL Workbench adalah aplikasi GUI (antarmuka pengguna grafis) yang membantu para pengguna untuk mengelola dan mengendalikan basis data MySQL. Ini memungkinkan pengguna untuk melakukan tugas-tugas seperti membuat dan mengubah tabel, menjalankan query SQL, dan membuat diagram ER (Entitas-Hubungan) dari basis data. Alat desain visual yang intuitif membuatnya mudah bagi para pengguna untuk memvisualisasikan dan memahami struktur basis data. Selain itu, MySQL Workbench juga menyediakan alat-alat manajemen yang berguna untuk memantau kinerja dan memperbaiki masalah dalam basis data. Dalam hal pengembangan aplikasi, MySQL Workbench memungkinkan para pengembang untuk membuat dan mengeksekusi skrip SQL dengan mudah. Secara keseluruhan, MySQL Workbench adalah alat yang sangat berguna bagi para administrator basis data dan pengembang aplikasi yang bekerja dengan basis data MySQL.

2.7 Postman

Postman adalah aplikasi alat pengembangan API yang memungkinkan pengguna untuk mengirim permintaan HTTP dan menangani respon yang diterima. Ini sangat berguna bagi pengembang yang ingin menguji API dan memastikan bahwa berfungsi dengan benar. Dengan menggunakan Postman, pengembang dapat mengirim permintaan GET, POST, PUT,

DELETE, dan metode lainnya ke API serta melihat respon yang diterima. Pengembang juga dapat menambahkan parameter, header, dan payload ke permintaan/*request*, serta membuat dan menyimpan koleksi permintaan/*request* untuk digunakan kembali.

2.8 Tinjauan Literatur

Dari penelitian yang sedang dilakukan, maka penelitian terdahulu sangat penting agar dapat diketahui hubungan antara penelitian yang dilakukan sebelumnya dengan penelitian yang dilakukan saat ini sebagai perbandingan dan juga landasan teori yang digunakan serta sebagai tolak ukur dalam kajian penelitian. Adapun hasil penelitian yang dijadikan acuan tidak terlepas dari topik penelitian mengenai implementasi REST API menggunakan arsitektur *microservices*.

- [1] Penelitian yang dilakukan oleh Andre Putra (2020) dengan judul penelitian tentang *“RESTful API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices”*. Penelitian ini membahas mengenai tujuan untuk membangun sebuah sistem yang lebih dinamis dan berkembang dengan memanfaatkan teknologi Arsitektur Microservice. Dibuat dengan Bahasa pemrograman Python, menggunakan database PostgreSQL.
- [2] Penelitian yang dilakukan oleh Muhammad Iqbal R (2020) dengan judul penelitian tentang *“Implementasi Restfull Api Menggunakan Arsitektur Microservices Untuk Pengelolaan Alat Mapala”* yang membahas mengenai implementasi dari arsitektur microservice untuk membantu ukm mapala dalam mengelola alat yang ada di ukm, mengetahui barang tersedia atau tidak, dan mempermudah setiap divisi dalam pemakaian alat. Penelitian ini dibuat dengan bahasa pemrograman PHP.
- [3] Penelitian yang dilakukan oleh Y.Yohakim Marwanta, dan Badiyanto (2019) dengan judul penelitian *“Implementasi Arsitektur Microservice Pada Pembuatan Surat Unit Kegiatan Mahasiswa Informatika Dan Komputer Menggunakan Node Js”* yang membahas mengenai penerapan arsitektur Microservice pada aplikasi Pembuatan Surat Unit Kegiatan Mahasiswa Informatika dan Komputer dengan menggunakan Node.js sebagai sistem backend. Hasil dari penelitian ini menghasilkan sebuah sistem aplikasi yang lebih flexible baik dalam pengembangan atau perawatan karena penerapan arsitektur Microservice yang dapat memisahkan atau membagi suatu sistem yang besar menjadi sistem-sistem kecil yang disesuaikan dengan fitur dan fungsinya. Penelitian ini menggunakan bahasa pemrograman Javascript yang memakai framework ExpressJs.

Tabel 2.1 Tinjauan Studi Terdahulu Yang Relevan

No	Nama dan Tahun Penelitian	Judul dan Lokasi Penelitian	Bahasa Pemrograman	Teknologi	Hasil Penelitian
1.	Andre Putra (2020)	RESTful API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices.	Python	Microservice, RESTful API	Pada penelitian ini menunjukkan hasil untuk membangun sebuah sistem yang lebih dinamis dan berkembang dengan memanfaatkan teknologi Arsitektur <i>Microservice</i> .
2.	Muhammad Iqbal R (2020)	Implementasi Restful Api Menggunakan Arsitektur Microservices Untuk Pengelolaan Alat Mapala (Studi Kasus: UKM Mapala STMIK AKAKOM)	PHP	Microservice, RESTful API	Penelitian yang dilakukan bahwa peneliti dapat membantu ukm mapala dalam mengelola alat yang ada di ukm, mengetahui barang tersedia atau tidak, dan mempermudah setiap divisi dalam pemakaian alat.
3.	Y.Yohakim Marwanta, dan Badiyanto (2019)	Implementasi Arsitektur Microservice Pada Pembuatan Surat Unit Kegiatan Mahasiswa Informatika Dan Komputer	Javascript	Node Js, Microservice, REST API	Hasil dari penelitian ini menghasilkan sebuah sistem aplikasi yang lebih flexible baik dalam pengembangan atau perawatan

