

**LAPORAN PROJEK UAS PEMROGRAMAN BERBASIS OBJEK
APLIKASI WEB DENGAN JAVA SPRING BOOT DAN THYMELEAF
"SISTEM MANAJEMEN ASET MASJID KAMPUS (SiMaskam)"**



Dosen Pengampu:

Mutaqin, S.S.T., M.T.

Disusun oleh:

Yanuar Nurul Hilal (222112418)

Kelas 2KS4

PROGRAM STUDI DIV KOMPUTASI STATISTIK

POLITEKNIK STATISTIKA STIS

2023

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	2
1.1 Latar Belakang.....	3
1.2 Tujuan.....	3
BAB II.....	5
2.1 Rancangan Kelas Diagram.....	5
2.2 Rancangan Diagram Proses Bisnis.....	6
2.3 Skema Database.....	7
2.4 Panduan Aplikasi.....	8
2.5 Penerapan Design Principles.....	19
2.6 Penerapan Design Pattern.....	21
BAB III.....	23
3.1 Kesimpulan.....	23
3.2 Saran.....	23
3.3 Lampiran.....	23

BAB I

PENDAHULUAN

1.1 Latar Belakang

Masjid Al Hasanah atau yang biasa disebut Masjid Kampus (MasKam) merupakan masjid yang berada di lingkungan kampus Politeknik Statistika STIS. Masjid ini tidak hanya digunakan oleh mahasiswa ataupun pegawai STIS saja, tetapi juga digunakan secara terbuka oleh masyarakat umum, terutama pada pelaksanaan ibadah shalat, baik itu shalat fardhu maupun shalat pada hari besar Islam. Sebagai masjid yang digunakan oleh banyak orang, tentunya masjid harus memiliki fasilitas yang memadai agar kenyamanan dalam beribadah bisa dicapai oleh semua orang dan hal itu tampaknya telah dilakukan oleh pengurus masjid.

Masjid ini memiliki fasilitas yang dapat dikatakan baik, baik dalam segi sarana maupun prasarana. Tidak hanya itu, perawatan secara berkala guna mengecek kelayakan fasilitas juga harus terus dilakukan agar pada masjid ini bisa mempertahankan kriteria “baik” yang tersandang padanya. Pencatatan aset barang masjid menjadi salah satu cara dalam rangka perawatan masjid. Pencatatan ini bisa mendata barang yang ada ataupun sumber barangnya sehingga suatu saat, jika saja barang itu tergolong barang yang awet atau barang yang cepat rusak, maka pengurus bisa melakukan evaluasi tentang pengadaan barang selanjutnya dengan cara mempelajari data pencatatan barang.

Oleh karena itu, penulis telah merancang sebuah aplikasi berbasis web yang dapat digunakan oleh pengurus Masjid Kampus untuk mencatat barang-barang dan mengelola aset yang ada di Masjid tersebut. Aplikasi ini diharapkan dapat mempermudah proses manajemen aset barang, sehingga memudahkan pengelolaan fasilitas Masjid Kampus.

1.2 Tujuan

Tujuan yang ingin dicapai penulis dalam merancang aplikasi ini antara lain:

- 1) Memudahkan proses manajemen aset barang: Aplikasi ini akan membantu pengurus Masjid Kampus dalam mencatat dan mengelola barang-barang yang ada di masjid dengan lebih efisien.
- 2) Meningkatkan efektivitas perencanaan pengadaan barang: Dengan adanya data pencatatan yang terdokumentasi dengan baik, pengurus dapat melakukan evaluasi terhadap aset barang yang ada di masjid. Informasi ini dapat membantu pengurus dalam mengambil keputusan yang tepat mengenai pengadaan barang baru atau penggantian barang yang rusak agar kebutuhan fasilitas masjid terpenuhi dengan baik.
- 3) Meningkatkan transparansi dan akuntabilitas: Dengan menggunakan aplikasi berbasis web, proses pencatatan dan pengelolaan aset barang menjadi lebih terstruktur dan transparan. Data yang tercatat secara rinci akan memudahkan pengurus dalam memberikan laporan kepada pihak terkait, seperti manajemen kampus atau donatur, yang menginginkan informasi mengenai aset masjid.

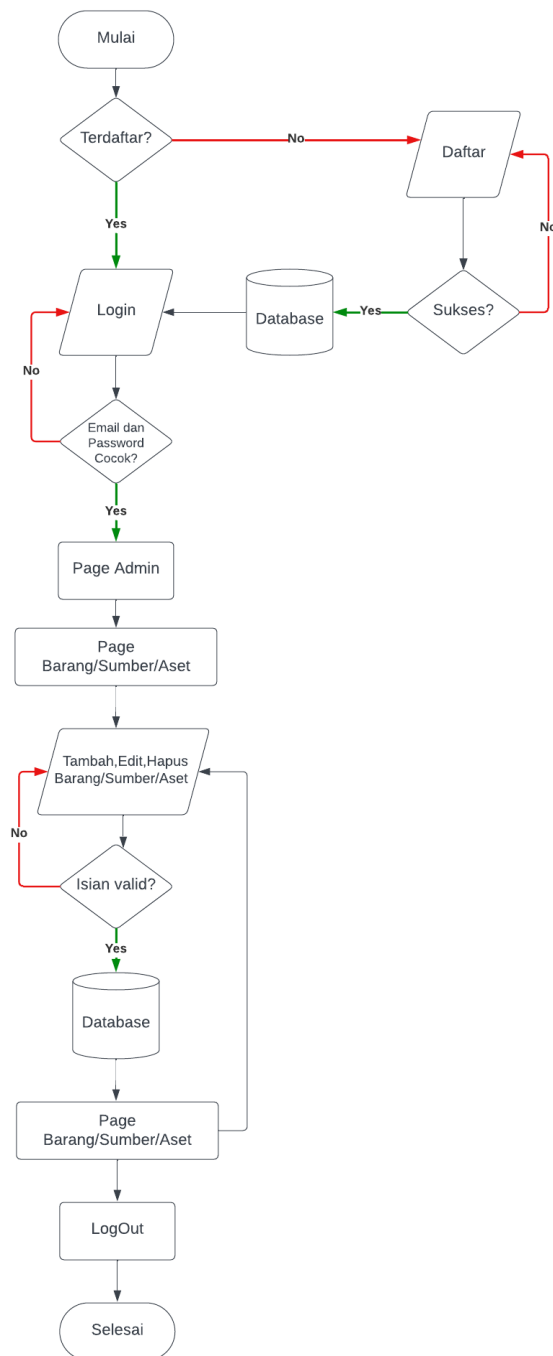
ISI

[illegible]

