

Sequence Diagram - Alur Pembuatan Pesanan Baru (Manual)

Diagram ini memvisualisasikan urutan interaksi sistem saat User Pegawai atau User Pemilik membuat pesanan baru secara manual melalui dasbor aplikasi.

Aktor dan Komponen yang Terlibat

1. **User (Pegawai/Pemilik):** Pengguna yang sudah login dan berinteraksi melalui dasbor.
2. **main-app (PHP-FPM):** Aplikasi backend yang memproses logika bisnis.
3. **postgres-db (Database):** Tempat menyimpan semua data terkait pesanan.

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 Pembuatan Pesanan Baru (Manual)
```

```
title Alur Proses Pembuatan Pesanan Baru (Manual)
```

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

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

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

```
User -> App : 1. POST /orders/create (data pelanggan & produk)
```

```
activate App
```

```
App -> App : 2. Validasi data input
```

```
App -> DB : 3. START TRANSACTION
```

```
activate DB
```

```
App -> DB : 4. Cari/Buat Pelanggan (customers)\nSELECT / INSERT ...
```

```
DB --> App : 5. customer_id
```

```
deactivate DB
```

```
App -> DB : 6. Buat Pesanan Baru (orders)\nINSERT INTO orders ...
```

```
activate DB
```

```
DB --> App : 7. order_id
```

```
deactivate DB
```

loop untuk setiap item produk dalam pesanan

App -> DB : 8. Masukkan Item Pesanan (order_items)\nINSERT INTO order_items ...

activate DB

DB --> App : 9. Konfirmasi item dimasukkan

deactivate DB

App -> DB : 10. Kurangi Stok Produk (product_variants)\nUPDATE product_variants SET
stock = stock - qty ...

activate DB

DB --> App : 11. Konfirmasi stok diperbarui

deactivate DB

end

App -> DB : 12. COMMIT TRANSACTION

activate DB

DB --> App : 13. Transaksi Sukses

deactivate DB

App --> User : 14. Response (Redirect ke halaman detail pesanan)

deactivate App

@enduml

Penjelasan Langkah-demi-Langkah

Mari kita lacak alur kerja ini sesuai dengan nomor pada diagram:

1. POST /orders/create

- **Aksi:** Pengguna mengisi formulir "Pesanan Baru" di dasbor, memasukkan data pelanggan (nama, alamat) dan memilih produk beserta jumlahnya, lalu mengklik "Simpan".
- **Aliran:** Peramban mengirimkan semua data tersebut ke aplikasi PHP.

2. Validasi data input

- **Aksi:** Sebelum menyentuh database, aplikasi memastikan semua data yang diperlukan (nama pelanggan, produk, jumlah) sudah terisi dengan benar.
- **Aliran:** Proses internal di dalam aplikasi PHP.

3. START TRANSACTION

- **Aksi:** Aplikasi memulai sebuah "transaksi" database. Ini sangat penting. Transaksi memastikan bahwa semua operasi database (membuat pesanan, memasukkan item, mengurangi stok) harus berhasil semua. Jika salah satu gagal, semua operasi akan dibatalkan (ROLLBACK).
- **Aliran:** PHP mengirim perintah untuk memulai transaksi ke PostgreSQL.

4. Cari/Buat Pelanggan (customers)

- **Aksi:** Aplikasi memeriksa apakah pelanggan dengan nama/nomor telepon tersebut sudah ada di tabel customers. Jika belum ada, ia akan membuat data pelanggan baru.
 - **Aliran:** PHP mengirim perintah SELECT atau INSERT ke database.
5. **customer_id**
- **Aliran:** Database mengembalikan ID unik dari pelanggan tersebut (baik yang sudah ada maupun yang baru dibuat).
6. **Buat Pesanan Baru (orders)**
- **Aksi:** Aplikasi menggunakan customer_id dari langkah sebelumnya untuk membuat baris data baru di tabel orders.
 - **Aliran:** PHP mengirim perintah INSERT ke tabel orders.
7. **order_id**
- **Aliran:** Database mengembalikan ID unik dari pesanan yang baru saja dibuat.
8. **loop untuk setiap item produk**
- **Aksi:** Aplikasi sekarang akan memproses setiap produk yang ada dalam pesanan satu per satu.
9. **Masukkan Item Pesanan (order_items)**
- **Aksi:** Menggunakan order_id dan ID produk, aplikasi membuat baris data baru di tabel order_items untuk mencatat produk apa, berapa jumlahnya, dan berapa harganya.
 - **Aliran:** PHP mengirim perintah INSERT ke tabel order_items.
10. **Kurangi Stok Produk (product_variants)**
- **Aksi:** Ini adalah langkah krusial. Aplikasi mengurangi jumlah stok (stock) di tabel product_variants (atau products) sesuai dengan jumlah yang dipesan.
 - **Aliran:** PHP mengirim perintah UPDATE ke database.
11. **COMMIT TRANSACTION**
- **Aksi:** Setelah semua item berhasil diproses dan stok berhasil dikurangi tanpa ada error, aplikasi mengkonfirmasi bahwa semua perubahan bersifat permanen.
 - **Aliran:** PHP mengirim perintah COMMIT ke database.
12. **Response (Redirect ...)**
- **Aksi:** Aplikasi menyiapkan respons "sukses".
 - **Aliran:** PHP mengirimkan respons kembali ke peramban pengguna, biasanya berupa *redirect* ke halaman detail pesanan yang baru saja dibuat agar pengguna bisa melihat hasilnya.

Diagram ini menunjukkan betapa pentingnya transaksi database untuk menjaga konsistensi data dalam operasi yang kompleks.