		Menggunakan Node.js			karena penerapan arsitektur Microservice yang dapat memisahkan atau membagi suatu sistem yang besar menjadi sistem-sistem kecil yang disesuaikan dengan fitur dan fungsinya.
--	--	---------------------	--	--	--

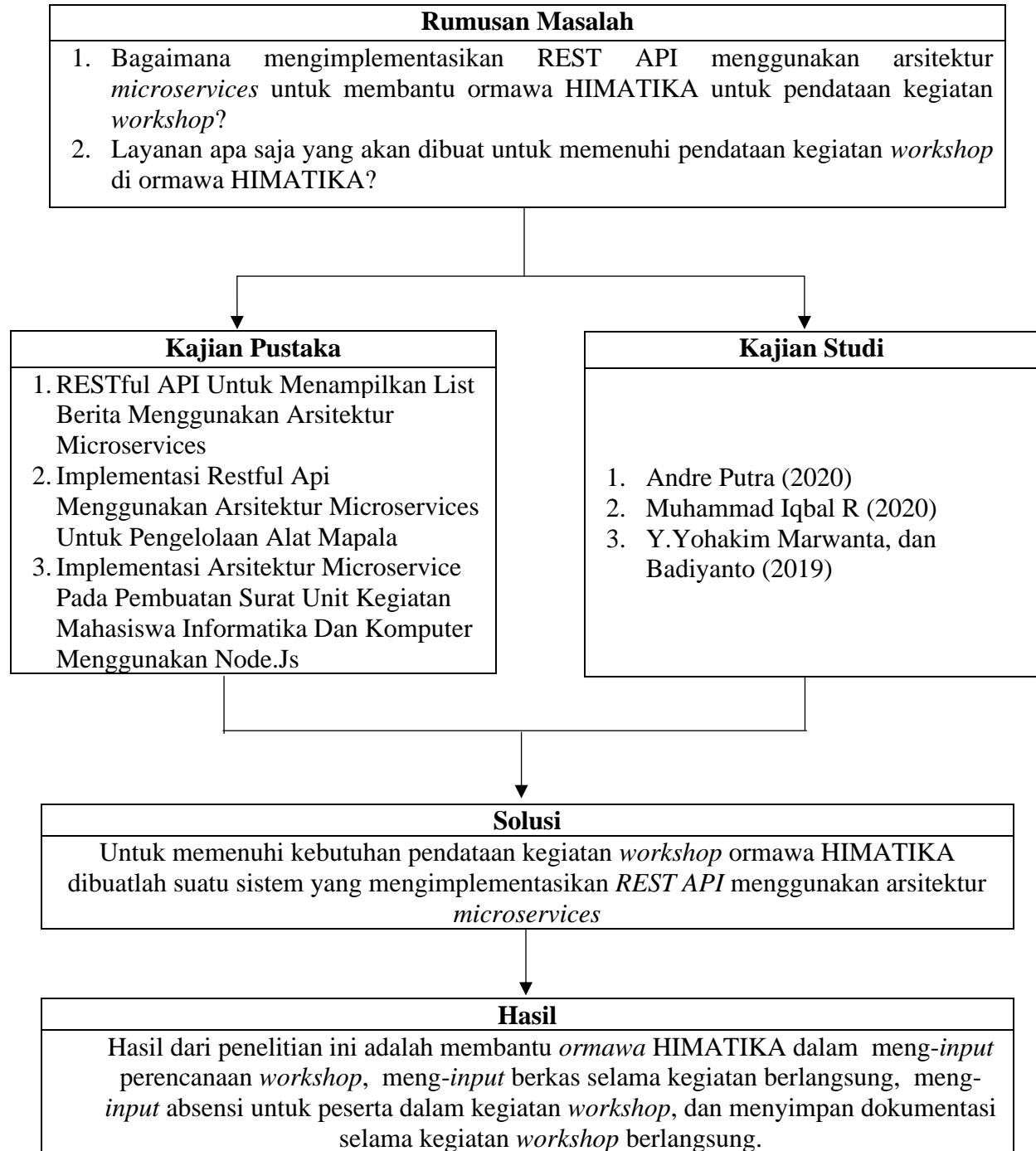
Perbandingan penelitian ini dengan penelitian terdahulunya adalah sebagai berikut:

