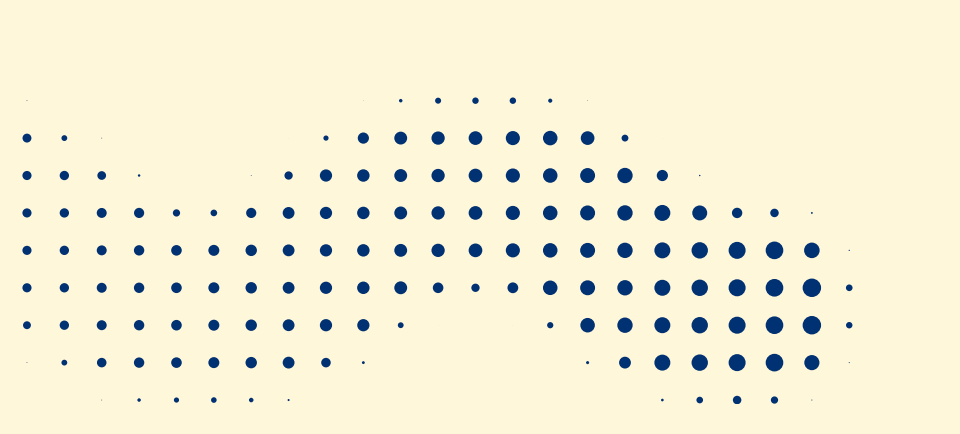


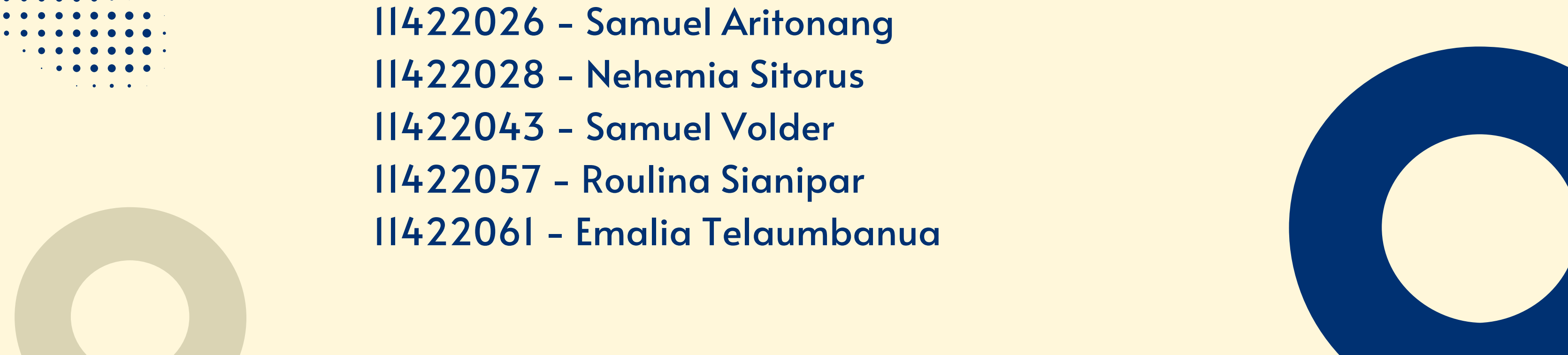


# **APLIKASI PENDATAAN ADMINISTRASI PELAYANAN MAJELIS SE-DISTRIK II SILINDUNG BERBASIS WEBSITE MENGUNAKAN ARSITEKTUR MICROSERVICE**

**Oleh : Kelompok 04**



**11422026 - Samuel Aritonang  
11422028 - Nehemia Sitorus  
11422043 - Samuel Volder  
11422057 - Roulina Sianipar  
11422061 - Emalia Telaumbanua**



# DAFTAR ISI

- Deskripsi Aplikasi
- Fungsi pada Aplikasi
- Perbedaan monolith dan Microservice
- Microservice
- Antarmuka Pengguna
- Database
- Server
- Pull dan Push Model

# DESKRIPSI APLIKASI

Pembangunan Aplikasi Pendataan Administrasi Pelayanan Majelis se-Distrik II Silindung berbasis Website ini bertujuan untuk membantu Distrik II Silindung dalam mengolah data majelis gereja yang berada di bawah naungan HKBP Distrik II Silindung.

