

**IMPLEMENTASI *REST API* MENGGUNAKAN ARSITEKTUR  
*MICROSERVICES* UNTUK SISTEM PENDATAAN DAN PEMINJAMAN  
ALAT MAPALA**



**PERBANAS  
INSTITUTE**

**OUTLINE SKRIPSI**

Disusun Untuk Memenuhi Sebagian Persyaratan Guna Memperoleh Sarjana  
Komputer

Oleh :

**RINANDA SOELAIMAN NUR SAHID**

NIM : 1914000029

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 Masalah

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.

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 lain sebagainya. Kegiatan-kegiatan yang dilakukan oleh seorang mapala tentunya membutuhkan banyak alat-alat. Seperti kegiatan pendakian gunung, seorang mapala harus memiliki tas *carrier* yang diperlukan untuk membawa peralatan mendaki lainnya dan bahan makanan yang akan diperlukan pada saat kegiatan mendaki gunung. Selain itu tenda juga merupakan salah satu alat yang wajib dibawa ketika mendaki gunung yang digunakan sebagai rumah sementara atau tempat beristirahat dan meletakkan barang-barang bawaannya. 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.

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.

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.

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.

## 1.2 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

## 1.3 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

## **1.4 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

## **1.5 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.

## **1.6 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 mempermudah 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

## **1.7 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**

### **STUDI LITERATUR**

#### **2.1 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 ‘Blue Print’ untuk suatu sistem. Ini memberikan abstraksi untuk mengelola kompleksitas sistem dan membangun mekanisme komunikasi. Ini mendefinisikan solusi terstruktur untuk memenuhi semua persyaratan teknis dan operasional, sambil mengoptimalkan atribut kualitas umum seperti kinerja dan keamanan. Setiap aplikasi memiliki pola arsitektur yang berbeda, pola arsitektur membantu menentukan karakteristik dasar suatu aplikasi yang membuat sebuah aplikasi berbeda dengan aplikasi lainnya. Ada beberapa jenis pola yang sering digunakan di setiap aplikasi seperti *Layered Architecture*, *Event-Driven Architecture*, *Microkernel Architecture*, *Microservices Architecture* dan *space based architecture*.

##### **2.1.1 Jenis-Jenis Arsitektur**

###### **1. Layered Architecture**

Bisa disebut juga dengan arsitektur berlapis, Seluruh komponen yang ada dalam arsitektur berlapis dibuat ke dalam bentuk horizontal, dimana setiap lapisan melakukan peran tertentu dalam sebuah aplikasi. Salah satu kelebihan dibanding arsitektur lain adalah arsitektur berlapis memisahkan kepentingan antar komponen (yang berada di layer yang berbeda).

###### **2. 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.

### 3. *Microkernel Architecture*

Arsitektur microkernel (kadang disebut sebagai arsitektur plug-in) adalah pola untuk menerapkan aplikasi berbasis produk yang mana dikemas dan tersedia untuk diunduh dalam versi produk pihak ketiga. Arsitektur microkernel terdiri dari dua jenis komponen arsitektur, yaitu sistem inti dan plug-in module. Sistem inti dari pola arsitektur microkernel hanya berisi fungsionalitas minimal yang diperlukan untuk membuat sistem operasional.

### 4. *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.

### 5. *space based 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 penggunaanya tidak dapat diprediksi. Pola ini dinamakan berdasar pada konsep tuple space dimana menggunakan shared memory yang terdistribusi.

## **2.1.2 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.

## **2.1.3 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.

## **2.2 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.

### **2.2.2 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 sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Contoh aplikasi basis data yang penulis gunakan adalah sebagai berikut :

#### **a. 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)

### 2.2.3 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.

## 2.3 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., & Cobleng, 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 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

## 2.4 Kerangka Pemikiran

Masalah		
1.1 Data pendataan yang ada di mapala impeesa masih menggunakan manual dan gampang hilang. 2.1 Belum ada sistem pendataan dan peminjaman untuk membantu anggota mapala.		
Eksperimen/Analisa		
Tools	Data	Metode
Golang, Mysql , Postman dan Vscope	Alat-alat mapala dan data anggota mapala	Arsitektur <i>Microservices</i>
Pengujian dan Validasi Hasil		
Untuk alat-alat dan anggota mapala akan menggunakan data dummy. akan memiliki 3 service yaitu auth service, pendataan service dan peminjaman service.		
Hasil		
Sebuah sistem pendataan dan peminjaman yang berjalan di local.		

