

**PRAKTIKUM STUDI KASUS SISTEM ASSET
MANAGEMENT MENGGUNAKAN
SPRINGBOOT (BACKEND) DAN JAVA SCRIPT
(FRONT END)**

**PEMROGRAMAN BASIS DATA AMIKOM X
PRAKTISI MENGAJAR ANGKATAN 4**



**UNIVERSITAS
AMIKOM
YOGYAKARTA**

**By: Satria Ardi Perdana S.T., M.Kom
2024**

MODUL 1 REQUIREMENT

Pada modul 1 kita akan mengumpulkan kebutuhan untuk latihan membuat Sistem Manajemen Aset. Sistem Manajemen Aset itu berguna untuk pengelolaan aset oleh sebut saja PT SGL. PT SGL memanfaatkan Sistem Manajemen Aset untuk:

1. Mencatat semua aset yang dimiliki seperti laptop, komputer, monitor, printer, mobil operasional dan yang lainnya.
2. Mencatat supplier pemasok aset PT SGL.
3. Mencatat pegawai yang melakukan atau yang mendapatkan fasilitas dari perusahaan
4. Mencatat jenis kerusakan yang terjadi.
5. Mencetak laporan / export ke file atau pdf.

System requirement:

1. Sistem operasi: Windows, Linux, macOS.

Dalam modul ini di contohkan menggunakan Linux Ubuntu 24.04 LTS.



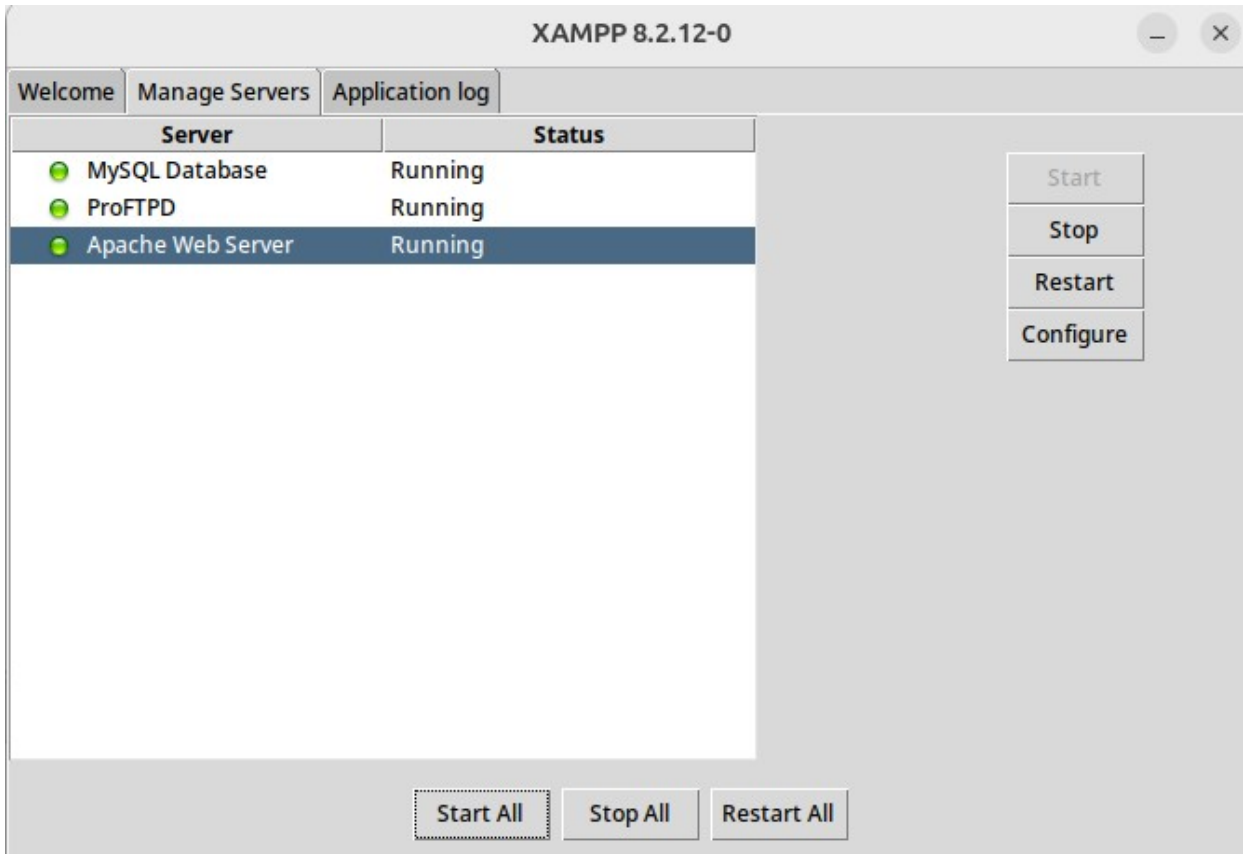
2. Java 21.

```
satria@satria-gli: ~  
satria@satria-gli:~$ java -version  
openjdk version "21.0.3" 2024-04-16  
OpenJDK Runtime Environment (build 21.0.3+9-Ubuntu-1ubuntu1)  
OpenJDK 64-Bit Server VM (build 21.0.3+9-Ubuntu-1ubuntu1, mixed mode, sharing)  
satria@satria-gli:~$
```

