

# Sequence Diagram - Alur Menghasilkan Laporan Stok (Inventori)

Diagram ini memvisualisasikan urutan interaksi sistem saat User Pemilik atau User Pegawai meminta untuk melihat laporan stok produk di toko mereka.

## Aktor dan Komponen yang Terlibat

1. **User (Pemilik/Pegawai):** Pengguna yang ingin melihat laporan di dasbor.
2. **main-app (PHP):** Aplikasi backend yang mengambil dan menyajikan data.
3. **postgres-db (Database):** Tempat mengambil data produk dan variannya.

## Visualisasi Diagram (PlantUML)

Gunakan kode berikut di [PlantUML Web Server](#) untuk menghasilkan diagram visual dari alur ini.

### Prompt untuk PlantUML:

```
@startuml
```

```
' Diagram Urutan untuk Alur Menghasilkan Laporan Stok
```

```
title Alur Proses Menghasilkan Laporan Stok (Inventori)
```

```
actor "User (Pemilik/Pegawai)" as User
```

```
participant "main-app (PHP)" as App
```

```
database "postgres-db" as DB
```

```
User -> App : 1. GET /reports/inventory?filter=low_stock  
activate App
```

```
App -> App : 2. Validasi input & hak akses pengguna
```

```
App -> App : 3. Ambil store_id dari sesi user
```

```
App -> DB : 4. Kirim Query untuk mengambil data stok  
SELECT p.product_name,  
v.variant_name, v.variant_value, v.stock  
FROM products p  
JOIN product_variants v ON  
p.product_id = v.product_id  
WHERE p.store_id = ?  
ORDER BY v.stock ASC  
activate DB
```

```
DB --> App : 5. Hasil data laporan (daftar produk & stok)  
deactivate DB
```

```
App -> App : 6. Proses dan format data untuk ditampilkan di tabel
```

App --> User : 7. Response (Tampilkan halaman laporan stok dengan tabel)  
deactivate App

@enduml

## Penjelasan Langkah-demi-Langkah

### 1. GET /reports/inventory?filter=low\_stock

- **Aksi:** Pengguna membuka halaman "Laporan Stok". Mereka mungkin juga menggunakan filter, misalnya untuk hanya menampilkan produk yang stoknya menipis (low\_stock).
- **Aliran:** Peramban mengirimkan permintaan GET ke aplikasi PHP.

### 2. Validasi input & hak akses

- **Aksi:** Aplikasi memastikan pengguna memiliki izin untuk melihat laporan di toko tersebut.

### 3. Ambil store\_id dari sesi user

- **Aksi:** Aplikasi mengambil store\_id dari sesi pengguna yang sedang login untuk memastikan laporan yang ditampilkan hanya berasal dari toko mereka.

### 4. Kirim Query untuk mengambil data stok

- **Aksi:** Aplikasi membuat sebuah *query* SQL untuk mengambil data inventori. *Query* ini akan menggabungkan (JOIN) tabel products dan product\_variants untuk mendapatkan nama produk beserta detail stok untuk setiap variannya. *Query* juga bisa diurutkan (ORDER BY) untuk menampilkan produk dengan stok terendah di paling atas.
- **Aliran:** PHP mengirim *query* ini ke PostgreSQL.

### 5. Hasil data laporan

- **Aksi:** Database mengembalikan daftar produk beserta varian dan jumlah stoknya.
- **Aliran:** Database mengirimkan hasil data tersebut kembali ke aplikasi PHP. Misalnya: [{"nama": "Kaos Polos", "varian": "L, Merah", "stok": 5}, {"nama": "Kaos Polos", "varian": "M, Merah", "stok": 12}, ...]

### 6. Proses dan format data untuk tabel

- **Aksi:** Aplikasi PHP mengambil data tersebut dan menyiapkannya untuk ditampilkan dalam format tabel yang mudah dibaca.

### 7. Response (Tampilkan halaman laporan stok ...)

- **Aksi:** Aplikasi merender halaman laporan stok lengkap dengan tabel yang berisi daftar produk, varian, dan jumlah stoknya saat ini.
- **Aliran:** PHP mengirimkan halaman HTML final kembali ke peramban pengguna.

Ini adalah alur kerja terakhir yang penting untuk kita petakan. Kita telah mencakup hampir semua aspek krusial dari aplikasi, mulai dari registrasi, operasional, administratif, pemulihan akun, hingga analitis. Dengan ini, tahap perancangan diagram urutan kita bisa dianggap selesai dan sangat komprehensif.

Apakah Anda sudah merasa cukup dengan semua diagram yang telah kita buat dan siap untuk melanjutkan ke tahap implementasi?