Aplikasi ini terdiri dari beberapa kelas utama, antara lain: Barang, Sumber, Aset, User, dan Role. Kelas Barang mewakili entitas atau objek barang dalam sistem, sedangkan Kelas Sumber mewakili entitas yang berkaitan dengan sumber suatu barang, seperti pemasok atau penyedia atau pemberi barang. Kelas Aset digunakan untuk merepresentasikan barang beserta sumbernya. Kelas User digunakan untuk menggambarkan entitas atau objek pengguna aplikasi, sedangkan Kelas Role digunakan untuk menggambarkan entitas atau objek peran atau hak akses pengguna. Kelas utama, yaitu SmakApplication, berperan sebagai titik masuk atau entry point dari aplikasi. Karena penulis menggunakan arsitektur MVC (Model, View, Controller) tentunya juga terdapat kelas Controller yang bertugas untuk menghubungkan model/entity dengan view. Kelas Controller menerima request dari user, memproses dan melakukan tindakan yang sudah diprogram. Lalu juga ada kelas ServiceImpl yang memiliki *dependency* pada kelas Repository yang mana kelas Repository meng-*extends method-method* yang ada pada JpaRepository. *Dependency* tersebut dibangun untuk berinteraksi dengan database. Kelas ServiceImpl juga mengimplementasikan *method* pada kelas Service yang menyediakan deklarasi metode yang harus diimplementasikan. Selain itu juga terdapat Kelas SpringSecurity dan CustomAuthenticationSuccessHandler memiliki peran penting dalam fitur login dan keamanan aplikasi. Kelas SpringSecurity

digunakan untuk mengelola fitur keamanan dalam aplikasi, sementara CustomAuthenticationSuccessHandler mengimplementasikan logika khusus yang dieksekusi setelah pengguna berhasil melakukan proses otentikasi atau login.

2.2 Rancangan Diagram Proses Bisnis



Untuk melihat tampilan diagram secara penuh bisa mengunjungi tautan berikut :
<https://drive.google.com/file/d/1u-QIG69zT2OrpHv-fu2W6Y0rUaU3OhxO/view?usp=sharing>

Saat pertama kali membuka aplikasi, user akan langsung dihadapkan pada halaman login, jika belum memiliki akun, user bisa melakukan pendaftaran terlebih dahulu pada menu register, jika register tidak sukses maka user akan diminta kembali melakukan pendaftaran, tetapi jika register telah sukses, user akan kembali dihadapkan pada halaman login. Login akan sukses jika kombinasi email dan password telah sesuai. Setelah login sukses, user akan dihadapkan pada page admin, di page tersebut terdapat navigasi Barang, Sumber, dan Aset masuk. User bisa berpindah antarnavigasi dan bisa melakukan proses melihat, menambah, mengedit, dan menghapus setiap entitas Barang, Sumber, atau Aset. Begitu telah selesai, user bisa melakukan logout dan proses bisnis telah selesai.

2.3 Skema Database

Pada pengerjaan proyek kali ini, penulis menggunakan driver database PostgreSQL yang skema tabelnya dapat dilihat pada gambar berikut.



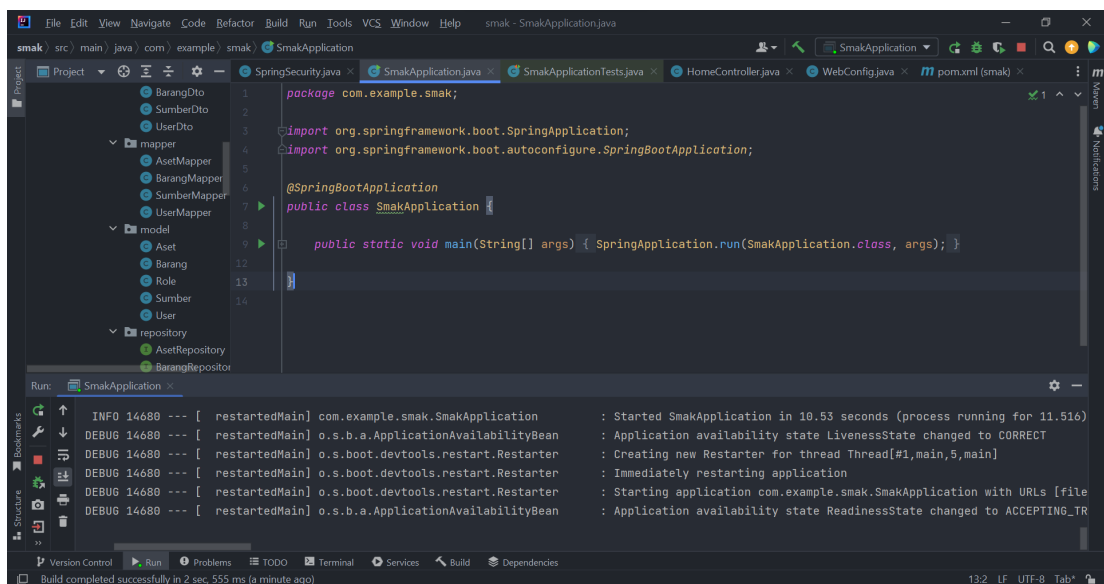
Dalam database yang dibuat, akan ada 6 tabel yang dibagi lagi menjadi 2 bagian utama yaitu tentang pencatatan barang (bagian kiri) dan akun (bagian kanan).

Pada bagian pencatatan barang, melibatkan 3 tabel yaitu barangs, yang berisi atribut dari barang yang akan dicatat, lalu sumbers, berisi atribut dari sumber suatu barang, dan asets yang memetakan barang dan sumbernya, hal ini dapat dilihat dari kolom asets yang berisi barang_id yang merupakan kolom id pada barangs dan kolom sumber_id yang merupakan kolom id pada tabel sumbers.