3. Springboot (Framework Java).
4. JPA (Java Persistence API) → mapping antara Object Java (entity) dengan database.
5. IDE → IntelliJ IDEA atau Spring Tool Suite, VS Code.
6. Swagger atau Postman → Dokumentasi dan API testing tool.
7. Maven → Dependency management.

```
satria@satria-gli: ~  
satria@satria-gli:~$ mvn -version  
Apache Maven 3.9.7 (8b094c9513efc1b9ce2d952b3b9c8eaedaf8cbf0)  
Maven home: /opt/apache-maven-3.9.7  
Java version: 21.0.3, vendor: Ubuntu, runtime: /usr/lib/jvm/java-21-openjdk-amd64  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "6.8.0-35-generic", arch: "amd64", family: "unix"  
satria@satria-gli:~$
```

8. Reporting → generate ke file.
9. Github.com → manajemen source code.
10. XAMPP



MODUL 2

MEMBUAT DATABASE

Database menggunakan MySQL/MariaDB bawaan dari XAMPP. Ikuti langkah sebagai berikut:

1. Buat database dengan nama **asset**.

```
CREATE DATABASE asset;
```

2. Gunakan database **asset**.

```
USE asset;
```

3. Buat tabel aset (tb_master_aset)

```
CREATE TABLE tb_master_aset (  
    tbma_id int primary key auto_increment,  
    tbma_nama varchar(50),  
    tbma_status boolean,  
    tbma_tbms_id int  
);
```

4. Buat tabel supplier (tb_master_supplier)

```
CREATE TABLE tb_master_supplier(  
    tbms_id int primary key auto_increment,  
    tbms_created_date datetime,  
    tbms_name varchar(50),  
    tbms_jenis_supplier varchar(50),  
    tbms_status boolean  
);
```

5. Buat tabel pegawai (tb_master_pegawai)

```
CREATE TABLE tb_master_pegawai(  
    tbmp_id int primary key auto_increment,  
    tbmp_created_date datetime,  
    tbmp_name varchar(50),  
    tbmp_alamat varchar(50),  
    tbmp_phone varchar(50),  
    tbmp_status boolean  
);
```

6. Buat tabel order (tb_transaction_order) → mencatat jika perusahaan pesan aset ke supplier

```
CREATE TABLE tb_transaction_order(  
    tbto_id int primary key auto_increment,  
    tbto_created_date datetime,  
    tbto_tbma_id int,  
    tbto_tbms_id int,  
    tbto_qty_pesan int,  
    tbto_status boolean  
);
```

7. Buat tabel pinjam (tb_transaction_pinjam) → mencatat aset yang dipinjamkan ke pegawai.

```
CREATE TABLE tb_transaction_pinjam(
    tbtp_id int primary key auto_increment,
    tbtp_created_date datetime,
    tbtp_tbma_id int,
    tbtp_tbmp_id int,
    tbtp_status boolean
);
```