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Data

Data studi kasus menggunakan data yang berada di mapala perbanas institute. Sumber data ini merupakan data anggota mapala dan alat-alat per divisi. Dalam hal ini penelitian memperoleh data atau informasi langsung dengan menggunakan instrument-instrumen yang telah ditetapkan. sumber data dalam penelitian ini adalah data *dummy* anggota mapala dan alat-alat per divisi yang berada di mapala perbanas institute selaku pengembang untuk penelitian ini.

### 3..1.1 Metode Pengumpulan Data

Peneliti menggunakan metode dalam melakukan analisis data dengan beberapa tahap yaitu :

a. Observasi

Observasi merupakan teknik data yang dilakukan dengan cara yang ingin diteliti atau melalui eksperimen (percobaan). Teknik ini cocok dilakukan untuk penelitian yang bertujuan untuk mendapatkan kebutuhan yang akan diperlukan penelitian.

b. Studi Literatur

Studi Literatur adalah cara untuk menyelesaikan persoalan dengan menelusuri sumber-sumber tulisan yang pernah dibuat sebelumnya. Dalam sebuah penelitian yang akan dijalankan, tentunya seorang peneliti harus memiliki wawasan yang luas terkait objek yang akan diteliti. Studi Literatur juga dilakukan untuk pengumpulan informasi terkait dengan penelitian.

### 3.3 Instrumen Penelitian

Penelitian menggunakan beberapa alat bantu untuk melakukan pengembangan pada penelitian. Berikut alat bantu pengembangan yang digunakan yaitu :

1. Kebutuhan perangkat lunak

Perangkat lunak atau software yang digunakan dalam proses *development* adalah :

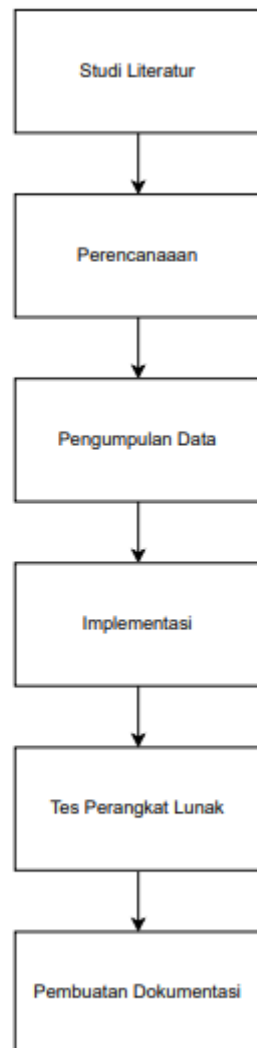
- a. ubuntu
- b. golang
- c. visual Studio (test editor)
- d. postman
- e. google chrome (Browser)

2. Kebutuhan perangkat keras

Perangkat keras atau hardware yang digunakan dalam proses penelitian adalah :

- a. Laptop dengan Prosesor AMD A9
- b. RAM 8 GB

### 3.3 Tahapan Penelitian



#### 3.3.1 Studi literatur

Tahap ini dilakukan dengan mempelajari jurnal-jurnal referensi atau sumber-sumber yang berkaitan dengan skripsi ini, baik dari text book maupun internet.

