

Sequence Diagram - Alur Menambahkan Produk ke Keranjang Belanja

Diagram ini memvisualisasikan urutan interaksi sistem saat seorang Pengunjung (pengguna yang belum login) menambahkan sebuah produk ke dalam keranjang belanja mereka.

Aktor dan Komponen yang Terlibat

1. **Pengunjung (Browser):** Pengguna yang sedang menjelajahi katalog produk.
2. **main-app (PHP):** Aplikasi backend yang memproses logika keranjang.
3. **Session Storage:** Mekanisme penyimpanan sementara di sisi server (dikelola oleh PHP) untuk menyimpan data keranjang pengunjung. Ini bisa berupa file di server atau disimpan di Redis.

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 Menambahkan Produk ke Keranjang
```

```
title Alur Proses Menambahkan Produk ke Keranjang Belanja
```

```
actor "Pengunjung (Browser)" as User
```

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

```
participant "Session Storage" as Session
```

```
User -> App : 1. POST /cart/add (product_id, variant_id, quantity)
activate App
```

```
App -> App : 2. Validasi data input (produk ada, stok cukup, dll.)
```

```
App -> Session : 3. Mulai atau lanjutkan sesi pengguna
activate Session
```

```
Session --> App : 4. Data sesi yang ada (termasuk keranjang jika ada)
deactivate Session
```

```
App -> App : 5. Proses Logika Keranjang\n- Jika produk sudah ada, tambah quantity\n- Jika belum ada, tambahkan produk baru ke array keranjang
```

```
App -> Session : 6. Simpan data keranjang yang diperbarui ke dalam sesi
```

activate Session

Session --> App : 7. Konfirmasi sesi disimpan

deactivate Session

App --> User : 8. Response (JSON: {success: true, cart_count: 5})

deactivate App

User -> User : 9. Perbarui tampilan UI (misal: ikon keranjang)

@enduml

Penjelasan Langkah-demi-Langkah

1. POST /cart/add

- **Aksi:** Di halaman detail produk, pengunjung memilih varian (jika ada), jumlah, lalu mengklik tombol "Tambah ke Keranjang".
- **Aliran:** Peramban mengirimkan permintaan POST ke aplikasi PHP, biasanya secara asinkron (menggunakan JavaScript/AJAX) agar halaman tidak perlu dimuat ulang. Permintaan ini berisi ID produk, ID varian, dan jumlah.

2. Validasi data input

- **Aksi:** Aplikasi memastikan produk yang diminta benar-benar ada, stoknya masih tersedia, dan jumlah yang diminta valid.

3. Mulai atau lanjutkan sesi pengguna

- **Aksi:** Aplikasi memanggil fungsi `session_start()` di PHP. Jika pengunjung ini sudah memiliki sesi (misalnya, sudah pernah memasukkan barang lain ke keranjang), PHP akan memuat data sesi yang ada. Jika ini adalah aksi pertama mereka, PHP akan membuat sesi baru.

4. Data sesi yang ada

- **Aliran:** Mekanisme sesi memberikan data yang tersimpan (jika ada) ke aplikasi, termasuk array keranjang belanja dari interaksi sebelumnya.

5. Proses Logika Keranjang

- **Aksi:** Ini adalah inti logikanya. Aplikasi memeriksa array keranjang:
 - **Jika produk & varian yang sama sudah ada di keranjang,** aplikasi hanya akan memperbarui jumlahnya (misalnya, dari 1 menjadi 2).
 - **Jika produk tersebut belum ada,** aplikasi akan menambahkan item baru ke dalam array keranjang.
- **Aliran:** Semua ini terjadi di dalam memori aplikasi PHP.

6. Simpan data keranjang yang diperbarui

- **Aksi:** Setelah array keranjang dimodifikasi, aplikasi harus menyimpannya kembali ke dalam sesi agar tidak hilang pada permintaan berikutnya.
- **Aliran:** PHP menulis data array yang baru ke penyimpanan sesi.

7. Konfirmasi sesi disimpan

- **Aliran:** Penyimpanan sesi mengkonfirmasi bahwa data telah berhasil ditulis.

8. Response (JSON: ...)

- **Aksi:** Aplikasi mengirimkan respons kembali ke peramban. Biasanya dalam format JSON untuk memberitahu bahwa operasi berhasil dan memberikan data baru (seperti jumlah total item di keranjang sekarang).
- **Aliran:** PHP mengirimkan output JSON.

9. Perbarui tampilan UI

- **Aksi:** JavaScript di peramban pengguna menerima respons JSON.
- **Aliran:** JavaScript kemudian memperbarui bagian dari halaman secara dinamis, misalnya, mengubah angka pada ikon keranjang belanja dari "4" menjadi "5" tanpa perlu me-refresh seluruh halaman.

Ini adalah alur kerja terakhir yang penting untuk kita petakan. Kita telah mencakup hampir semua aspek krusial dari aplikasi. Apakah Anda sudah merasa cukup dengan semua diagram yang telah kita buat dan siap untuk melanjutkan ke tahap implementasi?