8. Add foreign key pada `tb_master_aset.tbma_tbms_id`

```
ALTER TABLE tb_master_aset ADD FOREIGN KEY(tbma_tbms_id) REFERENCES
tb_master_supplier(tbms_id);
```

9. Add foreign key pada `tb_transaction_order`

```
ALTER TABLE tb_transaction_order ADD FOREIGN KEY(tbto_tbma_id) REFERENCES
tb_master_aset(tbma_id);
```

```
ALTER TABLE tb_transaction_order ADD FOREIGN KEY(tbto_tbms_id) REFERENCES
tb_master_supplier(tbms_id);
```

10. Add foreign key pada tabel `tb_transaction_pinjam`

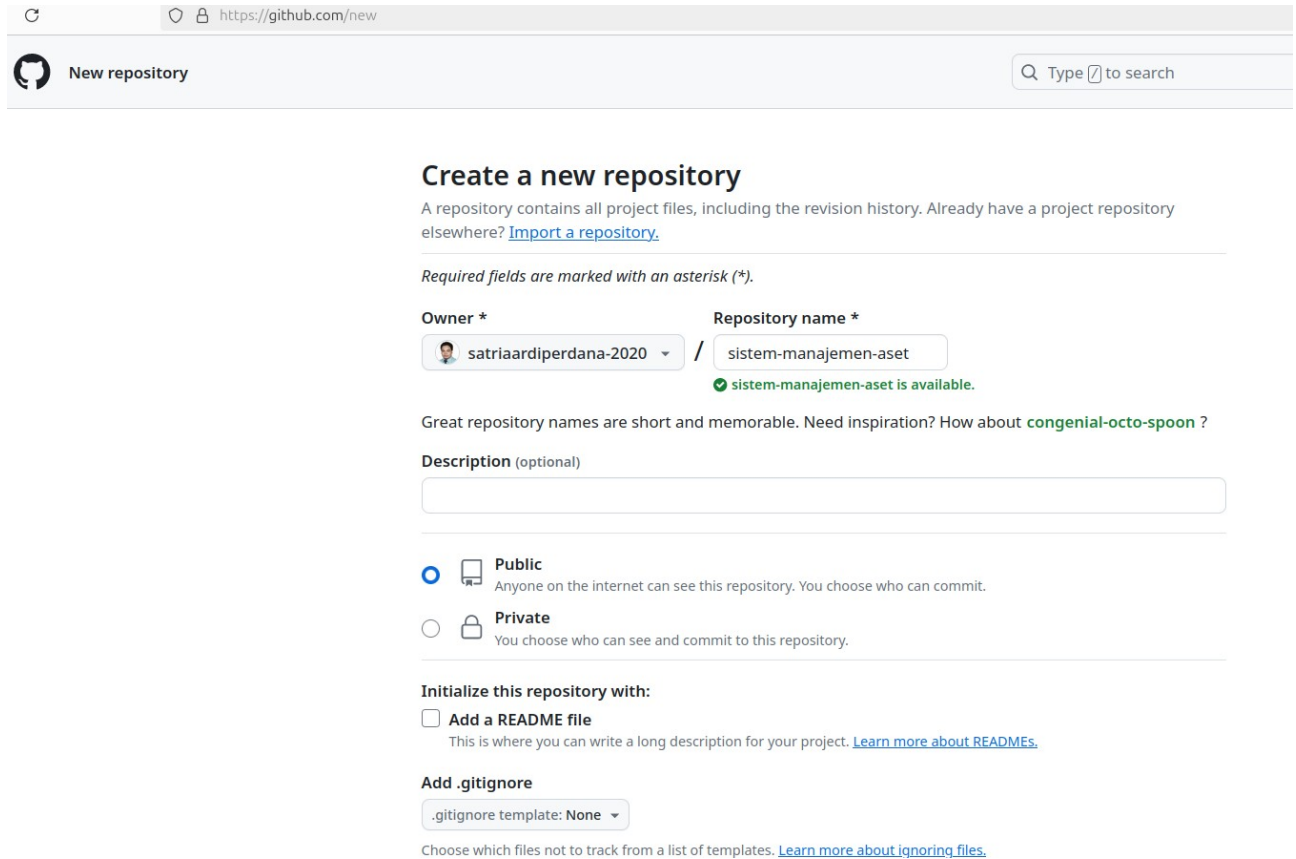
```
ALTER TABLE tb_transaction_pinjam ADD FOREIGN KEY(tbtp_tbma_id) REFERENCES
tb_master_aset(tbma_id);
```

```
ALTER TABLE tb_transaction_pinjam ADD FOREIGN KEY(tbtp_tbmp_id) REFERENCES
tb_master_pegawai(tbmp_id);
```

