# BAB I

# PENDAHULUAN

# Latar Belakang Masalah

Mahasiswa Pecinta Alam atau yang biasa disebut mapala merupakan organisasi mahasiswa yang bergerak di bidang *outdoor* dikarenakan organisasi tersebut dilakukan di alam terbuka. Sebagai bagian dari organisasi pecinta alam, seorang mahasiswa tersebut harus dapat beradaptasi dengan alam masyarakat dengan menanamkan kepedulian terhadap pelestarian lingkungan serta memiliki kontribusi terhadap masyarakat. Umumnya kegiatan yang dilakukan oleh seorang mapala adalah melakukan pendakian gunung (*mountaineering*), pemanjatan *(climbing*), penelusuran gua (*caving*), pengarungan arus liar (*rafting*), penghijauan, dan Selam (*diving*). Kegiatan-kegiatan yang dilakukan oleh seorang mapala tentunya membutuhkan banyak alat-alat. Sehubungan dengan berkembangnya teknologi informasi dan banyaknya alat-alat kebutuhan kegiatan mapala yang cukup banyak, penulis membuat sistem pendataan dan peminjaman alat mapala yang berguna untuk melakukan manajemen atau pendataan alat yang masuk dan keluar.

API (Application Programming Interface) adalah sebuah mekanisme yang memungkinkan dua atau lebih aplikasi untuk berkomunikasi dan berbagi data. API telah ada sejak lama dalam bentuk yang sederhana, tetapi pada tahun 2000-an, dengan berkembangnya teknologi web, API menjadi semakin populer. API pada awalnya digunakan untuk menghubungkan aplikasi desktop dan server, tetapi dengan munculnya web services dan teknologi web modern seperti RESTful API, API telah menjadi bagian penting dari teknologi web. Penggunaan API telah menyebar ke berbagai industri, termasuk e-commerce, keuangan, dan transportasi. Salah satu faktor yang mempengaruhi perkembangan API adalah meningkatnya kebutuhan akan integrasi dan interkoneksi antara aplikasi. Dengan munculnya sistem yang lebih kompleks dan terdistribusi, integrasi antara aplikasi menjadi semakin penting. API memberikan cara yang mudah dan terstandarisasi untuk berkomunikasi antara aplikasi yang berbeda.

*REST API* adalah standar yang digunakan untuk mengirimkan dan menerima data melalui jaringan internet, yang memastikan bahwa data dapat diterima dan dikirim dengan aman dan efisien. Implementasi *REST API* menggunakan arsitektur microservices akan memastikan bahwa sistem peminjaman alat Mapala memiliki performa yang baik dan data yang diterima dan dikirim aman dan efisien. Ini juga memungkinkan untuk meningkatkan fleksibilitas sistem dan mempermudah pemeliharaan aplikasi.

Microservices adalah sebuah metode pengembangan sistem yang memiliki skala, kompleksitas, serta transfer rate yang besar dimana sebuah sistemnya dibangun dalam satu kode yang sama yang sistemnya dibagi menjadi beberapa bagian kecil. Desain arsitektur ini memberikan banyak kemudahan, salah satunya lebih efektif sehingga dapat mempermudah pengguna saat penggunaannya. Fitur-fitur yang dimiliki *microservices* sangat beragam dan banyak komponen di dalamnya. Sistem yang akan dibuat penulis membutuhkan *REST API* untuk menghubungkan setiap unit pada aplikasi/website.

