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

## PEMROGRAMAN BASIS DATA AMIKOM X PRAKTISI MENGAJAR ANGKATAN 4

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

**2024**

**MODUL 1**
**REQUIREMENT**

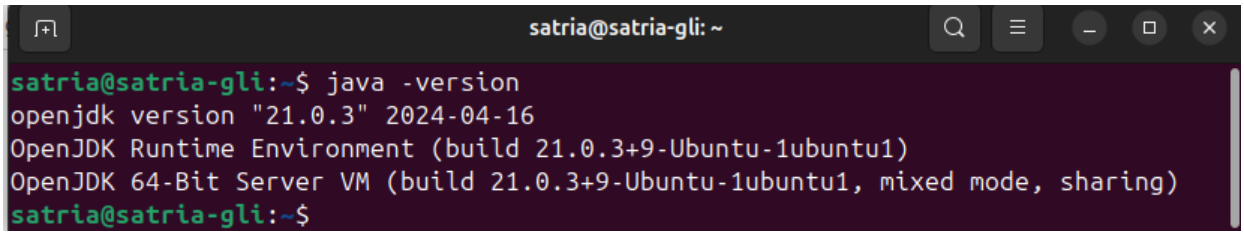
Pada modul 1 kita akan mengumpulkan kebutuhan untuk latihan membuat Sistem Manajemen Aset. Sistem Manajemen Aset itu berguna untuk pengelolaan aset oleh sebut saja PT SGL. PT SGL memanfaatkan Sistem Manajemen Aset untuk:

1. Mencatat semua aset yang dimiliki seperti laptop, komputer, monitor, printer, mobil operasional dan yang lainnya.
2. Mencatat supplier pemasok aset PT SGL.
3. Mencata 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:

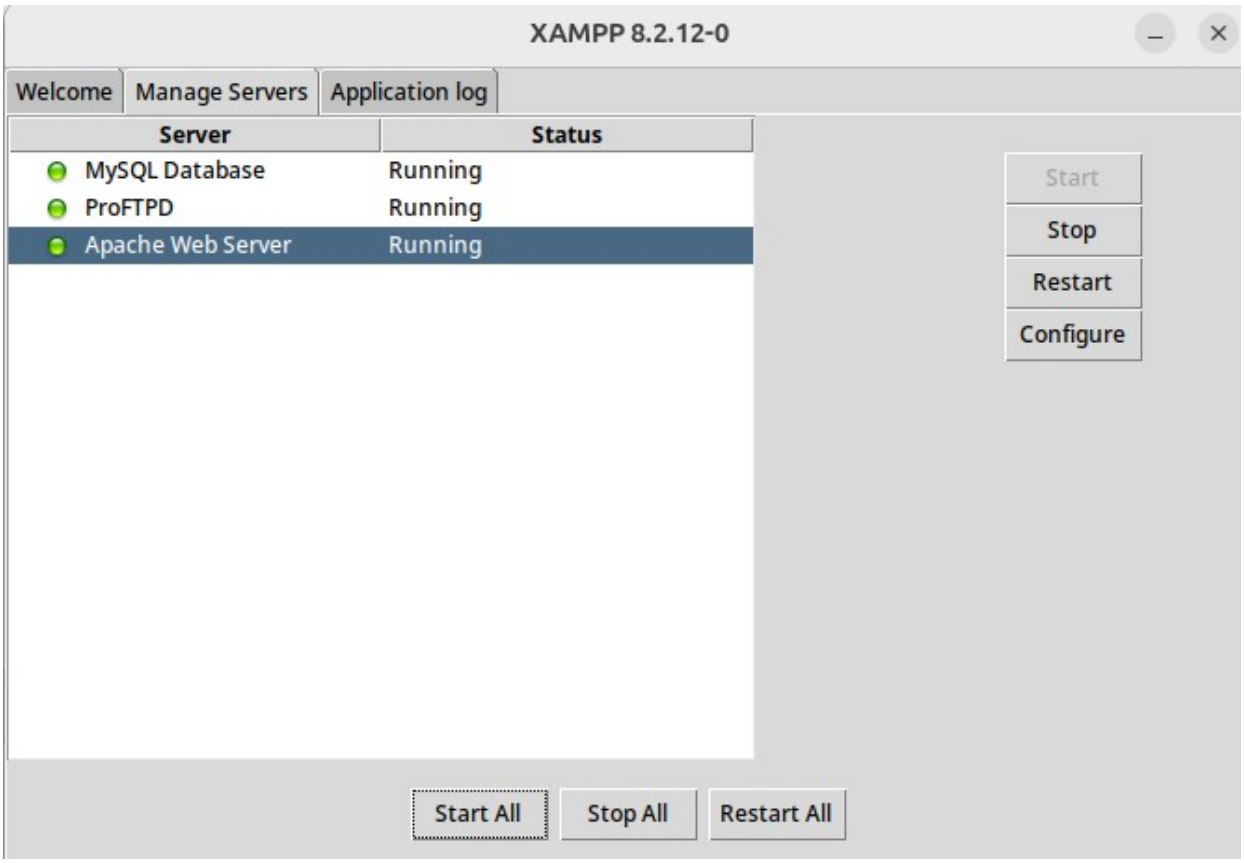
1. Sistem operasi: Windows, Linux, macOs.
   Dalam modul ini di contohkan menggunakan Linux Ubuntu 24.04 LTS.