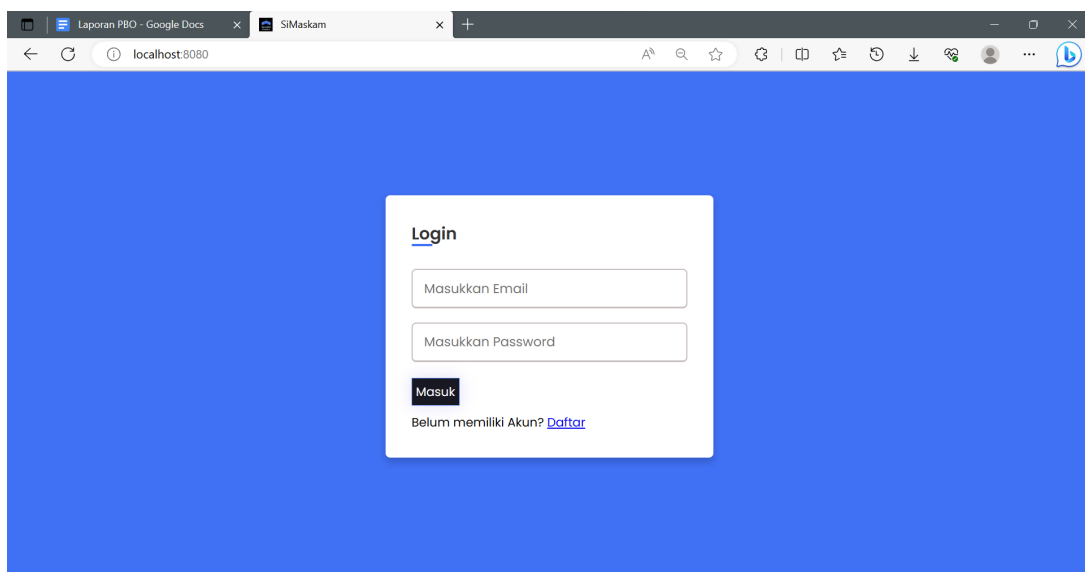
Pada bagian akun, melibatkan 3 tabel juga, yaitu users, yang berisi dari user atau pengguna yang telah melakukan registrasi, lalu ada tabel roles, perlu diperhatikan pada aplikasi ini hanya memiliki satu role, yaitu user sekaligus admin untuk mengelola data, dan ada tabel users_roles yang memetakan user dengan role-nya, hal ini dapat dilihat dari kolom users_roles yang berisi user_id yang merupakan kolom id pada tabel users dan kolom role_id yang merupakan kolom id pada tabel role.

2.4 Panduan Aplikasi

- 1) Hal pertama adalah build dan run aplikasi terlebih dahulu

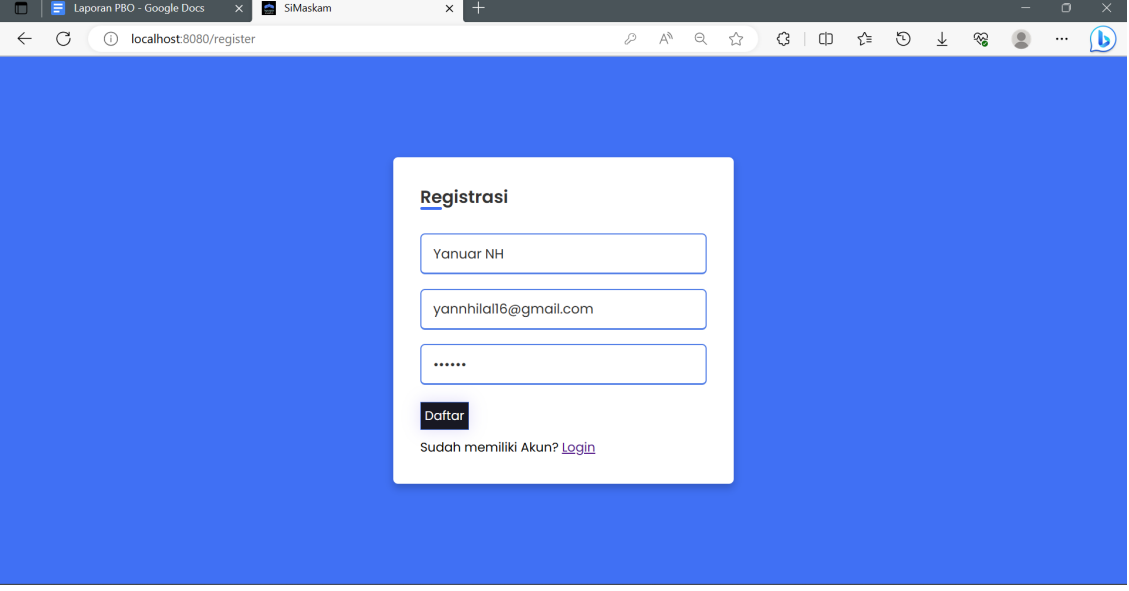


- 2) Setelah itu buka localhost:8080 pada browser



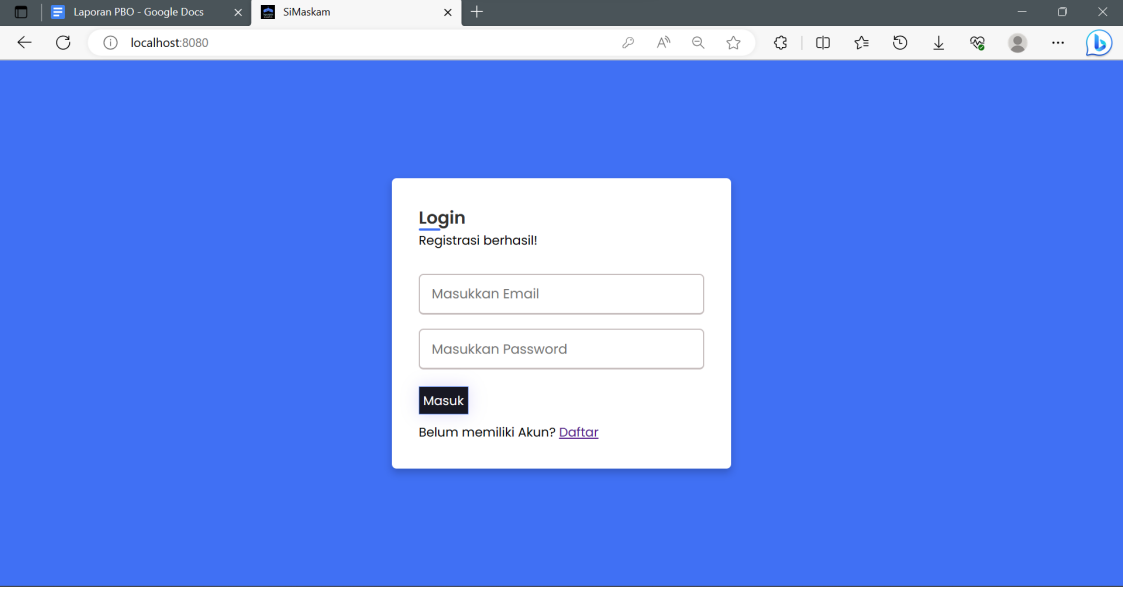
Halaman yang pertama kali muncul adalah halaman login, tetapi jika pengguna belum memiliki akun, bisa mendaftar terlebih dahulu pada menu register.