MODUL 3

BUAT SPRINGBOOT PROJECT

1. Buat repository dengan nama **sistem-manajemen-aset** pada github.com untuk manajemen source code.



https://github.com/new

New repository

Search Type to search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

satriaardiperdana-2020 / sistem-manajemen-aset

✓ sistem-manajemen-aset is available.

Great repository names are short and memorable. Need inspiration? How about [congenial-octo-spoon](#) ?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

2. Jika berhasil membuat repository pada github muncul seperti berikut:

https://github.com/satriaardiperdana-2020/sistem-manajemen-aset


sistem-manajemen-aset Public



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot



Add collaborators to this repository

Search for people using their GitHub username or

Invite collaborators

Quick setup — if you've done this kind of thing before

HTTPS **SSH** `https://github.com/satriaardiperdana-2020/sistem-manajemen-aset.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# sistem-manajemen-aset" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/satriaardiperdana-2020/sistem-manajemen-aset.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/satriaardiperdana-2020/sistem-manajemen-aset.git
```

3. Buat Personal access tokens di github klik Settings → Personal access tokens → Tokens(classic) → Generate new token → Generate new token (classic). Kemudian isi Note, expired akses dan centang seperti berikut:

https://github.com/settings/tokens/new

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

sistemmanajemenaset

What's this token for?

Expiration *

30 days The token will expire on Fri, Jul 19 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read:org	Read org and team membership, read org projects
<input checked="" type="checkbox"/> manage_runners:org	Manage org runners and runner groups

4. Setelah klik button Generate Token (warna hijau) akan muncul toke yang hanya muncul sekali (harus dicatat) seperti berikut:

https://github.com/settings/tokens

Settings / Developer Settings

Scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Personal access tokens (classic) Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ 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 Delete

5. Jadi saya atau teman teman bisa akses repository saya dengan password:

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

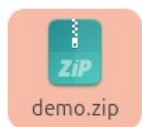
6. Buat template project dengan spring initializr <https://start.spring.io/> . Ikuti langkah berikut dan klik GENERATE:

The screenshot shows the Spring Initializr web application at <https://start.spring.io>. The interface is divided into several sections:

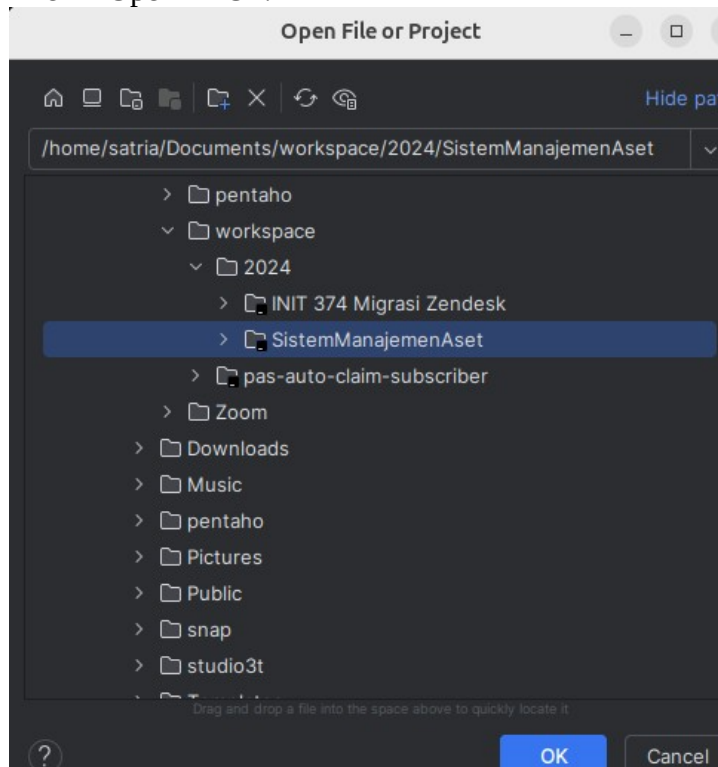
- Project:** Options for Gradle - Groovy, Gradle - Kotlin, and Maven (selected).
- Language:** Options for Java (selected), Kotlin, and Groovy.
- Spring Boot:** Options for 3.3.1 (SNAPSHOT), 3.3.0, 3.2.7 (SNAPSHOT), and 3.2.6 (selected).
- Project Metadata:**
 - Group:
 - Artifact:
 - Name:
 - Description:
 - Package name:
 - Packaging: ☒ Jar, ☐ War
 - Java: ☐ 22, ☒ 21, ☐ 17
- Dependencies:**
 - Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
 - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
 - MySQL Driver** (SQL): MySQL JDBC driver.
 - Lombok** (DEVELOPER TOOLS): Java annotation library which helps to reduce boilerplate code.

At the bottom, there are three buttons: **GENERATE** (CTRL + G), **EXPLORE** (CTRL + SPACE), and **SHARE...**

7. Akan terdownload template project sesuai nama artifactnya kemudian extract dan rename menjadi SistemManajemenAset:



8. Import kedalam IDE
File → Open → Ok:



9. Masuk ke path project (SistemManajemenAset) dan inisiasi, commit, push ke github.

Git init