2. Java 21.




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.


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 langkas 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`

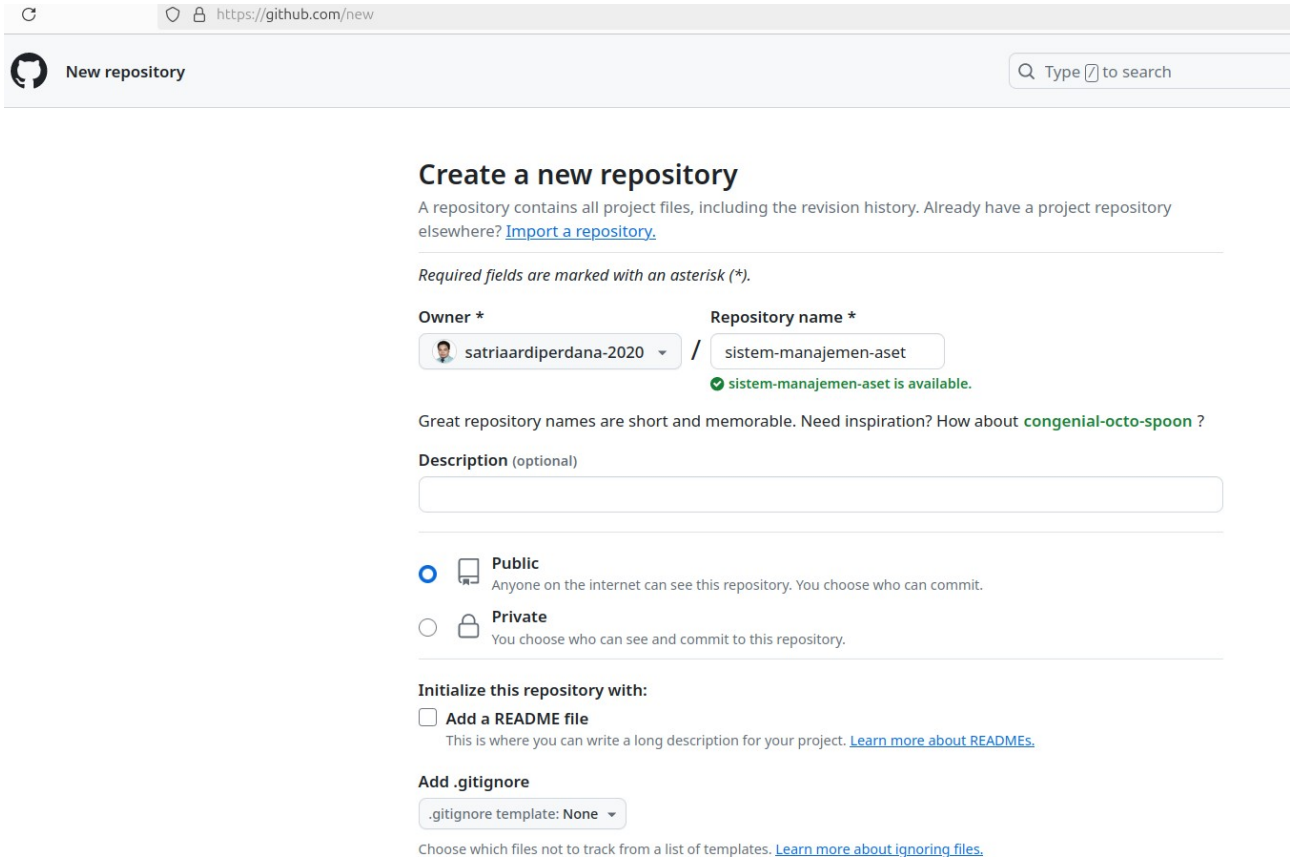
```
ALTER TABLE tb_transaction_order ADD FOREIGN KEY(tbto_tbma_id) REFERENCES tb_master_aset(tbma_id);