- 3) Lakukan registrasi terlebih dahulu dengan memasukkan nama, email dan password, setelah itu klik daftar



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/register'. The page has a solid blue background. In the center, there is a white rectangular form titled 'Registrasi'. The form contains three input fields: the first is labeled 'Yanuar NH', the second is labeled 'yannhilal16@gmail.com', and the third is a password field with six dots. Below the password field is a black button with the text 'Daftar' in white. Underneath the button, there is a link that says 'Sudah memiliki Akun? [Login](#)'.

- 4) Akan muncul pesan “Registrasi Berhasil!” pada bagian bawah login yang artinya registrasi berhasil dilakukan, lalu dapat dipastikan dengan mengecek pada database bahwa user telah berhasil terdaftar



The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The page has a solid blue background. In the center, there is a white rectangular form titled 'Login'. Below the title, it says 'Registrasi berhasil!'. The form contains two input fields: the first is labeled 'Masukkan Email' and the second is labeled 'Masukkan Password'. Below the password field is a black button with the text 'Masuk' in white. Underneath the button, there is a link that says 'Belum memiliki Akun? [Daftar](#)'.

The screenshot shows the pgAdmin 4 interface. On the left, the 'public' database is selected, and the 'users' table is highlighted under 'Tables (6)'. The central pane displays a SQL query:

```
1 SELECT * FROM public.users
2 ORDER BY id ASC
```

The 'Data Output' pane shows the results of the query in a table format:

id	created_at	email	name	password	updated_at
10	2023-06-30 11:00:07.064	yanuarhilal@gmail.com	Yanuar Nurul Hilal	\$2a\$10\$3zDQyCGdG...	2023-06-30 11:00:07.064
11	2023-06-30 11:10:01.447	admin@gmail.com	Admin	\$2a\$10\$SeqZsLyCBC...	2023-06-30 11:10:01.448
12	2023-06-30 17:09:11.148	yanuar@gmail.com	yanuar	\$2a\$10\$K0tUF9JoAIW...	2023-06-30 17:09:11.148
13	2023-06-30 22:09:35.712	yanhilal16@gmail.com	Yanuar NH	\$2a\$10\$SLqP2JnrrP31...	2023-06-30 22:09:35.712

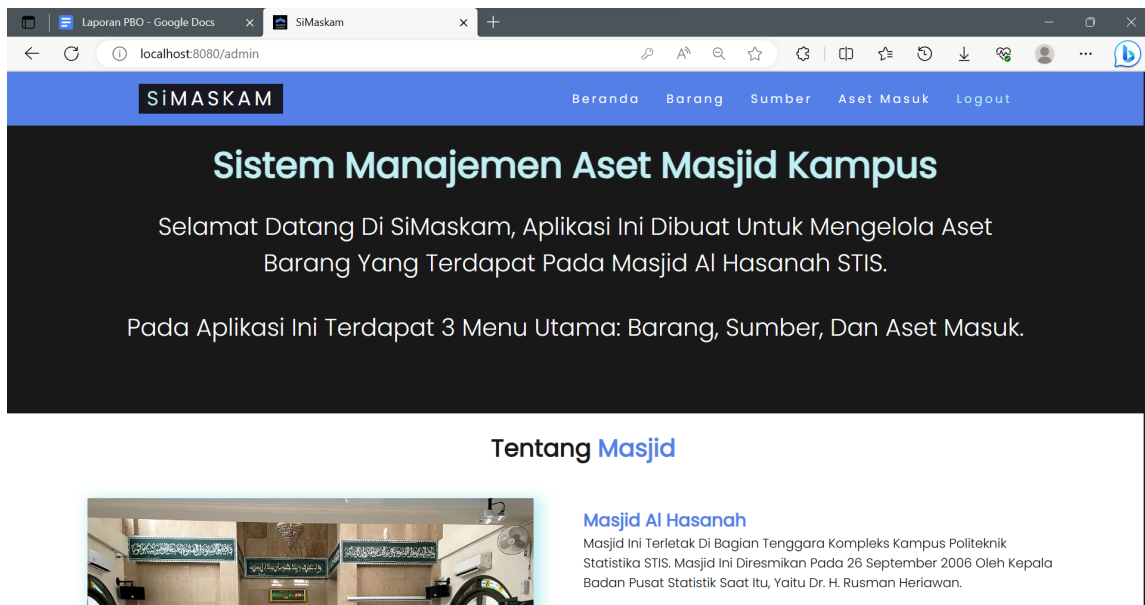
The status bar at the bottom indicates 'Total rows: 4 of 4' and 'Query complete 00:00:00.216'.

5) Login menggunakan email dan password yang telah terdaftar tadi

The screenshot shows a web browser window with the address bar set to 'localhost:8080'. The page has a solid blue background. In the center, there is a white login form with the following elements:

- Login** (Section Header)
- Registrasi berhasil!
- Email input field containing 'yanhilal16@gmail.com'
- Password input field with masked characters '.....'
- Masuk** (Login Button)
- Link: 'Belum memiliki Akun? [Daftar](#)'

6) Jika berhasil user akan langsung beralih pada halaman awal seperti pada gambar, pada tiap halaman juga terdapat navigasi barang, sumber, aset, dan logout.



- 7) Pada menu barang, user bisa menambahkan, melihat, mengedit informasi barang, dan menghapus barang.