## User

- Sintua
- Operator Gereja
- Admin Distrik

## Bahasa Pemrograman

- Back-end menggunakan bahasa Go
- Front-end menggunakan bahasa PHP

## Tools

- VSCode
- DBeaver
- Postman
- Browser

# FUNGSI PADA APLIKASI

1. Fungsi Login

2. Fungsi Menambahkan Informasi

3. Fungsi Mengedit Informasi

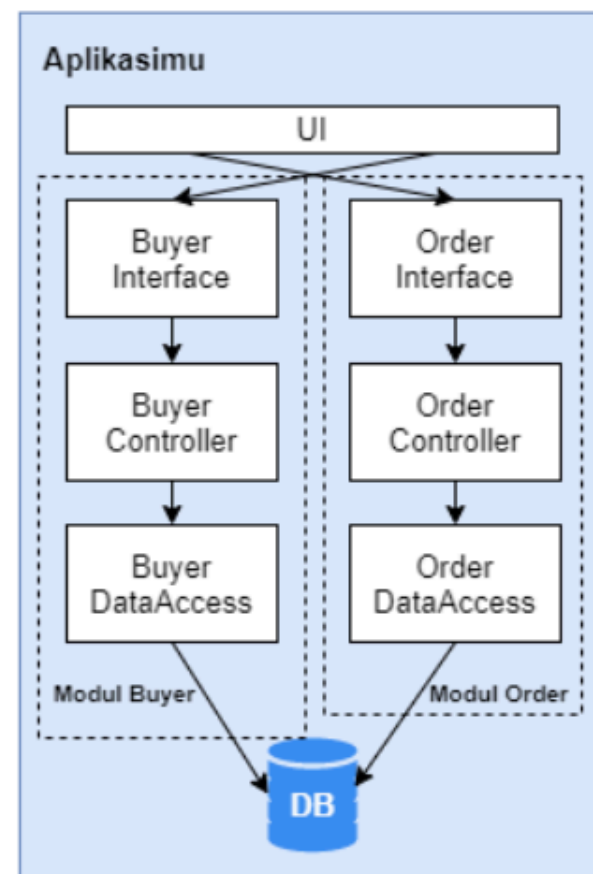
4. Fungsi Menghapus Informasi

5. Fungsi Melihat Detail Informasi

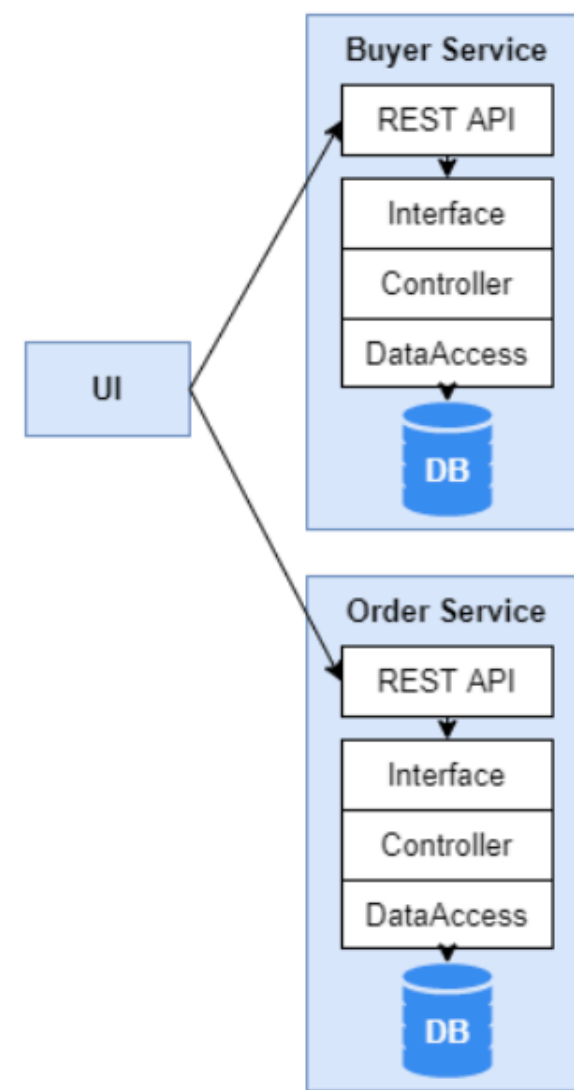
6. Fungsi Logout

# PERBEDAAN MONOLITH DAN MICROSERVICE

## MONOLITH



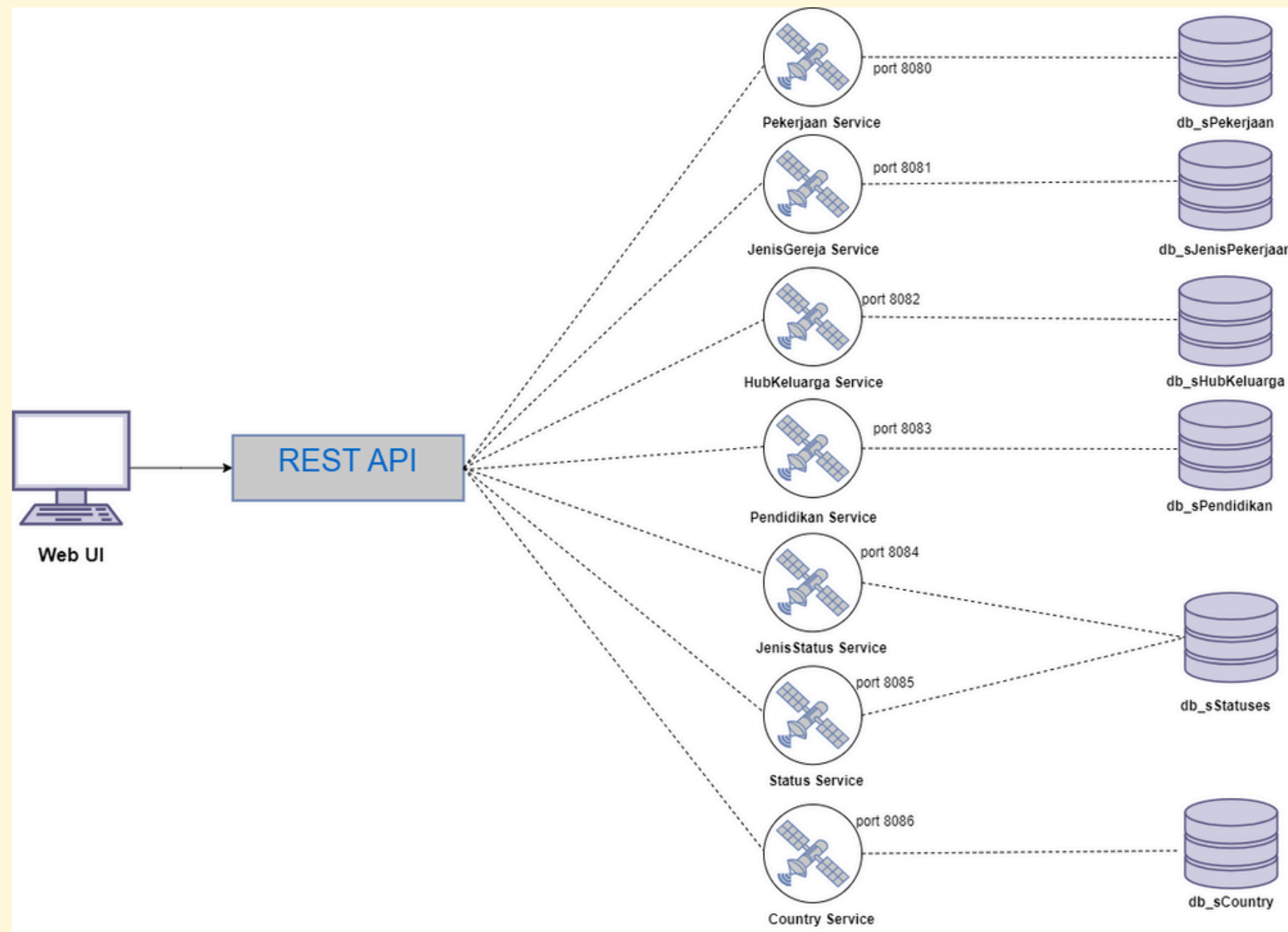
## MICROSERVICES



Arsitektur monolitik adalah arsitektur dimana keseluruhan kode akan dikompilasi menjadi satu aplikasi (biasanya menjadi satu binary atau artifact) dimana aplikasi tersebut menjalankan seluruh proses yang dibutuhkan. Kebutuhan akan komunikasi kepada aplikasi atau service lain bisa jadi tidak ada, karena aplikasi ini telah mencakup seluruh kode yang dibutuhkan

Arsitektur microservice adalah arsitektur yang menekankan kepada banyak service atau aplikasi yang tidak terikat erat satu dengan lainnya (loosely coupled). Service ini memiliki tujuan yang sangat terfokus dan memiliki metode komunikasi antar service yang tidak berat

# MICROSERVICE



Service yang tersedia adalah Pekerjaan Service dengan database `db_sPekerjaan`, JenisGereja Service dengan database `db_sJenisGereja`, HubKeluarga Service dengan database `db_sHubKeluarga`, Pendidikan Service dengan database `db_sPendidikan`, JenisStatus Service dan Status Service dengan database `db_sStatuses`, dan Country Service dengan database `db_sCountry`.