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`

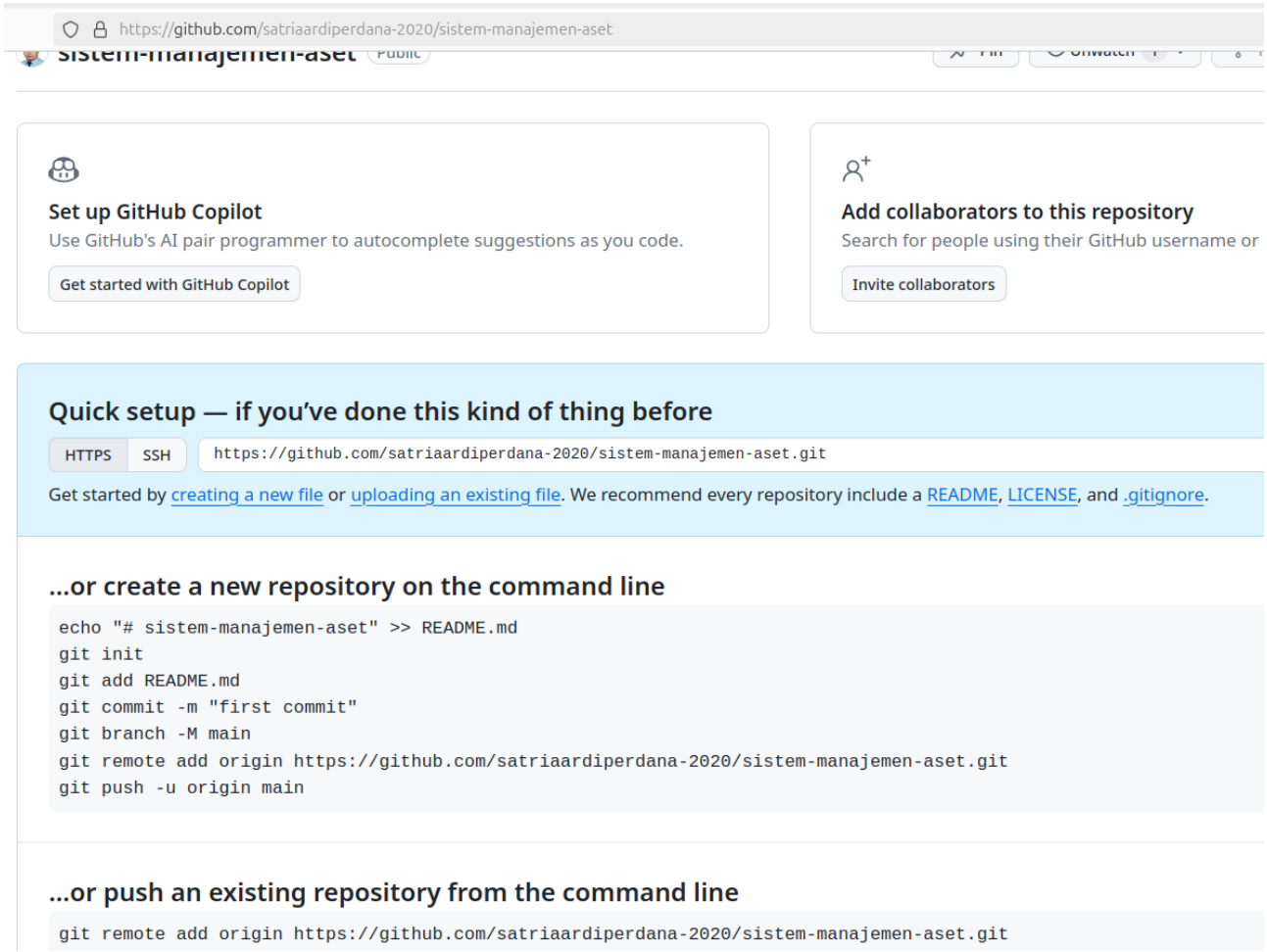
```
ALTER TABLE tb_transaction_pinjam ADD FOREIGN KEY(tbtp_tbma_id) REFERENCES tb_master_aset(tbma_id);
ALTER TABLE tb_transaction_pinjam ADD FOREIGN KEY(tbtp_tbmp_id) REFERENCES tb_master_pegawai(tbmp_id);
```

**MODUL 3**
**BUAT SPRINGBOOT PROJECT**