Arsitektur microservices lebih mudah untuk dilakukan pengembangan dan pemeliharaan sistem. Selain itu, metode ini mengurangi risiko terjadinya kegagalan saat melakukan tahapan *deployment* versi baru dari sebuah sistem serta pada setiap layanan dapat dipantau dan diperbarui secara mandiri. Kemudian, penulis membuat “IMPLEMENTASI REST API MENGGUNAKAN ARSITEKTUR MICROSERVICES UNTUK SISTEM PENDATAAN DAN PEMINJAMAN ALAT MAPALA”*,* service-servicenya dibangun secara terpisah dengan menggunakan *RESTFULL API* sebagai komunikasi antara frontend dengan backend

# Identifikasi Masalah

Identifikasi masalah dalam penelitian ini adalah sebagai berikut :

1. Belum ada sistem yang membantu mempermudah pendataan dan peminjaman di mapala perbanas institute
2. Data alat-alat mapala sering hilang dikarenakan penyimpanannya masih manual

# Batasan Masalah

Ruang lingkup pembahasan penulisan ini akan dibatasi agar dalam pengerjaan tugas akhir dapat lebih jelas dan tidak melenceng kemana-mana, untuk itu batasan masalahnya adalah sebagai berikut :

1. Sistem microservice hanya berjalan di local saja tidak sampai tahap deployment dan hanya memiliki dua fitur yaitu pendataan dan peminjaman
2. Sistem akan mengelola alat yang tersedia untuk digunakan setiap divisi, seperti alat untuk panjat tebing, alat gunung dan alat selam

# Rumusan Masalah

Berdasarkan permasalahan yang dipaparkan pada latar belakang maka dirumuskan beberapa masalah yang dihadapi yaitu:

1. mengimplementasikan arsitektur microservices dalam pendataan dan peminjaman alat yang ada di mapala
2. mengetahui barang tersedia atau tidak, dan mempermudah setiap divisi dalam pemakaian alat

# Tujuan Penelitian

Tujuan dari penelitian ini adalah menjawab berbagai masalah yang telah penulis uraikan pada perumusan masalah, yaitu :

1. Dapat membuat sistem dengan konsep microservice.
2. Mengimplikasikan arsitektur microservice menggunakan restfull api untuk pendataan dan peminjaman alat ukm mapala.

# Manfaat Penelitian

Dari tujuan diatas, terdapat beberapa manfaat yang di dapat dari penelitian ini, antara lain:

1. Untuk membantu ukm mapala impeesa dalam pendataan dan peminjaman alat dan Dapat mempermudahkan setiap divisi dalam pendataan dan peminjaman alat
2. Developer dapat dengan mudah mengembangkan setiap service tanpa mengganggu service yang lain dan meminimalisir terjadi error
3. Tidak ada hambatan untuk menambah fitur baru

# Sistematika Penulisan

Secara keseluruhan, sistematika penulisan pada penulisan ini adalah sebagai berikut:

**BAB I : PENDAHULUAN**

Bab ini berisikan latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan dan manfaat, dan sistematika penulisan yang dipakai.

**BAB II : Tinjauan Literatur**

Bab ini berisikan berbagai jurnal-jurnal terdahulu dan teori yang berhubungan dengan sistem yang akan penulis buat. Selain itu, bab ini memuat alat yang digunakan beserta platform yang digunakan untuk penulisan ini.

**BAB III : METODOLOGI PENELITIAN**

Bab ini membahas analisa dan metode-metode yang dilakukan dalam sistem pendataan dan peminjaman beserta tahapan yang ada pada metode pengembangan scrum.

**BAB IV : HASIL PENELITIAN DAN PEMBAHASAN**

Bab ini berisikan uraian tentang hasil implementasi sistem pemanfaatan *rest api* dan *microservice* dalam sistem pendataan dan peminjaman dan pengujian fungsi

**BAB V : PENUTUP**

Bab ini berisikan kesimpulan dari hasil penulisan yang dilakukan dan saran-saran

terhadap kekurangan dari penulisan tersebut.