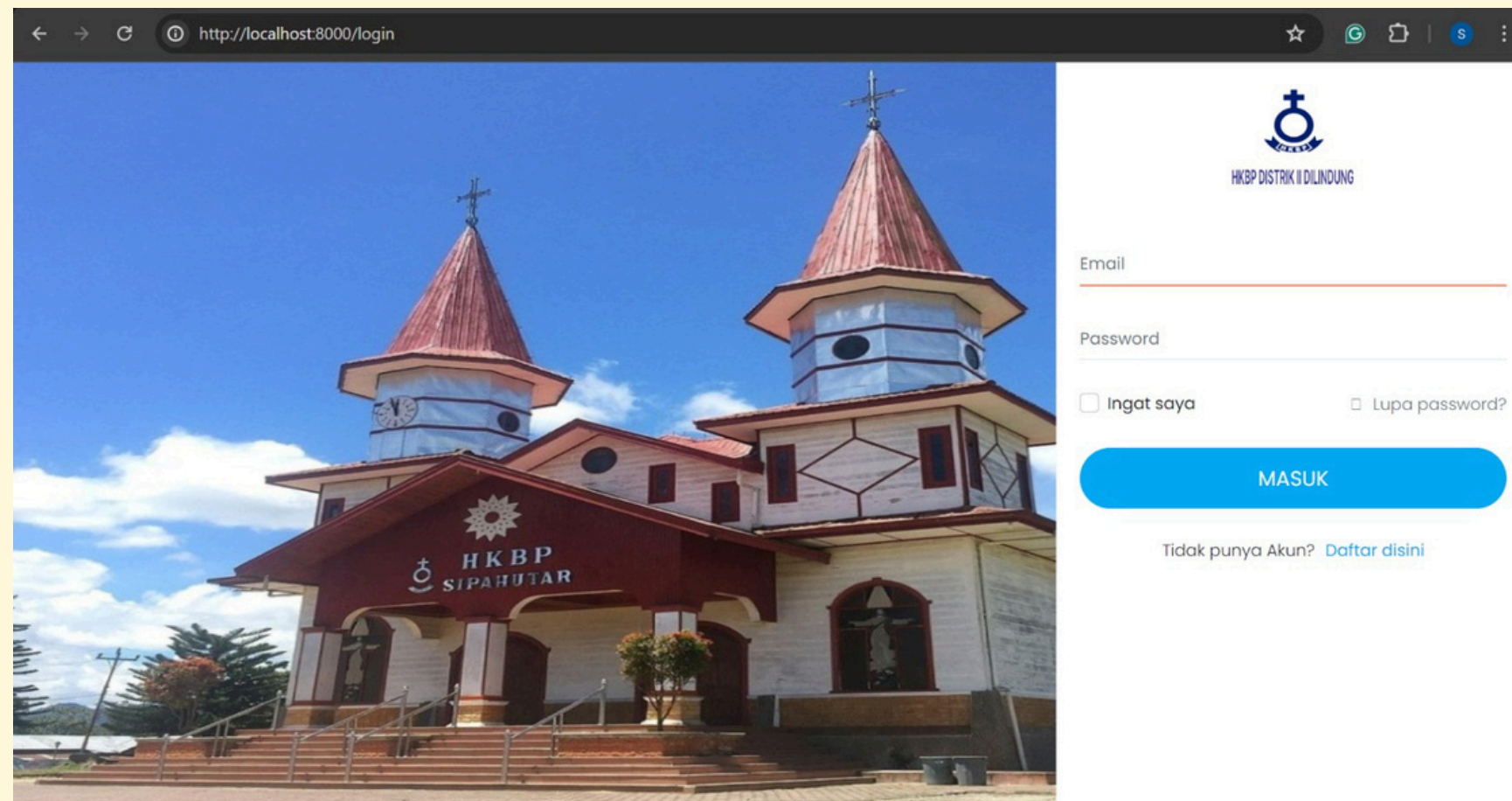
Setiap service akan berjalan secara tunggal sehingga satu service tidak akan memengaruhi service lainnya. Namun, meskipun berdiri sendiri, setiap service ini tetap saling terhubung antara satu dengan yang lainnya. Antarservice ini akan saling berkomunikasi dengan HTTP Request berstandar REST API. Setiap database pada masing-masing service diatur untuk menggunakan port yang berbeda. Ini bertujuan agar terdapat pemisahan yang jelas antara setiap layanan. Dengan penggunaan port yang berbeda, jika salah satu service mengalami masalah dan tidak dapat beroperasi, service lainnya tetap dapat berjalan tanpa terganggu.