```
satria@satria-gli: ~/Documents/workspace/2024/SistemManajemenAset
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git init
```

git add .

```
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git add .
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$
```

commit

```
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git commit -m "pertama kali"
[master (root-commit) bd23c76] pertama kali
 8 files changed, 563 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .mvn/wrapper/maven-wrapper.properties
 create mode 100755 mvnw
 create mode 100644 mvnw.cmd
 create mode 100644 pom.xml
 create mode 100644 src/main/java/com/praktikum/demo/DemoApplication.java
 create mode 100644 src/main/resources/application.properties
 create mode 100644 src/test/java/com/praktikum/demo/DemoApplicationTests.java
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$
```

pilih main branch

```
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git branch -M main
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$
```


remote github repository

```
satria@satria-gli: ~/Documents/workspace/2024/SistemManajemenAset
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git remote add origin https://github.com/satriaardiperdana-2020/sistem-manajemen-aset.git
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$
```


push (kirim) source code yang ada di local ke repository. Username menggunakan email atau nick github dan password menggunakan personal access token pada nomor 4.

```
satria@satria-gli: ~/Documents/workspace/2024/SistemManajemenAset
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git remote add origin https://github.com/satriaardiperdana-2020/sistem-manajemen-aset.git
satria@satria-gli:~/Documents/workspace/2024/SistemManajemenAset$ git push -u origin main
Username for 'https://github.com': satriaardiperdana-2020
Password for 'https://satriaardiperdana-2020@github.com':
```


10. Cek di repository, jika berhasil ada history source code yang masuk.







 satriaardiperdana-2020 / sistem-manajemen-aset Q Type to search


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


 **sistem-manajemen-aset** Public Pin Unwatch 1

main 1 Branch 0 Tags Go to file Add file Code

 **satria** pertama kali bd23c76 · 6 minutes ago 1 Commit

 .mvn/wrapper	pertama kali	6 minutes ago
 src	pertama kali	6 minutes ago
 .gitignore	pertama kali	6 minutes ago
 mvnw	pertama kali	6 minutes ago
 mvnw.cmd	pertama kali	6 minutes ago
 pom.xml	pertama kali	6 minutes ago

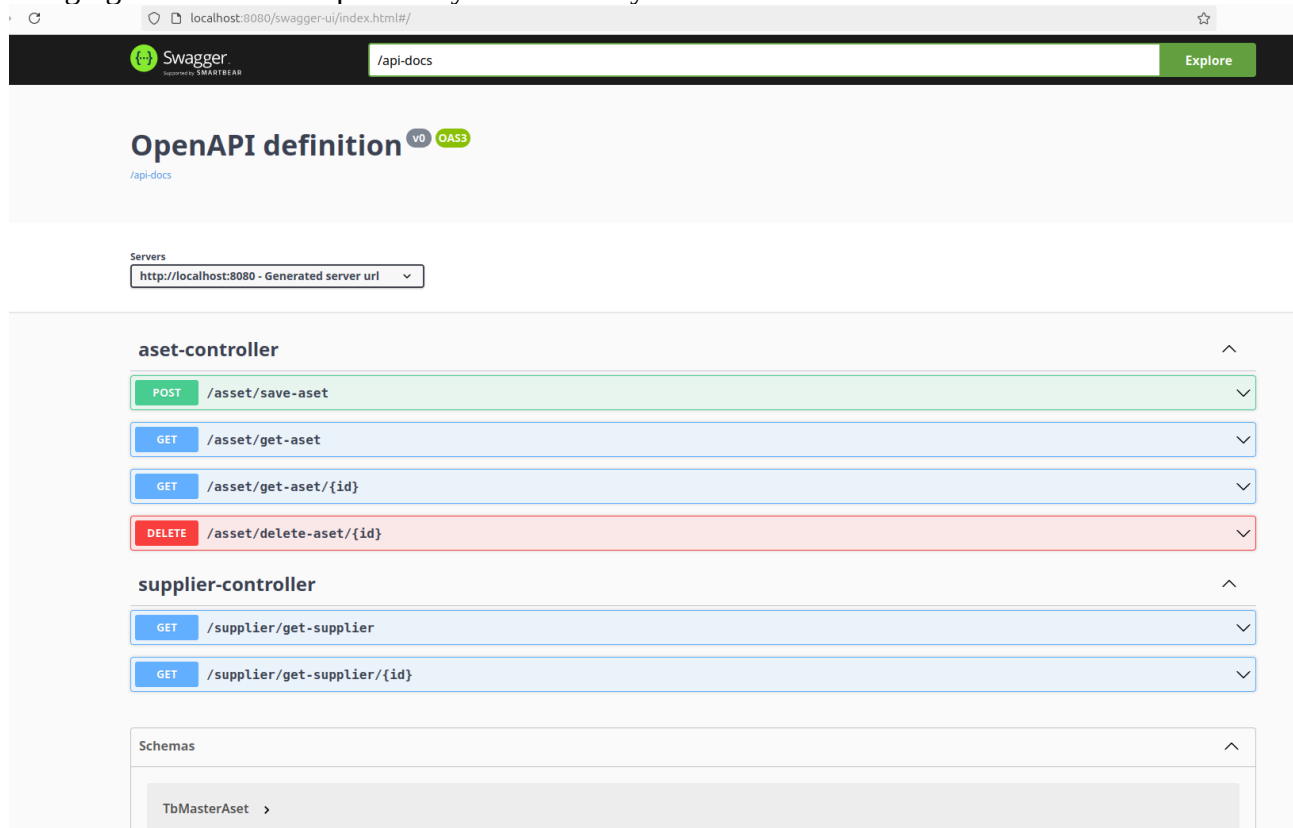
 **README**


Add a README

MODUL 4 BACKEND

Pada bagian ini akan di buat CRUD (Create, Read, Update, Delete)API menggunakan Springboot. Ikuti langkah live code.

Hasil API nya bisa didokumentasikan menggunakan swagger. Swagger berguna agar kita tidak mengingat link url dan request body dan headernya.



MODUL 5
FRONT END DAN REPORTING