# BAB II

# LANDASAN TEORI

# 2.1 Tinjauan Pustaka

## 2.1.1 Arsitektur Perangkat Lunak

Arsitektur perangkat lunak dari sebuah program atau sistem komputasi adalah struktur atau struktur dari sistem, yang terdiri dari unsur-unsur perangkat lunak, sifat-sifat yang terlihat secara eksternal dari elemen-elemen, dan hubungan di antara mereka (Len Bass. 2003). Arsitektur berfungsi sebagai rencana atau desain sistem yang memberikan struktur untuk mengelola kompleksitas dan membangun mekanisme komunikasi. Arsitektur ini mendefinisikan solusi terstruktur untuk memenuhi persyaratan teknis dan operasional sambil meningkatkan kinerja dan keamanan. Setiap aplikasi memiliki pola arsitektur yang berbeda, yang membantu menentukan karakteristik dasar yang membedakan aplikasi tersebut dari yang lain, seperti *Layered Architecture, Event-Driven Architecture, Monolithic Architecture, Microservices Architecture, dan spacebased architecture.*

## Jenis – Jenis Arsitektur Perangkat Lunak

* + - 1. *Layered* *Architecture*

Arsitektur berlapis juga dikenal dengan sebutan pola arsitektur layer, di mana seluruh komponen dalam aplikasi diatur secara horizontal dan setiap lapisan memiliki peran tertentu dalam menjalankan aplikasi. Kelebihan arsitektur ini adalah kemampuannya untuk memisahkan kepentingan antar komponen yang berada di lapisan yang berbeda. Jenis arsitektur berlapis yang merupakan metode tradisional dalam merancang perangkat lunak. Tujuannya adalah untuk mengembangkan setiap komponen secara mandiri sehingga saling berhubungan, namun tidak saling bergantung.

* + - 1. *Event-Driven Architecture*

Arsitektur berbasis acara relatif kompleks dan sulit untuk diterapkan karena sifatnya yang didistribusikan secara tidak sinkron. Untuk menerapkan arsitektur ini, harus mempertimbangkan ketersediaan proses jarak jauh, kurangnya respon, dan koneksi ulang ketika terjadi kegagalan. Salah satu kesulitan penerapan arsitektur ini adalah penciptaan awal arsitektur, pemeliharaan, dan pengelolaan komponen.

* + - 1. *Monolithic Architecture*

Aristektur monolitik adalah model pemrograman tradisional, yang berarti bahwa elemen program perangkat lunak terjalin dan saling bergantung satusamalain.(Margaret Rouse, 2016)

Arsitektur monolitik ini menggunkan kode sumber dan teknologi yang serupa untuk menjalankan semua tugas-tugasnya.(Boyke Dian Triwahyudi, 2016)

* + - 1. *Microservices Architecture*

Konsep dari pola ini adalah unit yang dikelola secara terpisah, yang mana setiap komponen arsitektur digunakan sebagai unit yang terpisah dan memungkinkan untuk penyebaran yang lebih mudah, meningkatkan skalabilitas, dan tingkat aplikasi yang tinggi

* + - 1. *spacebased architecture*

arsitektur berbasis ruang (terkadang disebut sebagai cloud architecture pattern atau pola arsitektur awan) dirancang khusus untuk mengatasi dan memecahkan masalah skalabilitas yang ekstrem dan konkurensi. Pola ini juga berguna untuk aplikasi yang volume penggunanya tidak dapat diprediksi. Pola ini dinamakan berdasar pada konsep tuple space dimana menggunakan shared memory yang terdistribusi

## Api

API (Application Programming Interface) merupakan kumpulan perintah, fungsi, kelas, dan protokol yang memungkinkan satu perangkat lunak untuk berinteraksi dengan perangkat lunak lain. Tujuannya adalah untuk mempermudah para programmer dalam menggunakan ulang infrastruktur dari sebuah perangkat lunak tanpa harus menghabiskan waktu untuk menulis dan membuat ulang infrastruktur tersebut.

## Rest Api

Rest Api adalah layanan web yang diterapkan dengan menggunakan HTTP dan prinsip REST (Representational State Transfer), dan sumber datanya tersimpan dalam basis data. Layanan ini digunakan sebagai media untuk bertukar data antara klien dan server yang menyimpan basis data. Klien dapat mengirimkan permintaan dan server akan memprosesnya, seperti membuat, menerima, mengubah, atau menghapus sumber daya. Setelah server selesai memproses permintaan, server akan mengirimkan respons ke klien sebagai hasil dari aksi yang selesai. Format data yang diterima dapat berupa xml atau json, kedua format ini sangat umum digunakan dalam pertukaran data dan didukung oleh banyak bahasa pemrograman.