- [1] Penelitian yang dilakukan oleh Andre Putra (2020) melakukan penelitian implementasi REST API untuk menampilkan list berita menggunakan arsitektur microservice yang menggunakan bahasa pemrograman python. Sedangkan peneliti melakukan penelitian implementasi REST API terhadap pendataan kegiatan workshop yang menggunakan framewok springboot.
- [2] Penelitian yang dilakukan oleh Muhammad Iqbal R (2020) membahas mengenai implementasi REST API menggunakan arsitektur microservice untuk pengelolaan alat mapalayang menggunakan bahasa pemrograman PHP. Sedangkan peneliti melakukan penelitian implementasi REST API terhadap pendataan kegiatan workshop yang menggunakan framewok springboot.
- [3] Penelitian yang dilakukan oleh Y.Yohakim Marwanta, dan Badiyanto (2019) melakukan penelitian implementasi REST API menggunakan arsitektur microservice pada pembuatan surat unit kegiatan mahasiswa informatika dan komputer yang menggunakan bahasa pemrograman javascript dengan framework node.js. Sedangkan peneliti melakukan penelitian implementasi REST API terhadap pendataan kegiatan workshop yang menggunakan framewok springboot.

2.9 Kerangka Pemikiran

Kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting (Sugiyono, 2017:60). Penelitian ini dibuat untuk memenuhi kebutuhan suatu masalah yang dialami oleh ormawa HIMATIKA untuk pendataan dalam menjalankan kegiatan *workshop* dengan mengimplementasikan REST API menggunakan arsitektur *microservices* yang melalui tahap awal untuk melakukan perancangan fitur atau layanan apa saja yang akan diimplementasikan. Kemudian, melakukan tahap *development* untuk melakukan *coding*, selanjutnya masuk ke proses *staging* untuk tahap testing, dan pada tahap terakhir melakukan *deployment* atau *production*. Sehingga sistem *microservices* yang dibuat akan berjalan dengan lancar dan semestinya.

2.10 Kerangka Konsep



BAB III

METODE PENELITIAN

3.1 Data

Dalam penelitian ini sampel data yang digunakan adalah data-data *workshop* yang sudah pernah dilakukan kemudian dianalisis serta disusun agar dibuat suatu sistem atau *service* yang sesuai dengan fungsinya. Kemudian dari *service* tersebut akan dibuat menggunakan arsitektur *microservices* dengan pemrograman bahasa *java* yang memakai *framework springboot*. Setelah itu, *service-service* tersebut akan diuji sistemnya apakah ada *error* atau tidak, apabila pembuatan sistem tersebut telah membuahkan hasil maka hasil tersebut akan diimplementasikan serta suatu saat dapat dikembangkan menjadi aplikasi atau *web service*.

3.2 Metodologi Penelitian

Penelitian ini menggunakan metode *Software Development Live Cycle* (SDLC). Metode SDLC sering digunakan untuk membangun atau merancang sebuah aplikasi/*software* karena memiliki tahapan yang jelas serta fleksibel dari tahapan awal hingga tahapan akhir, tahapannya terdiri dari rencana(*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*). Penelitian ini menggunakan pendekatan kualitatif. Penelitian kualitatif adalah metode penelitian yang berlandaskan pada filsafat *postpositivisme*, digunakan untuk meneliti pada kondisi obyek yang alamiah, (sebagai lawannya adalah eksperimen) dimana peneliti adalah sebagai *instrument* kunci, teknik pengumpulan data dilakukan secara *trianggulasi* (gabungan), analisis data bersifat induktif/kualitatif, dan hasil penelitian kualitatif lebih menekankan makna dari *generalisasi* (Sugiyono, 2009). Alasan saya menggunakan pendekatan kualitatif dikarenakan penelitian ini melakukan proses pengumpulan data terhadap suatu objek yang alamiah dan mendalam terhadap pendataan pada kegiatan *workshop* HIMATIKA. Data yang diperoleh nantinya akan disusun secara kompleks kemudian dilakukan proses *development* yang diimplementasikan kedalam *REST API* menggunakan arsitektur *microservices*, setelah itu *codingan script* yang sudah jadi akan masuk ke tahap *staging* untuk melakukan *testing* supaya dapat berjalan dengan lancar. Selanjutnya pada tahap akhir akan melakukan proses *deployment* agar sistem tersebut dapat dikembangkan menjadi suatu aplikasi atau *web service*.

3.3 Metode Pengumpulan Data

Pengumpulan data dilakukan guna memperoleh informasi dan data yang berhubungan dengan penelitian ini. Dalam melakukan pengumpulan data dan informasi tersebut, dilakukan metode pengumpulan data primer maupun sekunder.