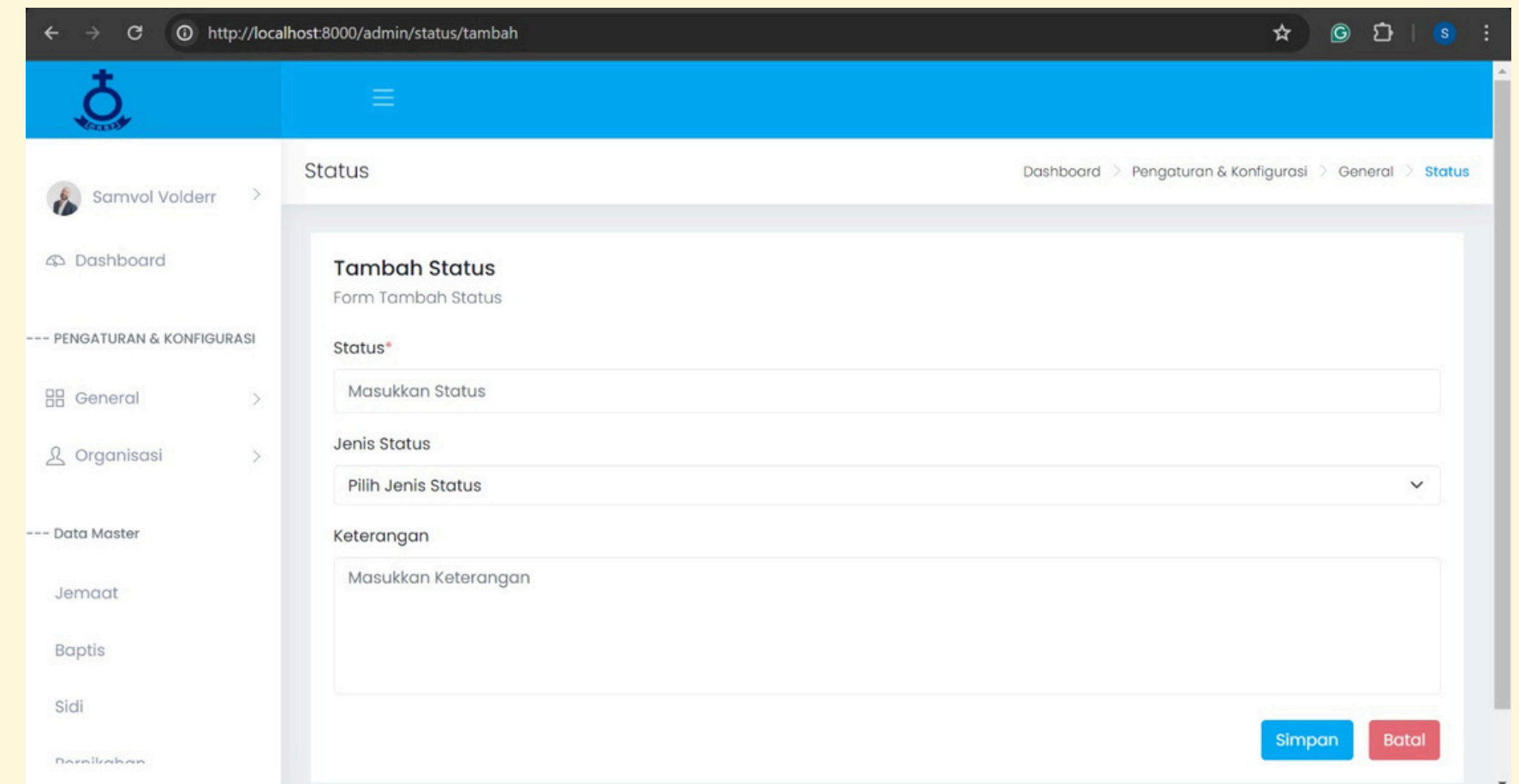
# ANTARMUKA PENGGUNA

Antarmuka pengguna (UI) adalah bagian dari aplikasi yang berinteraksi langsung dengan pengguna. Ini mencakup elemen seperti tombol, formulir, dan komponen visual lainnya. Dalam arsitektur mikroservis:

Setiap mikroservis dapat bertanggung jawab untuk merender bagian spesifik dari antarmuka pengguna. Ini memungkinkan pengembangan UI yang lebih modular dan skalabel. Tim yang berbeda dapat bekerja pada bagian-bagian UI yang berbeda secara bersamaan tanpa saling mengganggu.