# Pemrograman Web

Menurut Abdulloh (2018:2) “Pemrograman web dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis web menggunakan bahasa pemrograman tertentu sehingga dapat memproses data dan menghasilkan informasi sesuai yang dikehendaki oleh pemilik website”.

## 2.2.1 Bahasa Pemrograman

Bahasa pemrograman web terdiri dari beberapa unsur bahasa. penulis menggunakan 1 bahasa utama yaitu Golang. Bahasa pemrograman Go adalah proyek open source untuk membuat program lebih produktif. Go itu ekspresif, ringkas, bersih, dan efisien. Mekanisme konkurensinya memudahkan untuk menulis program yang mendapatkan hasil maksimal dari mesin multi core dan jaringan, sementara sistem tipe barunya memungkinkan konstruksi program yang fleksibel dan modular. Go mengkompilasi dengan cepat ke kode mesin namun memiliki kemudahan pengumpulan sampah dan kekuatan refleksi run-time. Ini adalah bahasa yang dikompilasi dengan cepat, diketik secara statis, yang terasa seperti bahasa yang diinterpretasikan dan diketik secara dinamis.

## Basis Data

Basis data kumpulan data yang tersimpan didalam komputer yang digunakan suatu program untuk memperoleh informasi dari basis data tersebut. Menurut Abdulloh (2018:103) “Database atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Contoh aplikasi basis data yang penulis gunakan adalah sebagai berikut :

* + - * 1. Mysql

MySQL adalah multiuser database yang menggunakan bahasa structured query language (SQL). MySQL merupakan software yang tergolong sebagai DBMS (Database Management System) yang bersifat open source(Agustini, 2017

## JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan. Sangat mudah bagi manusia untuk membaca dan menulis. Mudah bagi mesin untuk mengurai dan menghasilkan. Ini didasarkan pada subset dari Standar Bahasa Pemrograman JavaScript ECMA-262 Edisi ke-3 - Desember 1999. JSON adalah format teks yang sepenuhnya bebas bahasa tetapi menggunakan konvensi yang akrab bagi pemrogram dari keluarga bahasa C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan banyak lainnya. Properti ini menjadikan JSON sebagai bahasa pertukaran data yang ideal.

# Tinjauan literatur

## 2.3.1 Penelitian 1

Topik penelitian “IMPLEMENTASI ARSITEKTUR MICROSERVICES PADA BACKEND COMRADES”, penulis Budi, C. S., Bachtiar, A. M., No, J. D., & Coblong, B. (2018). Dalam penelitian ini masalah yang dihadapi adalah seberapa besar pengaruh performa yang dihasilkan dengan menerapkan arsitektur *microservices* pada *web services comrades*. Solusi yang digunakan dalam penelitian ini dengan cara pengujian sistem untuk melihat dan menemukan kekurangan-kekurangan yang dimiliki oleh sistem baru yang telah diimplementasikan, sehingga dapat membantu pengembangan sistem di masa yang akan datang. Penelitian ini menggunakan teknologi *microservice,* nodejs dan *Domain Driven design.* Hasil penelitian ini fungsional-fungsional *web services* sudah berjalan sesuai dengan tugasnya masing-masing. dan arsitektur *web services* yang menggunakan *microservices* terbukti lebih unggul dibandingkan *monolithic.*