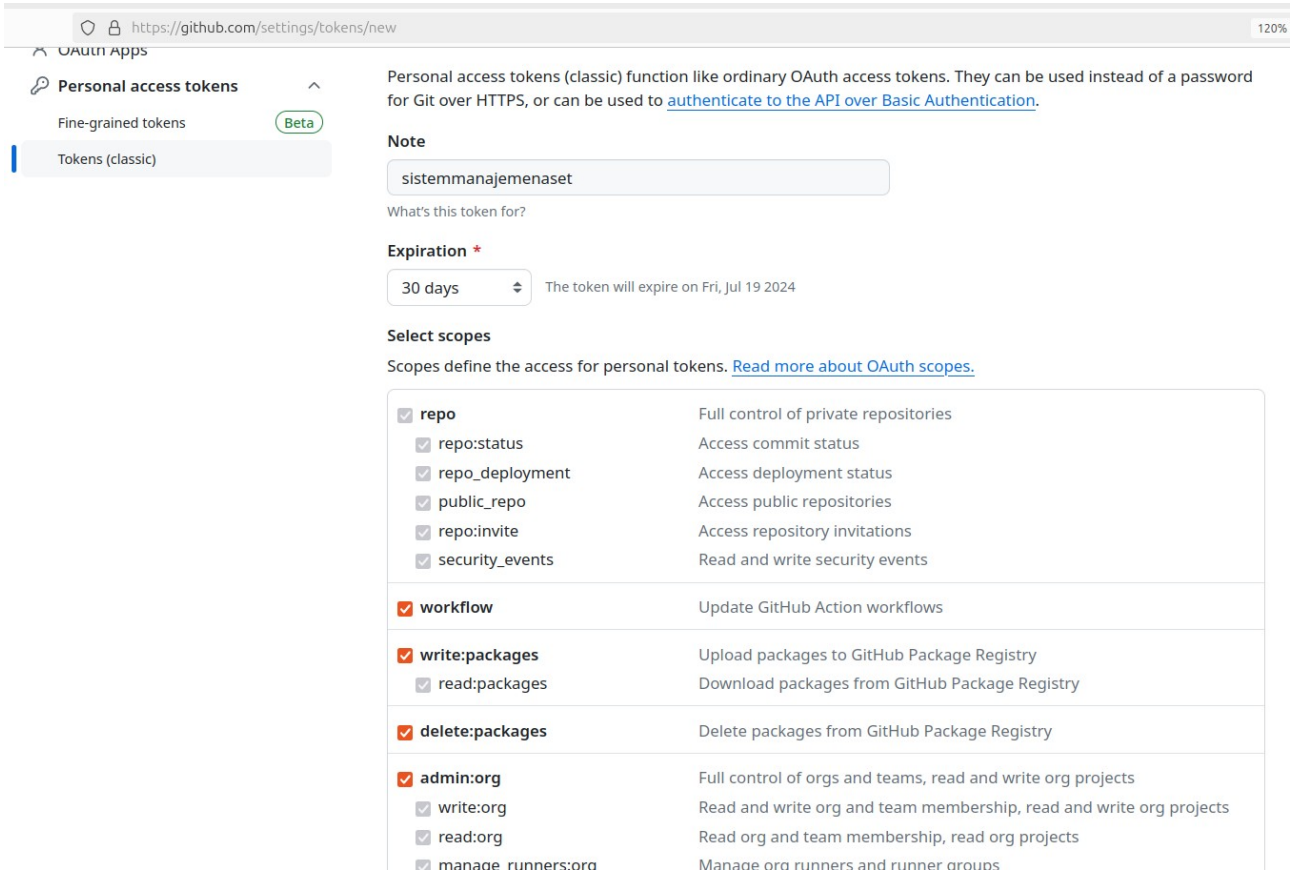
1. Buat repository dengan nama **sistem-manajemen-aset** pada github.com untuk manajemen source code.



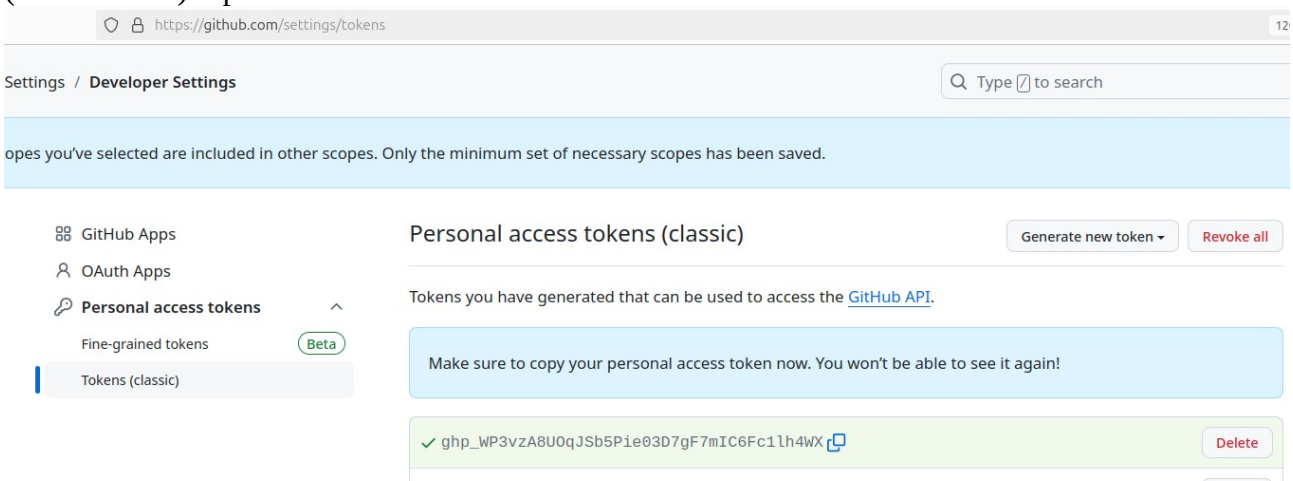

2. Jika berhasil membuat repository pada github muncul seperti berikut:




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:



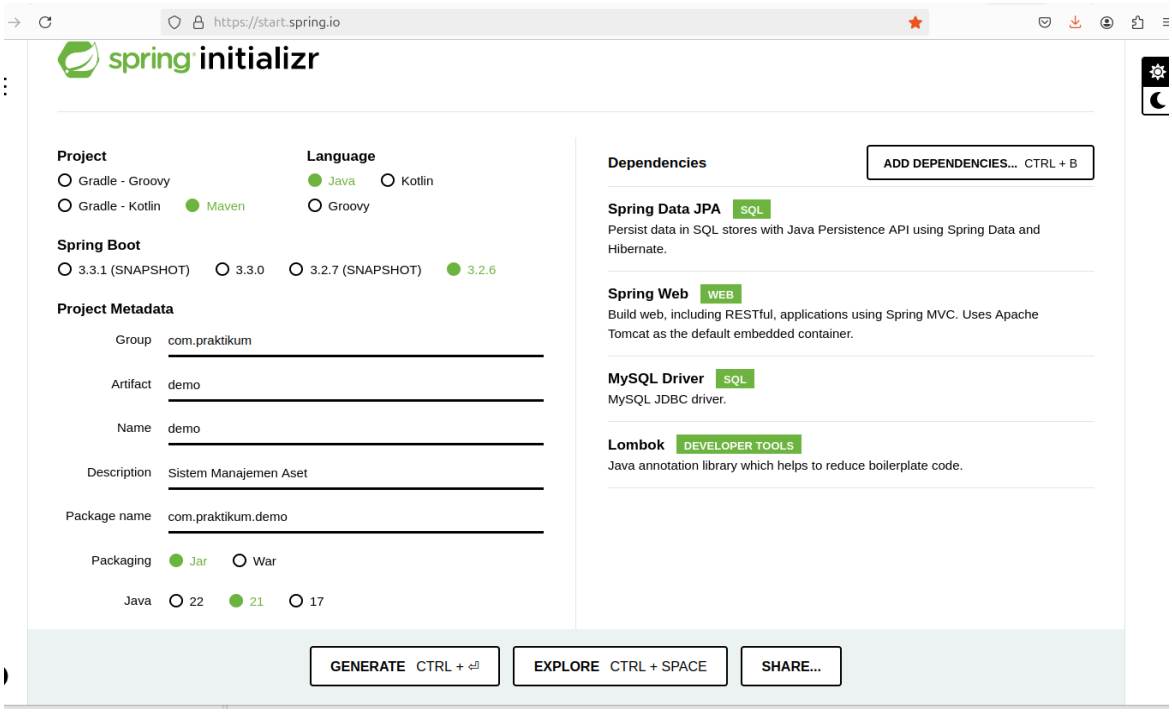

4. Setelah klik button Generate Token (warna hijau) akan muncul toke yang hanya muncul sekali (harus dicatat) seperti berikut:




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

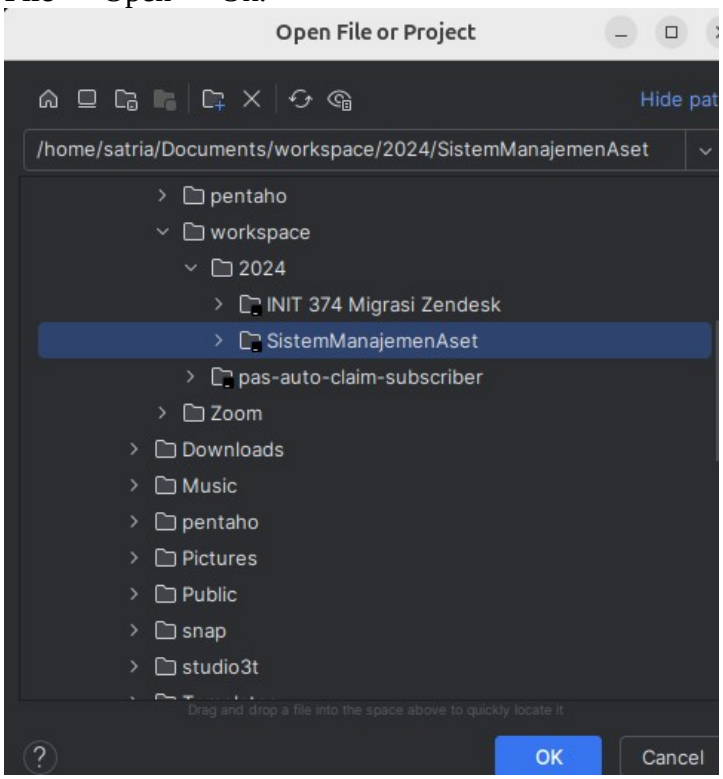
   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