SiMASKAM Beranda Barang Sumber Aset Masuk Logout

Form Tambah Barang

Silakan Untuk Melengkapi Informasi Pada Barang Yang Akan Ditambahkan

Nama Barang

Nama Barang

Kode Barang

Kode Barang

Kategori

Kode Barang

Jumlah Barang

SiMASKAM Beranda Barang Sumber Aset Masuk Logout

Form Edit Barang

Silakan Untuk Mengubah Informasi Pada Barang Terkait

Nama Barang

AC

Kode Barang

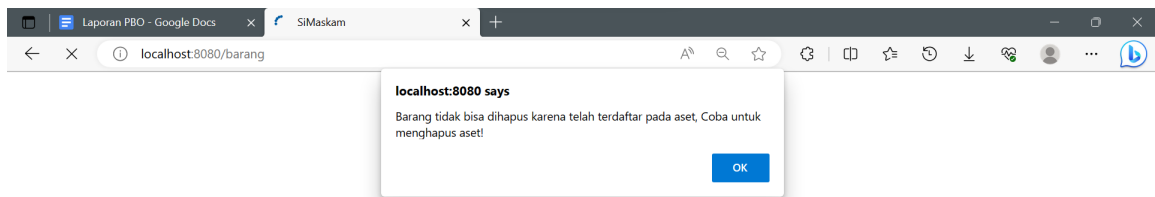
DM101

Kategori

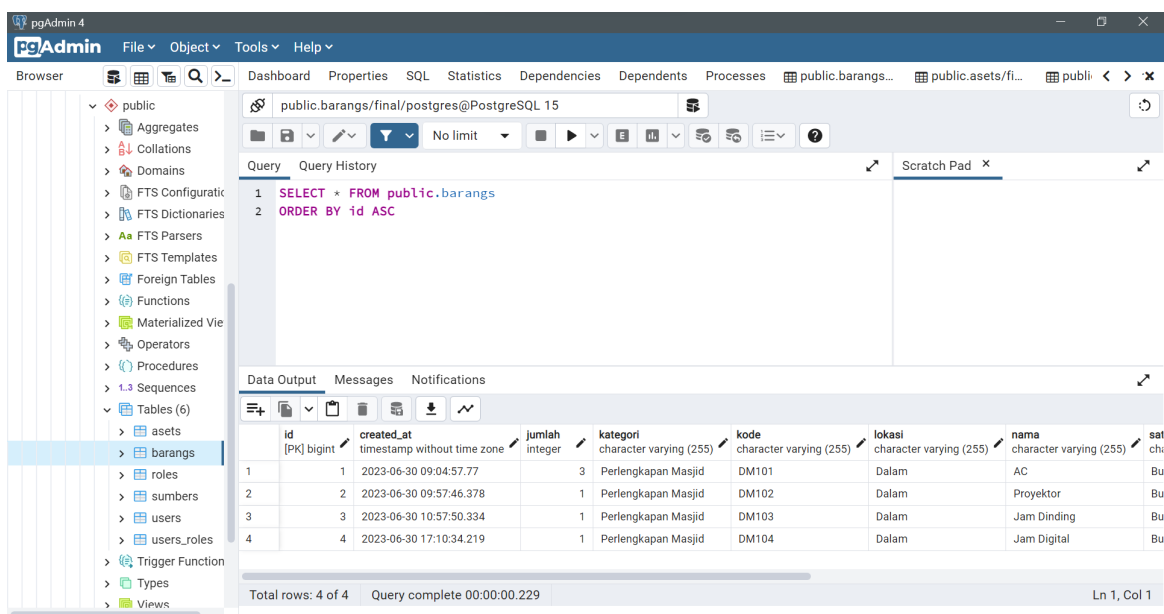
Perlengkapan Masjid

Jumlah Barang

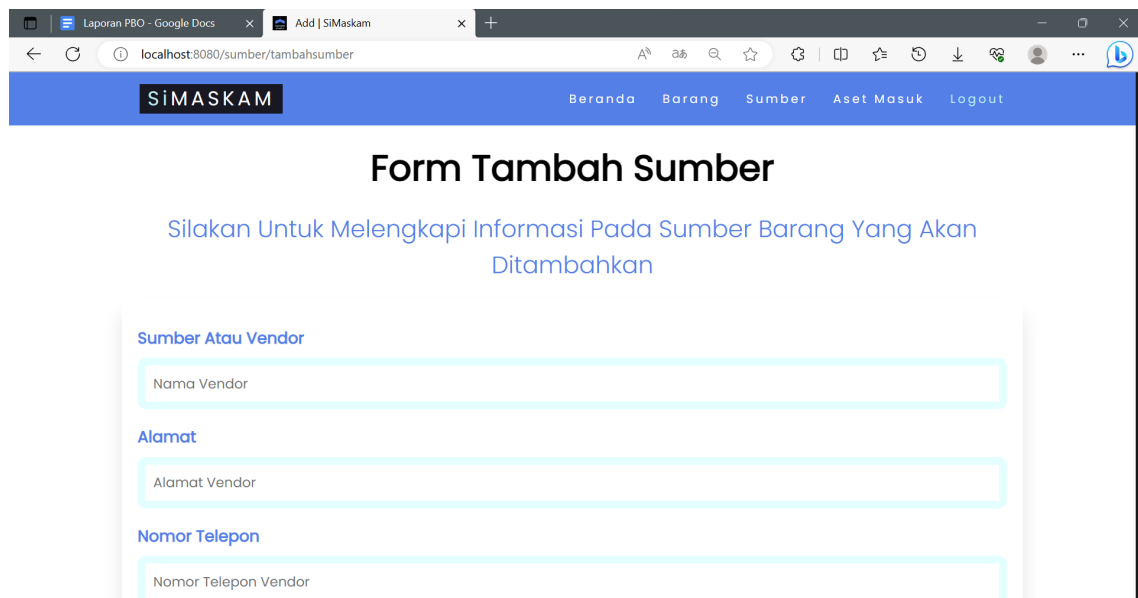
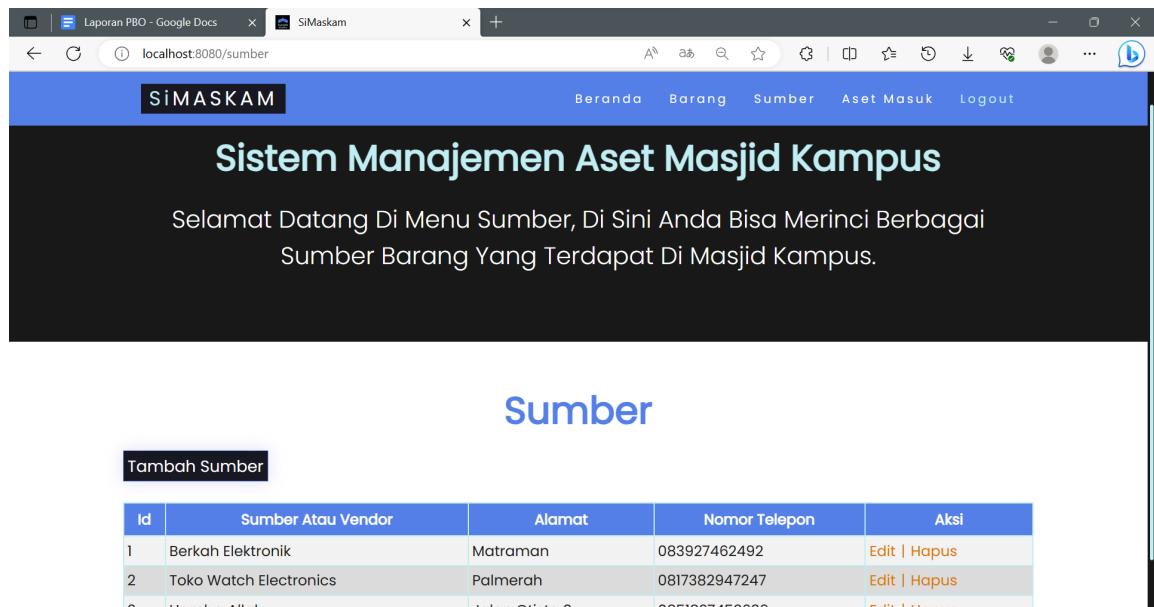
Hal yang perlu diperhatikan saat menghapus barang adalah barang tidak bisa dihapus saat barang sudah ditambahkan ke dalam aset. Barang hanya bisa dihapus apabila barang belum atau tidak ditambahkan pada aset.