## 2.3.2 Penelitian 2

Topik penelitian “IMPLEMENTASI RESTFULL API MENGGUNAKAN ARSITEKTUR MICROSERVICES UNTUK PENGELOLAAN ALAT MAPALA (STUDI KASUS UKM MAPALA STMIK AKAKOM)”, penulis iqbal R (2020). Dalam penelitian ini masalah yang dihadapi bagaimana mengimplementasikan arsitektur *microservices* untuk membantu ukm mapala dalam mengelola alat yang ada di ukm dan mengetahui barang tersedia atau tidak. Solusi yang digunakan oleh penulis adalah menggunakan arsitektur microservice dan teknologi yang digunakan adalah *microservice* dan *restfull api.* Hasil dari penelitian yaitu berhasil membuat rest api untuk membuat sebuah aplikasi pengelolaan data alat mapala, Berhasil menampilkan data dari backend untuk ditampilkan di frontend dan berhasil menggunakan php untuk implementasi arsitektur microservices dan membuat rest api. Penulis meninggalkan saran yaitu menambahkan beberapa fitur pada aplikasi, memperbaiki UI dan menambah proses peminjaman alat.

## 2.3.3 Penelitian 3

Topik penelitian “Implementasi Arsitektur Microservice Pada Aplikasi Online Travel Tourinc”, penulis M danil Rafiqi(2019). Dalam penulisan ini masalah yang dihadapi PT. Tourinc membutuhkan aplikasi untuk membantu menjalankan proses bisnisnya. karena inti bisnis yang dijalankan adalah penjualan tiket secara online. Solusi yang tepat untuk pengembangan yaitu menggunakan arsitektur microservice karena dapat mengatasi kompleksitas yang cukup besar dan teknologi yang digunakan *Microservices* dan *RESTfull API.* Penulis juga menjelaskan tentang *monolithic* dimana banyak kekurangannya seperti: (1) performa akan menurun ketika aplikasi semakin besar, (2) sulit untuk adaptasi tekologi baru, (3) aplikasi semakin besar dan kompleks sehingga sulit dipahami. Dari penelitian diatas terlihat *microservices* lebih unggul dibandingkan *monolithic* Penulis juga meninggalkan saran yaitu Menambah service yang ada dari arsitektur microservice yang sudah dibuat sebelumnya dan Menambah fitur cache pada pengembangan selanjutnya agar dapat mempercepat pengaksesan data.

## 2.3.4 Penelitian 4

Topik penelitian “Refactoring Arsitektur Microservice Pada Aplikasi Absensi PT. Graha Usaha Teknik”, penulis Mufrizal, R., & Indarti, D. (2019). Dalam penulisan ini menjelaskan kenapa harus *refactoring* arsitektur *monolithic* menjadi arsitektur *microservice* karena berdasarkan hasil pengujian perangkat lunak menggunakan load test pada arsitektur *microservice* dan *monolithic* dapat disimpulkan bahwa arsitektur *microservice* yang telah dibangun dapat menangani beban yang telah ditentukan, yaitu sebesar 15 tps dan menggunakan spesifikasi komputer yang telah di tentukan. Penulis juga mengusulkan pada penelitian ini dapat digunakan sebagai acuan untuk melakukan proses *refactoring* pada system *monolithic* yang lebih besar seperti system ERP.

## 2.3.5 Penelitian 5

Topik penelitian “PERANCANGAN ARSITEKTUR PERANGKAT LUNAK MICROSERVICES PADA APLIKASI OPEN LIBRARY UNIVERSITAS TELKOM MENGGUNAKAN gRPC”, penulis Lombogia, J. F., Syahrina, A., & Musnansyah, A. (2022). Dalam penulisan ini melakukan migrasi perangkat lunak dimulai dengan mngengumpulkan data sampai menerapkan konsep *domain driven design.* Peneliti juga melakukan migrasi arsitektur perangkat lunak monolitik ke microsercives dengan menerapkan konsep strangler pattern. Hasil dari penelitian ini dapat meningkatkan performa dan kebebasan dalam memilih teknologi sesuai dengan masalah yang sedang dihadapi.

Dari penelitian terdahulu diatas dapat penulis simpulkan bahwa pemanfaatan *microservices* dalam pendataan dan penyimpanan dengan metode scrum sebagai hasil dan analisis implementasi mampu menyelesaikan permasalahan lebih baik dibandingkan monolitik dan mudah untuk dikembangkan.