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



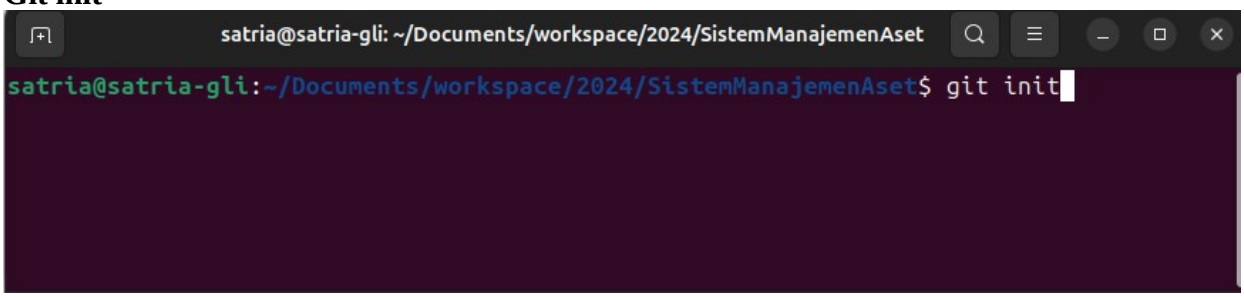

7. Akan terdownload template project sesuai nama artifactnya kemudian extract dan rename menjadi SistemManajemenAset:


8. Import kedalam IDE
   File → Open → Ok:

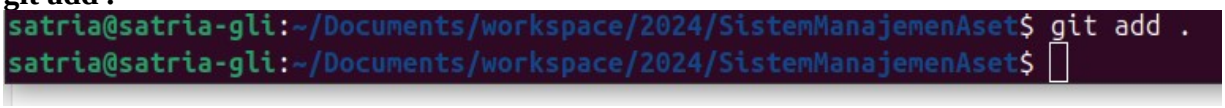


9. Masuk ke path project (SistemManajemenAset) dan inisiasi, commit, push ke github.