Setiap perubahan pada barang juga akan diikuti perubahan yang serupa pada database



8) Pada menu sumber, user bisa menambahkan, melihat, mengedit informasi tentang sumber, ataupun menghapus sumber.



SiMaskam Beranda Barang Sumber Aset Masuk Logout

Form Edit Sumber

Silakan Untuk Mengubah Informasi Pada Sumber Barang Terkait

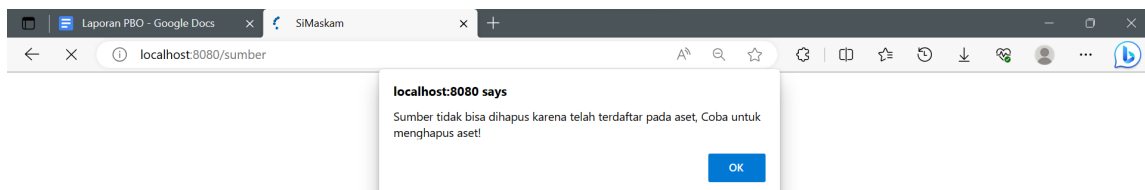
Sumber Atau Vendor
Berkah Elektronik

Alamat
Matraman

Nomor Telepon
083927462492

Submit

Sama seperti menu barang, aksi menghapus sumber tidak bisa dihapus saat sumber sudah ditambahkan ke dalam aset (dipetakan dengan barang). Sumber hanya bisa dihapus apabila sumber belum atau tidak ditambahkan pada aset.



Setiap perubahan yang terjadi pada sumber juga akan terjadi pada database.

The screenshot shows the pgAdmin 4 interface. On the left is a tree view of the database structure. The main pane displays a SQL query: `SELECT * FROM public.somers ORDER BY id ASC`. Below the query, the 'Data Output' tab shows the results of the query in a table format.

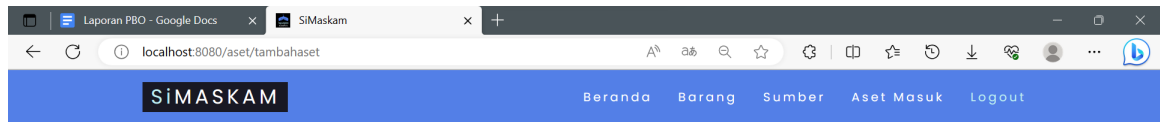
id [PK] bigint	alamat character varying (255)	created_at timestamp without time zone	notelp character varying (255)	updated_at timestamp without time zone	vendor character varying (255)
1	Matraman	2023-06-30 09:05:22.518	083927462492	2023-06-30 09:06:57.261	Berkah Elektronik
2	Palmerah	2023-06-30 17:12:26.152	0817382947247	2023-06-30 17:12:26.152	Toko Watch Electronics
3	Jalan Otista 3	2023-06-30 17:26:06.968	0851827452639	2023-06-30 17:26:06.968	Hamba Allah

Total rows: 3 of 3 Query complete 00:00:00.403 Ln 1, Col 1

- 9) Pada menu aset, user bisa menambahkan dalam hal ini memetakan antara barang dan sumber, melihat, mengedit informasi tentang aset, ataupun menghapus aset.

The screenshot shows the SiMaskam web application interface. The header includes the logo 'SiMaskam' and navigation links: Beranda, Barang, Sumber, Aset Masuk, and Logout. The main content area displays the title 'Sistem Manajemen Aset Masjid Kampus' and a welcome message. Below this, there is a section titled 'Aset' with a 'Tambah Aset' button and a table listing assets.

Id	Nama Barang	Nama Vendor	Tanggal Dan Waktu Pencatatan	Aksi
1	Proyektor	Berkah Elektronik	2023-06-30 09:05:30.637	Edit Hapus
4	Jam Digital	Toko Watch Electronics	2023-06-30 22:24:28.286	Edit Hapus