#### 3.3.2 Perancangan

Dalam tahapan ini penulis melakukan perencanaan kebutuhan yang dibutuhkan untuk software yang hendak dikerjakan

#### 3.3.2 Pengumpulan data

Tahapan ini penulis melakukan pengumpulan data dengan cara observasi

### 3.3.3 Implementasi

Tahapan selanjutnya menekankan penulis dalam melakukan pengkodean suatu software

### 3.3.4 Tes Perangkat lunak

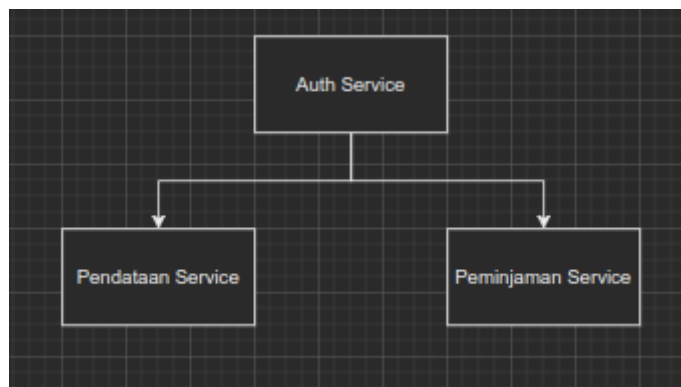
tahap ini dimaksudkan untuk pengetesan dan pengecekan perangkat lunak yang telah diproduksi agar bug yang ditemukan mampu diperbaiki segera.

### 3.3.5 Pembuatan Dokumentasi

Jika tahap pengetesan selesai, maka dilanjutkan tahap dokumentasi agar memberikan kemudahan pada proses pemeliharaan kedepannya.

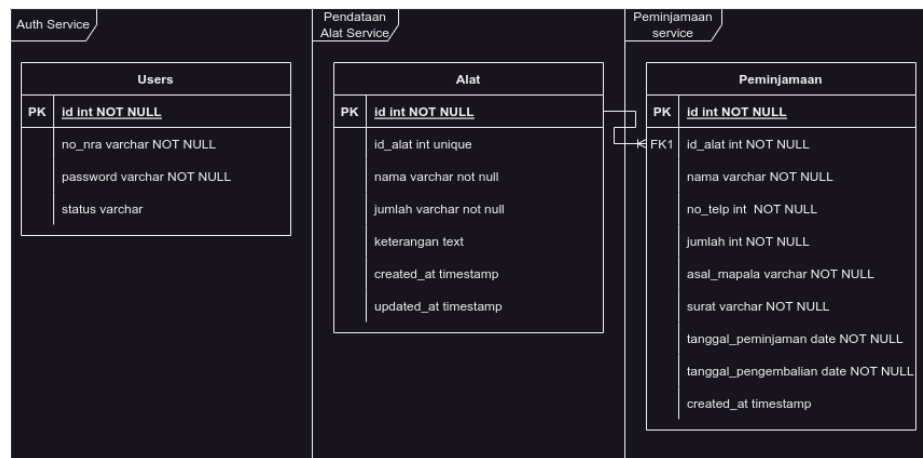
## 3.4 Penerapan Rest Api

### 3.4.1 Arsitektur Sistem



Gambar diatas adalah arsitektur sistem yang berada di penelitian ini. Yaitu ada tiga service yang akan dibuat yaitu Auth service, pendataan service dan peminjaman service. Setiap service memiliki fungsi masing-masing, untuk auth service berfungsi untuk anggota mapala agar bisa masuk ke dalam service pendataan service, pendataan service berisi crud alat-alat per divisi yang ada di mapala perbanas institute dan peminjaman service menyimpan data mapala mana saja yang akan meminjam alat yang ada di sekretariat mapala perbanas institute.