1. Data Primer

Data primer dari penelitian ini adalah data *workshop* ormawa HIMATIKA yang sudah berjalan serta diimplementasikan dengan REST API menggunakan arsitektur *microservices*, sehingga sistem tersebut dapat menyelesaikan permasalahan yang dialami oleh divisi tersebut.

2. Studi Pustaka

Peneliti menggali data dan informasi melalui internet, jurnal, *ebook* yang ada kaitannya dengan penelitian yang dilakukan.

3.4 Instrumen Penelitian

Adapun instrumen penelitian yang digunakan dalam penelitian yaitu:

1. *Hardware*

Hardware yang digunakan untuk mengembangkan dan mengumpulkan data pada pembuatan sistem ini adalah sebagai berikut:

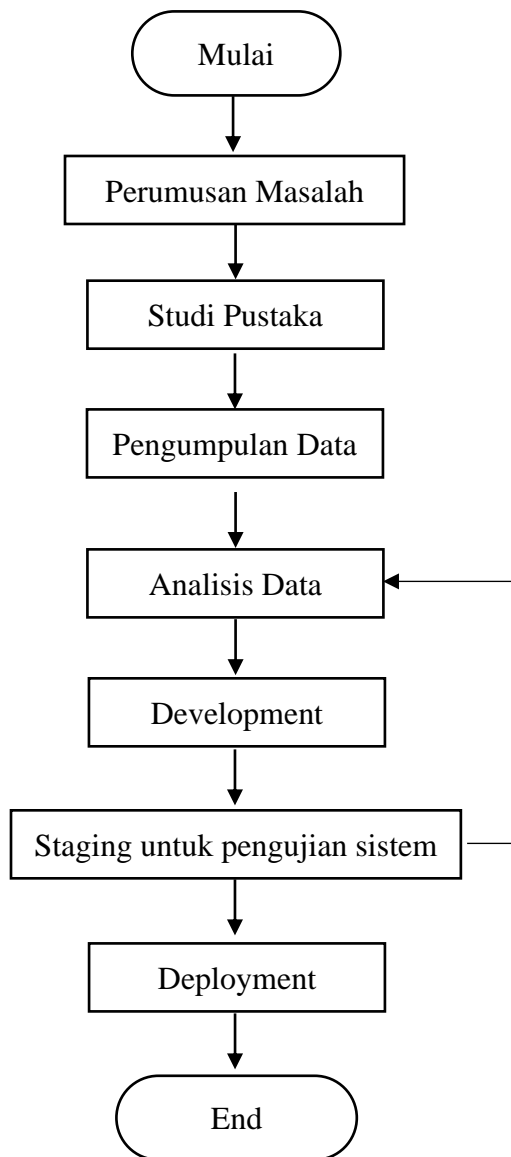
1. Laptop

2. *Software*

Adapun *software* yang digunakan pada pembuatan sistem ini adalah sebagai berikut:

1. Sistem Operasi Windows 11
2. StarUML
3. Aplikasi IntelliJ IDE
4. Mysql Workbench
5. Postman

3.5 Langkah Penelitian



Keterangan:

1. Perumusan Masalah

Pada tahap ini, peneliti merumuskan masalah yang terjadi di HIMATIKA khususnya pada divisi pengembangan sumber daya manusia untuk mengembangkan suatu sistem pendataan kegiatan workshop.

2. Studi Pustaka

Pada tahap ini dilakukan pengumpulan jurnal-jurnal yang berhubungan dengan penelitian yang dilakukan. Studi pustaka meliputi Implementasi REST API dengan arsitektur microservice dan jurnal-jurnal lain yang masih berhubungan dengan judul yang diambil oleh peneliti.

3. Pengumpulan Data

Pada tahap ini, peneliti melakukan pengumpulan data-data yang akan digunakan untuk melakukan analisa. Data yang diambil ini berasal dari kegiatan workshop yang sudah berjalan.

4. Analisis Data

Tahap ini peneliti melakukan analisis data yang sudah dikumpulkan untuk dilakukan filtering serta mengelompokkan sesuai dengan ekstensi file.

5. Development

Kemudian pada tahap development, peneliti melakukan perancangan ERD, perancangan sistem, dan proses *coding*. Tahap ini berkaitan dengan *programming*, *database*, dan integrasi antara *database* dan *source code*.

6. Staging Untuk Pengujian Sistem

Pada tahap ini, peneliti melakukan testing sistem yang sudah dibangun untuk melihat apakah ada yang error atau tidak. Pengujian sistem ini dilakukan menggunakan aplikasi *Postman* yang dapat melihat respon ketika mengirim atau meminta request dapat berjalan sesuai dengan fungsinya.

7. Deployment

Pada tahap terakhir, peneliti melakukan deployment karena sistem sudah lolos tahap testing dan siap digunakan. Proses ini hanya masih dapat dijalankan pada server local.