Form Tambah Aset

Silakan Untuk Melengkapi Informasi Pada Aset Yang Akan Ditambahkan

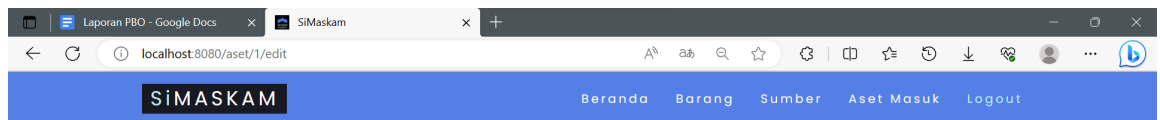
Nama Barang

AC

Nama Vendor

Berkah Elektronik

Submit



Form Edit Aset

Silakan Untuk Mengubah Informasi Pada Aset Terkait

Nama Barang

Proyektor

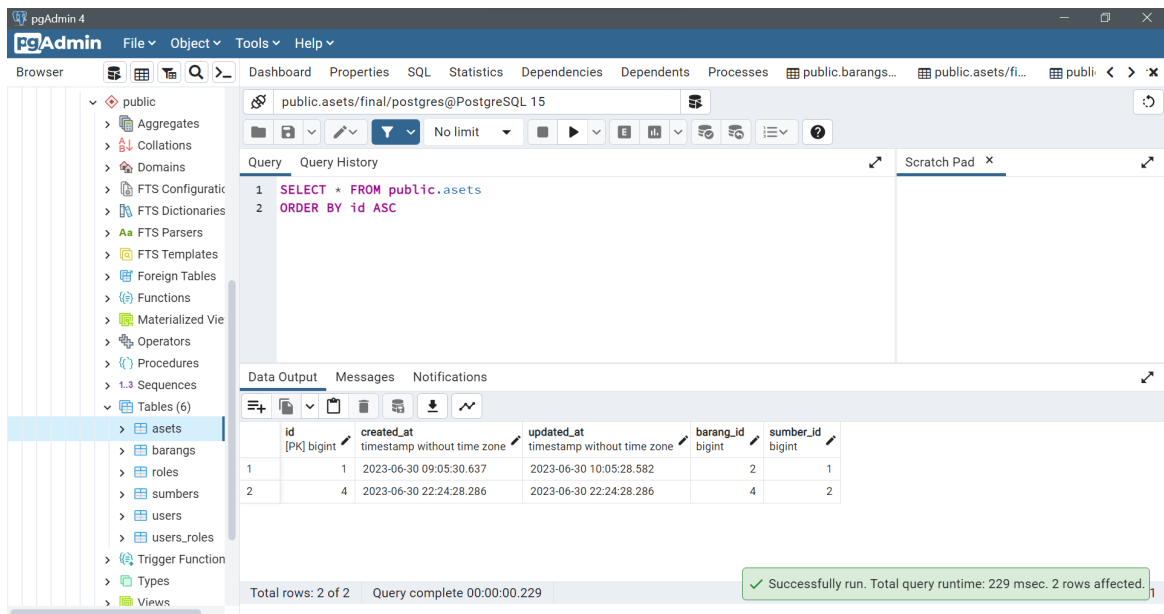
Nama Vendor

Berkah Elektronik

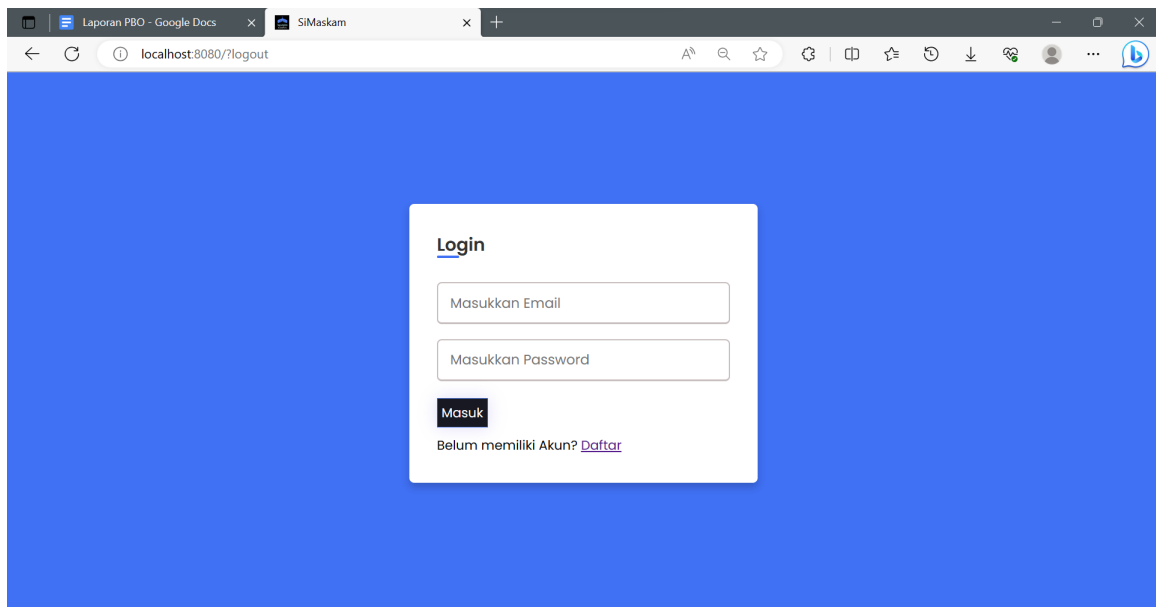
Submit

Penghapusan aset juga bisa dilakukan dengan cara klik aksi hapus pada baris tertentu, aset yang dihapus juga akan berpengaruh terhadap barang dan sumber. Keduanya juga akan bisa dihapus.

Setiap perubahan yang terjadi pada sumber juga akan terjadi pada database



10) User bisa melakukan logout dengan cara mengklik navigasi logout dan akan diarahkan ke halaman login, tanda logout sukses bisa dilihat pada url, yaitu <http://localhost:8080/?logout>



11) Aplikasi dirancang sesederhana mungkin agar user dapat dengan mudah menggunakan aplikasi ini.

2.5 Penerapan Design Principles

1. Single Responsibility Principles

Single Responsibility Principle (SRP) adalah prinsip desain berorientasi objek yang mengharuskan setiap kelas atau objek memiliki satu tanggung jawab utama saja.

Penerapan:

Pada kelas Model (Barang, Sumber, Aset, User, dan Role) hanya bertanggung jawab untuk mengelola data dan logika terkait pengolahan data saja. Lalu pada kelas Controller (BarangController, SumberController, dst) hanya bertanggung jawab untuk menerima dan memproses permintaan dari user. Selanjutnya pada kelas Repository (BarangRepository, SumberRepository, dst) hanya bertanggung jawab terkait operasi yang berhubungan dengan akses data, seperti pengambilan, dan penyimpanan data.

2. Open/Closed Principles

Open/Closed Principle (OCP) adalah prinsip desain berorientasi objek yang mengharuskan sebuah kelas atau objek harus terbuka untuk perluasan (open) dan tertutup untuk modifikasi (closed).

Penerapan:

Pada Interface AsetService menyediakan kontrak untuk method-method tertentu yang terkait dengan entitas Aset. Interface ini terbuka untuk perluasan yang memungkinkan untuk penambahan method baru dan sekaligus tertutup untuk modifikasi pada method yang sudah ada padanya. Interface AsetService diimplementasi oleh kelas AsetServiceImpl yang juga pada kelas itu bisa menambahkan method baru tanpa memengaruhi Interface AsetService.

3. Liskov Substitution Principles

Liskov Substitution Principle (LSP) adalah prinsip desain berorientasi objek yang menyatakan bahwa objek dari sebuah kelas harus dapat diganti dengan objek dari subkelasnya tanpa mengubah fungsionalitas program.