### 3.4.2 Rancangan struktur tabel



Gambar diatas adalah rancangan struktur tabel dalam penelitian ini yaitu ada tabel users, alat dan peminjaman

### 3.4.2 Penerapan Rest Api

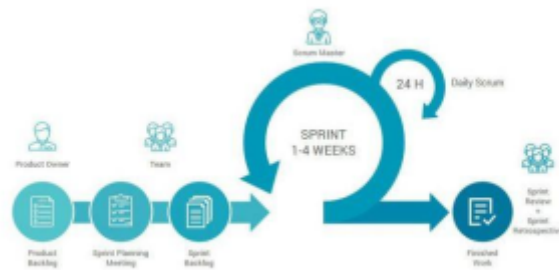
Untuk membuat implementasi REST API menggunakan arsitektur microservices untuk sistem pendataan dan peminjaman alat Mapala, beberapa hal yang perlu diperhatikan adalah:

- Desain API:** Terlebih dahulu, harus ditentukan bagaimana API akan terlihat seperti dan bagaimana interaksi antar microservice akan terjadi. Ini dapat dicapai dengan membuat dokumentasi API yang jelas dan menentukan endpoint API dan metode HTTP yang sesuai untuk setiap tugas.
- Pembagian Microservices:** Berikutnya, sistem harus dibagi menjadi beberapa microservice yang berbeda. Misalnya, satu microservice untuk mengelola data peminjaman, satu lagi untuk mengelola data alat
- Implementasi API:** Setelah desain API dan pembagian microservices selesai, maka dapat dilakukan implementasi API. Ini dapat dilakukan menggunakan bahasa pemrograman seperti Golang, Python, dll. dengan menggunakan framework seperti gin atau flask.

### 3.5 Metode yang digunakan

Metode pengembangan sistem yang dipakai dalam penelitian ini adalah metode Agile (scrum). Metode Scrum merupakan salah satu metode yang ada pada model Agile. Scrum adalah salah satu metode rekayasa perangkat lunak dengan menggunakan prinsip-prinsip pendekatan

Agile, yang bertumpu pada kekuatan kolaborasi tim, incremental product dan proses iterasi untuk mewujudkan hasil akhir.



#### 1. Product Backlog

Pada bagian ini, hal-hal yang diperlukan pada suatu produk harus tersedia. Hal-hal tersebut disebut dengan product backlog.

#### 2. Sprint Backlog

Perencanaan Sprint dilakukan dalam pertemuan antara klien dan tim developer yang akan melakukan kerja sama untuk memilih product backlog agar dimasukkan ke dalam proses sprint. Hasil dari pertemuan yang dilakukan disebut Sprint Backlog.

#### 3. Aktivitas Sprints

Aktivitas sprint merupakan kerangka waktu agar dapat mengembangkan produk sesuai dengan daftar kebutuhan. Pada Sprint terdapat dua bagian, yaitu:

- Pertemuan harian

Pertemuan harian adalah pertemuan yang dilakukan setiap hari agar tim pengembang dapat bertemu untuk membahas aktivitas proses pengembangan

- Refleksi sprint

Refleksi sprint merupakan pertemuan yang dilakukan setiap bulan dengan tujuan membahas hal dari Sprint Backlog yang berjalan dan memperbaiki beberapa fitur untuk meningkatkan kualitas produk.

#### 4. Sprint Review

Sprint Review merupakan kegiatan melihat kembali fitur yang telah dikerjakan agar memastikan bahwa fitur yang telah dikerjakan dapat bekerja dengan baik.

## 5. Sprint Retrospective

Pada Sprint Retrospective, tim melihat kembali bagaimana pekerjaan berjalan pada Sprint sebelumnya dengan harapan adanya perbaikan sehingga Sprint selanjutnya dapat dikerjakan dengan lebih baik lagi.

### Jadwal Penelitian

	Bulan 1	Bulan 2	Bulan 3	Bulan 4	Bulan 5
Activity 1	Perumusan masalah				
Activity 2		Penyusunan dan pemilihan objek			
Activity 3			Melakukan pengumpulan data penelitian		
Activity 4			Melakukan pengolahan data		
Activity 5				Membangun system	
Activity 6				Menguji System	
Activity 7					Evaluasi dan validasi hasil
Activity 8					Melanjutkan penulisan bab 5 dan 6



**BAB IV**  
**HASIL DAN PEMBAHASAN**