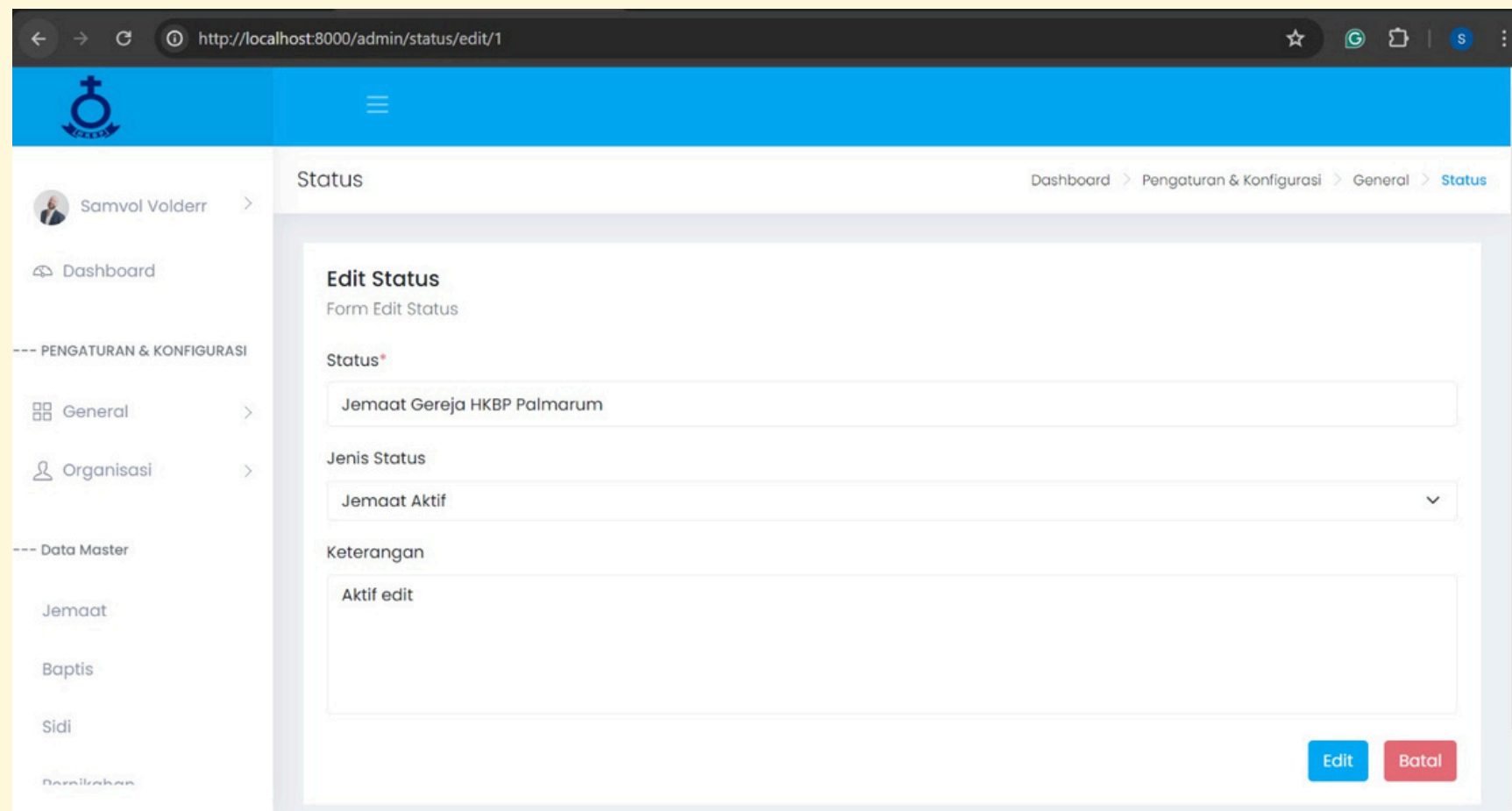


Gambar 1. Halaman Login

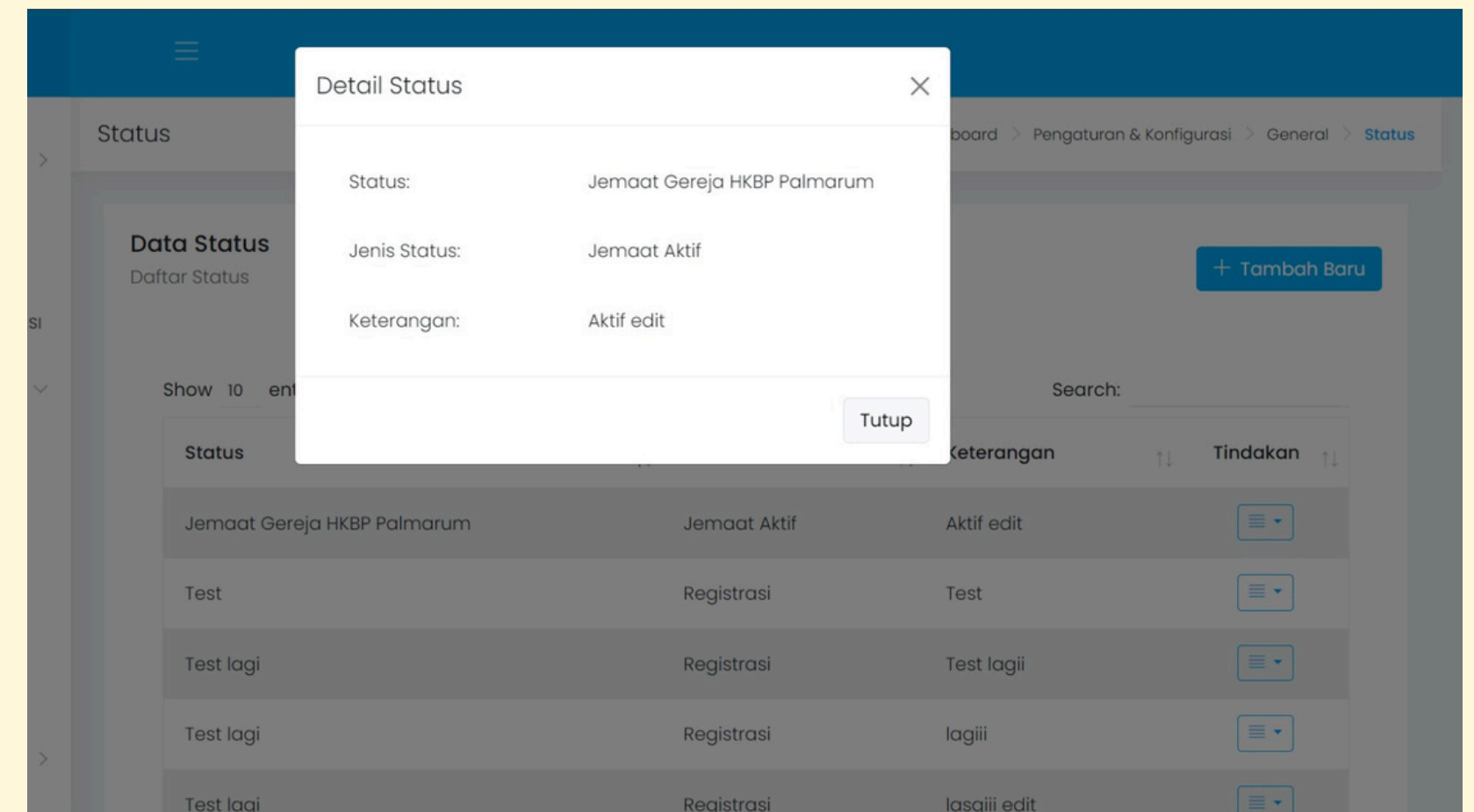


Gambar 2. Halaman Tambah Status

# ANTARMUKA PENGGUNA



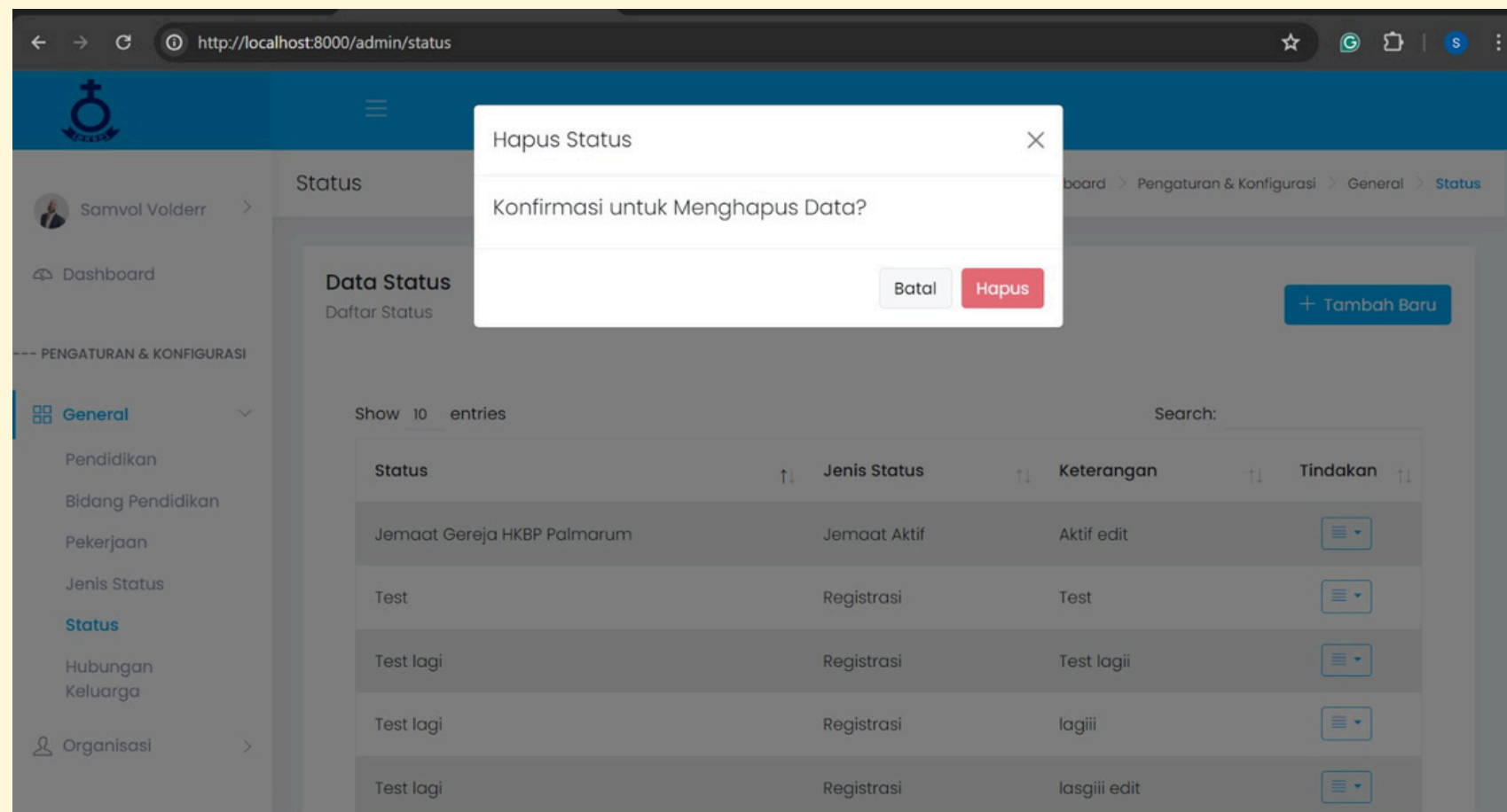
Gambar 3. Halaman Edit Status



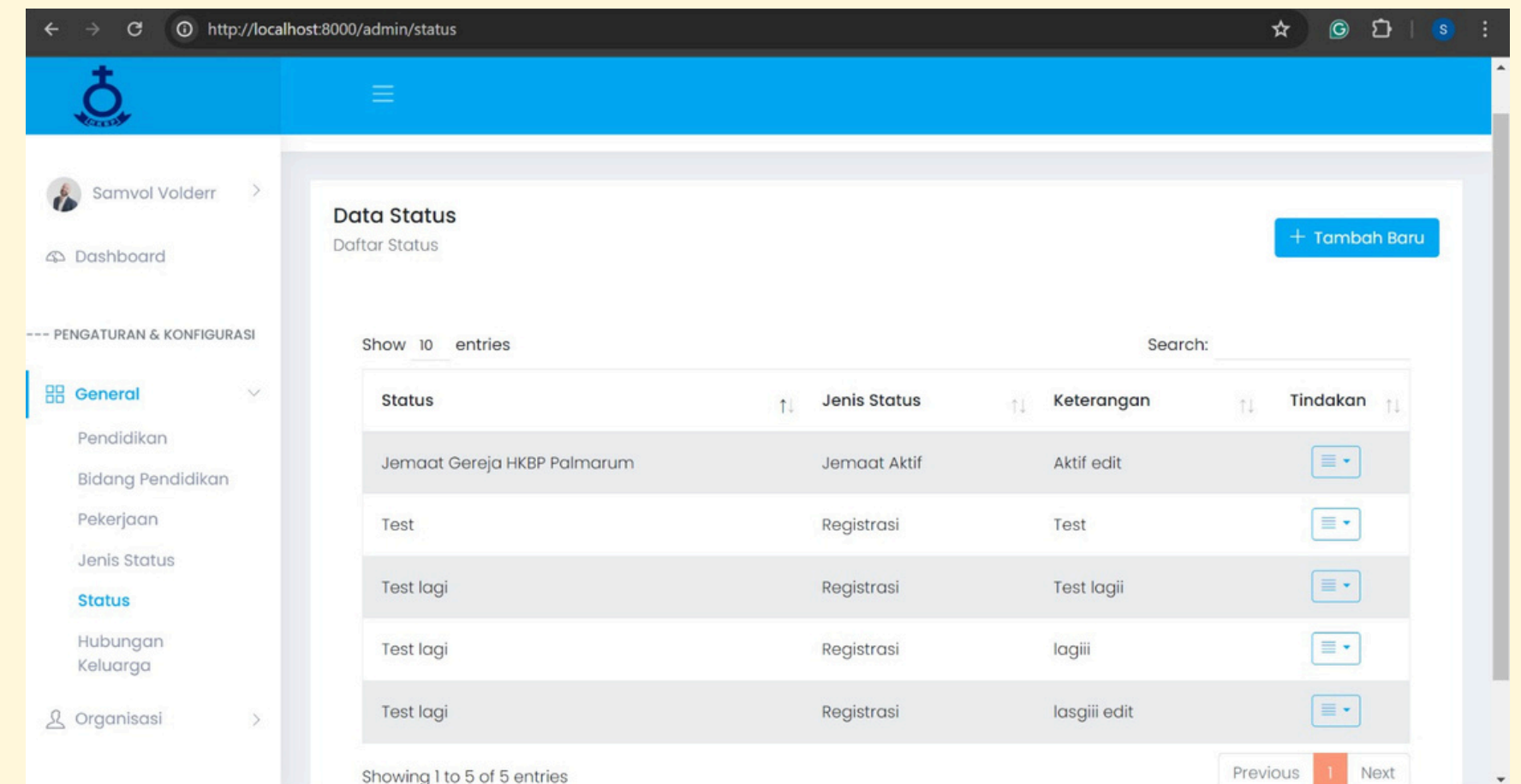
Gambar 4. Halaman Detail Status



# ANTARMUKA PENGGUNA



Gambar 5. Halaman Hapus Status



Gambar 6. Halaman Index Status

[illegible]

# SERVER

Service Pekerjaan diakses melalui localhost:8080/pekerjaan

Service JenisGereja diakses melalui localhost:8081/jenisgereja

Service HubKeluarga diakses melalui localhost:8082/hubkeluarga

Service Pendidikan diakses melalui localhost:8083/pendidikan