Penerapan:

Interface BarangRepository dan SumberRepository merupakan turunan dari JpaRepository yang mana kedua Interface tersebut mewarisi method-method

yang didefinisikan oleh JpaRepository seperti findAll(), save(), dll yang mana perilaku dari kedua interface tersebut konsisten dengan perilaku induknya dan kedua interface bisa digunakan sebagai pengganti interface JpaRepository karena memiliki fungsionalitas yang sama

4. Interface Segregation Principles

Interface Segregation Principle (ISP) adalah prinsip desain berorientasi objek yang menyatakan bahwa antarmuka (interface) yang besar dan kompleks harus dipecah menjadi beberapa antarmuka yang lebih kecil dan lebih spesifik. Hal ini dilakukan untuk menghindari adanya ketergantungan pada antarmuka yang tidak diperlukan dan meminimalkan efek samping dari perubahan pada antarmuka.

Penerapan:

Method pada semua Interface Service (BarangService, UserService, dst) adalah method yang relevan dengan fungsinya. Interface Service hanya menyediakan method yang sesuai dengan kebutuhan saja dan Kelas ServiceImpl, kelas yang mengimplementasi Interface Service, hanya akan bergantung pada yang method-method tersebut dan tidak bergantung pada method lain yang tidak relevan dengan fungsionalitasnya.

5. Dependency Inversion Principles

Dependency Inversion Principle (DIP) adalah prinsip desain berorientasi objek yang menyatakan bahwa modul yang lebih tinggi tidak harus bergantung pada modul yang lebih rendah.

Penerapan:

Pada Interface Service (BarangService, SumberService, dst) berperan sebagai abstraksi yang mendefinisikan kontrak method yang berkaitan dengan modelnya masing-masing, lalu komponen yang lebih tinggi yaitu Kelas BarangServiceImpl, SumberServiceImpl, dst bergantung pada abstraksi tersebut dan tidak memiliki ketergantungan lain terhadap kelas lain yang lebih rendah, Ia hanya bergantung kepada abstraksi saja.

2.6 Penerapan Design Pattern

1) Singleton

Singleton adalah sebuah pattern yang digunakan dalam pemrograman untuk memastikan bahwa sebuah kelas hanya memiliki satu instance tunggal, dan menyediakan akses global ke instance tersebut.

Penerapan:

Karena aplikasi ini menggunakan framework Spring maka ketika suatu kelas diberi anotasi `@Service`, `@Repository` atau `@Controller`, kelas itu akan dikelola oleh Spring sebagai beans, dan ketika melakukan anotasi `@Autowired`, Spring akan mencari beans yang sesuai dengan tipe yang diperlukan dan menyediakan instance yang sama (tunggal).

2) Builder

Builder merupakan pattern yang digunakan untuk membuat objek yang kompleks secara bertahap.

Penerapan:

Kelas-kelas pada Model (Barang, Sumber, dst) memiliki Anotasi `@Builder` yang digunakan secara otomatis untuk membuat instance Barang, Sumber, ataupun yang lain dengan mengatur atributnya melalui method `builder()`. Selain itu pada Mapper (BarangMapper, SumberMapper) memiliki method untuk membuat objek `BarangDto`, `SumberDto`, ataupun yang lain dari objek Barang, Sumber, dst. Dalam hal ini pattern Builder membantu dalam membuat objek Barang, Sumber, Aset, dan User dengan kode program yang efisien.

3) Dependency Injection

Dependency Injection merupakan *pattern* dalam pengembangan perangkat lunak untuk mengimplementasikan prinsip Inversion of Control (IoC). Pada pola DI. Objek-objek dependensi (dependencies) dari sebuah kelas tidak diinstansiasi atau dikelola langsung oleh kelas itu sendiri, tetapi diberikan (diinjeksikan) ke dalam kelas tersebut dari luar.

Penerapan:

Pada kelas Controller (BarangController, SumberController, dst) terdapat anotasi `@Autowired` untuk menginjeksikan kelas Service lain sebagai dependensi, lalu pada kelas `ServiceImpl`(BarangServiceImpl,

SumberServiceImpl, dst) terdapat penggunaan anotasi `@Autowired` untuk menginjeksikan kelas Repository sebagai dependensi. Penggunaan anotasi tersebut akan secara otomatis dikelola Spring dengan mencari bean instance yang sesuai dan menginjeksikan ke dalam kelas yang membutuhkan.

BAB III

PENUTUP

3.1 Kesimpulan

Aplikasi ini dibuat untuk memudahkan pengurus dan pengelola Masjid Kampus dalam rangka melakukan pencatatan aset barang menjadi lebih terstruktur dan transparan. serta dengan pencatatan yang terdokumentasi dengan baik, pengurus diharapkan dapat melakukan evaluasi terhadap pengadaan aset barang yang akan datang pada Masjid Kampus.

3.2 Saran

- 1) Dapat dikembangkan lebih lanjut untuk fitur foto barang, sehingga akan semakin mempermudah dalam mendapatkan visual yang baik
- 2) Dapat diperbaiki fitur validasi pada form create barang dan create sumber sehingga mencegah terjadinya inputan salah yang mungkin saja tidak disengaja

3.3 Lampiran

- 1) Video demo aplikasi dapat diakses menggunakan tautan berikut:
https://drive.google.com/file/d/1okooo3NE_juKYEE_5VEtf6-tGxyL5RDj/view?usp=sharing
- 2) Kode program dapat dilihat pada Repository atau dapat diakses menggunakan tautan berikut:
https://git.stis.ac.id/222112418/uas_pbo_2ks4_yanuar-nurul-hilal_222112418/-/tree/main/smak
- 3) Diagram UML dapat diakses menggunakan tautan berikut:
<https://drive.google.com/file/d/1EyPGruju6ltptijKhf-tdKxLrjPAsPHGs/view?usp=sharing>

- 4) Diagram Proses Bisnis dapat diakses menggunakan tautan berikut:

<https://drive.google.com/file/d/1u-QIG69zT2OrpHv-fu2W6Y0rUaU3OhxO/view?usp=sharing>

- 5) Folder yang berisi Video demo aplikasi, Diagram Kelas UML, dan Diagram Proses Bisnis dapat diakses menggunakan tautan berikut:

<https://drive.google.com/drive/folders/1WmOKxql0dlzSt7NR2qvuOp0JTpzNPPvO?usp=sharing>