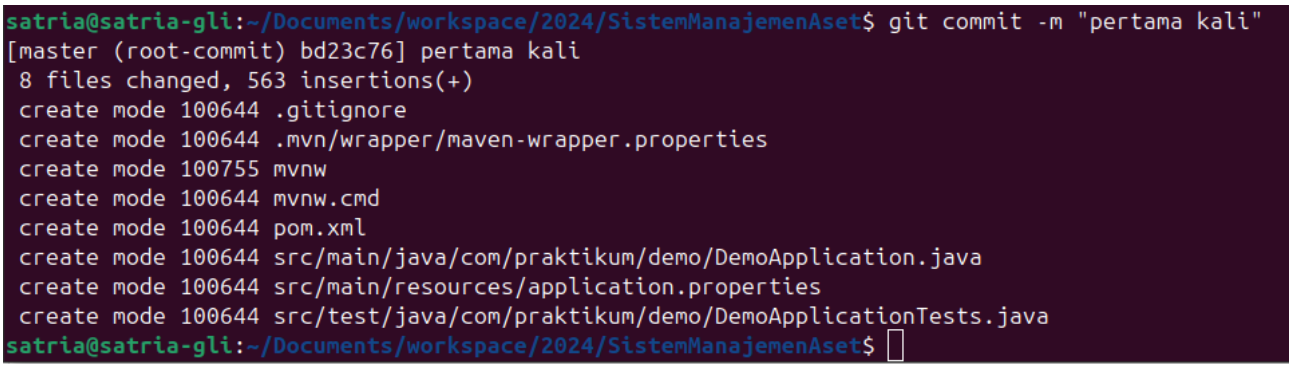
**Git init**



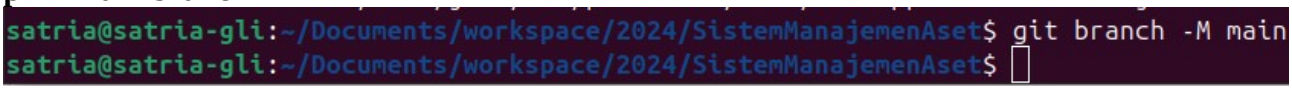

**git add .**



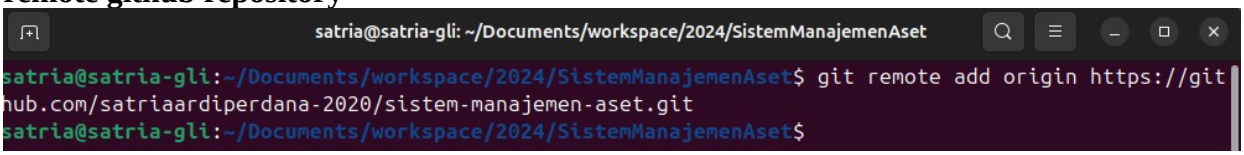

**commit**



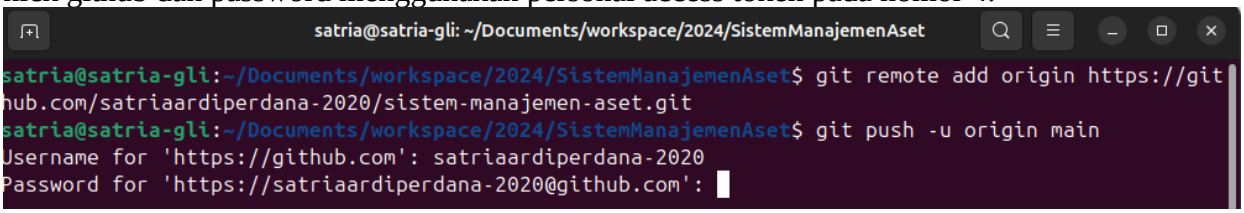

**pilih main branch**



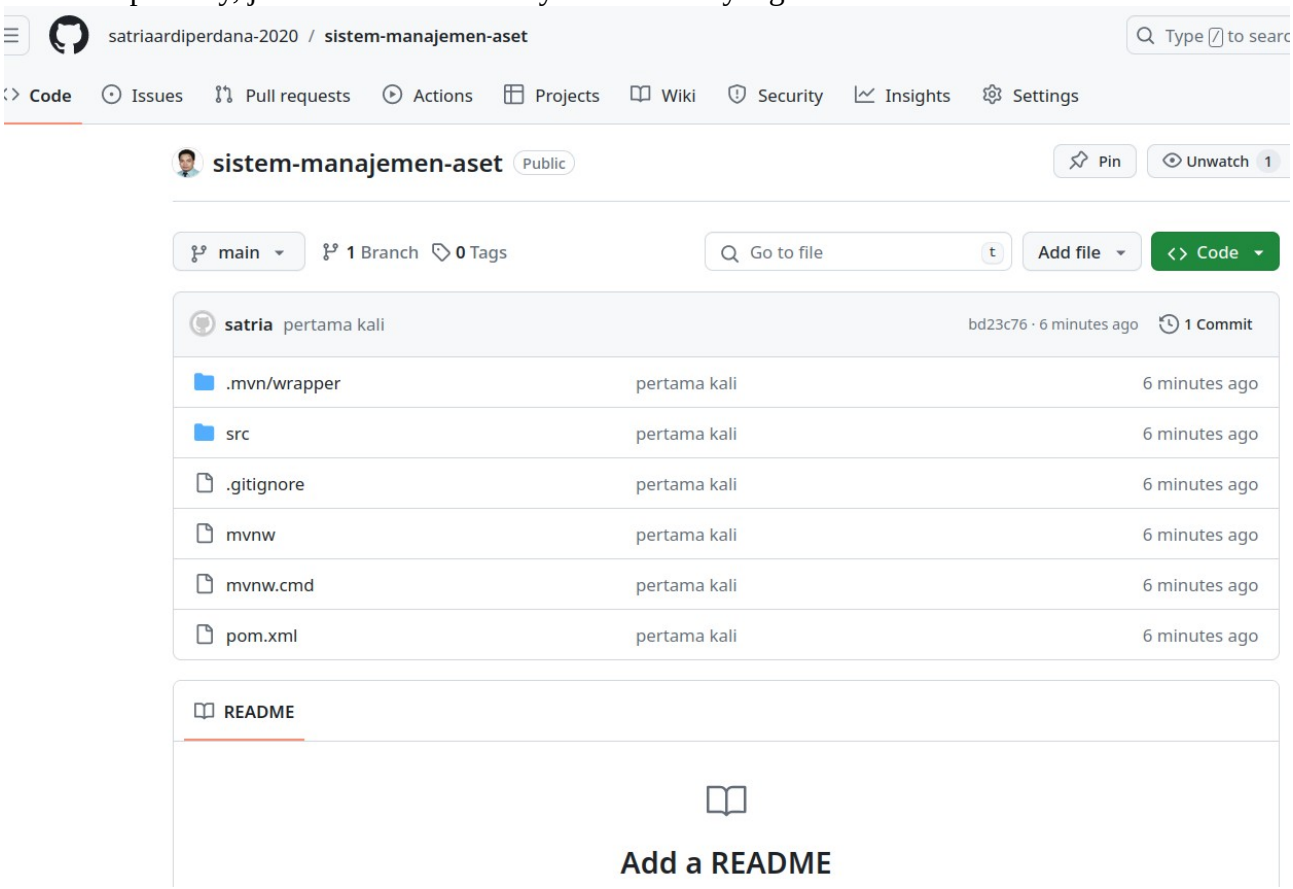

**remote github repository**