Service JenisStatus diakses melalui localhost:8084/statuses

Service Status diakses melalui localhost:8085/statuses

Service Country diakses melalui localhost:8086/country

# PULL & PUSH MODEL

Pull model adalah ketika sebuah klien atau layanan meminta(menarik) data dari layanan lain hanya saat dibutuhkan. Contohnya klien(aplikasi) mengirimkan permintaan GET ke `localhost:8080/pekerjaan` ketika ingin menampilkan daftar pekerjaan. Server mengirimkan respons yang berisi daftar pekerjaan yang tersedia kembali kepada klien. Saat klien membuat permintaan ke server untuk mendapatkan data yang ingin diedit, setelah menerima data, klien memodifikasinya dan mengirimkan permintaan edit ke server. Server akan memperbarui data sesuai dengan permintaan klien.

Push model adalah ketika sebuah layanan mengirimkan(mendorong) data atau notifikasi ke layanan lain atau klien secara proaktif saat ada perubahan atau pembaruan data. Contohnya server menyiapkan koneksi SSE dan mengirimkan pembaruan ke klien saat ada perubahan dalam data pekerjaan. Aplikasi klien akan membuat koneksi SSE ke `localhost:8080/pekerjaan/events`. Setiap kali ada perubahan dalam data pekerjaan, server mengirimkan pembaruan melalui koneksi SSE. Aplikasi klien menerima pembaruan dan mengupdate tampilan dengan data yang baru.



TERIMA KASIH