**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.



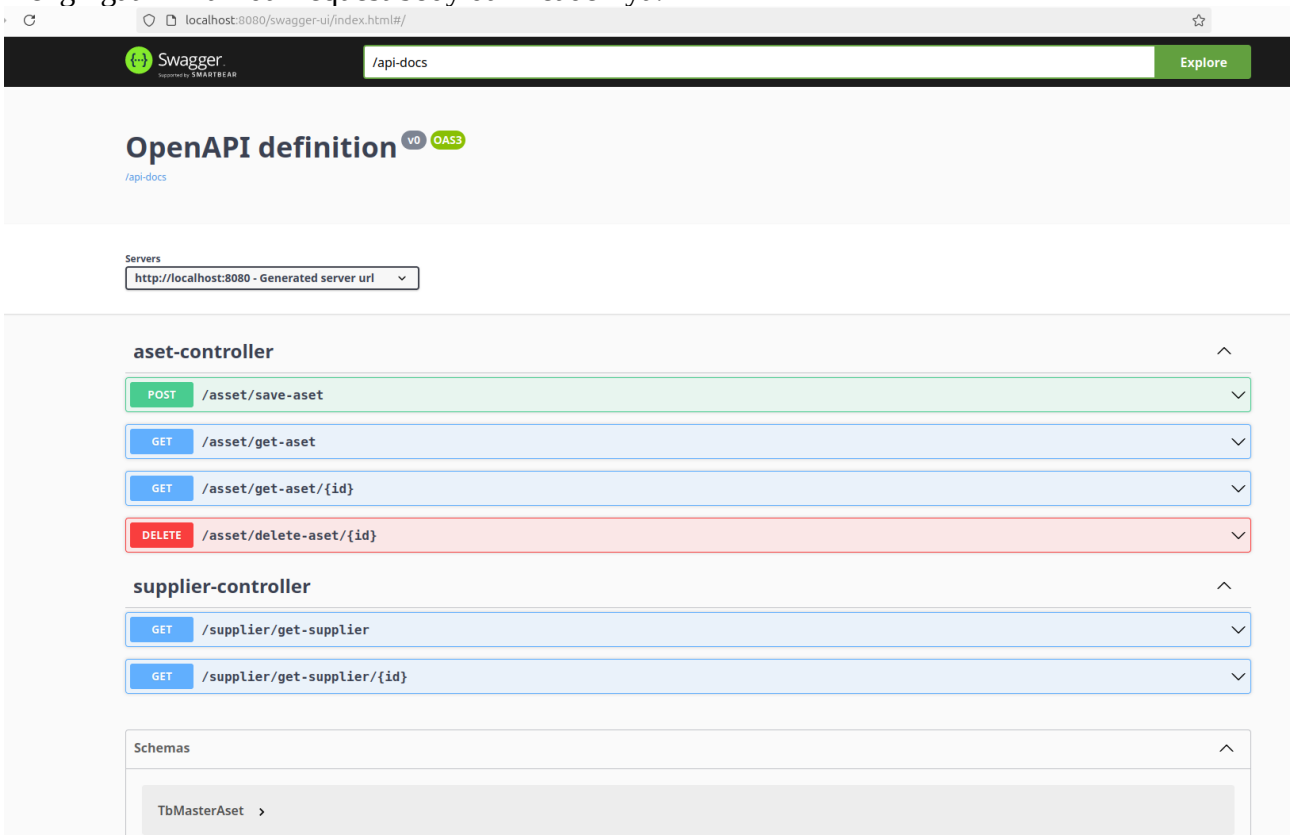
10. Cek di repository, jika berhasil ada history source code yang masuk.



**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 dar request body dan headernya.



# MODUL 5
# FRONT END DAN REPORTING