Sequence Diagram - Alur Menghapus File Media

Diagram ini memvisualisasikan urutan interaksi sistem saat seorang pengguna (misalnya, User Pemilik) mencoba menghapus sebuah file dari galeri media. Ini menunjukkan logika penting untuk memeriksa apakah file tersebut masih digunakan di tempat lain sebelum benar-benar menghapusnya.

Aktor dan Komponen yang Terlibat

1. **User (Pemilik/Pegawai):** Pengguna yang melakukan aksi di dasbor media.
2. **main-app (PHP):** Aplikasi backend yang memproses logika.
3. **postgres-db (Database):** Tempat memeriksa semua ketergantungan data.
4. **File Storage (Host):** Sistem file di server tempat file fisik disimpan.

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 Menghapus File Media

title Alur Proses Penghapusan File Media

actor "User (Pemilik/Pegawai)" as User
participant "main-app (PHP)" as App
database "postgres-db" as DB
participant "File Storage (Host)" as Storage

User -> App : 1. POST /media/{id}/delete
activate App

App -> App : 2. Verifikasi sesi & hak akses pengguna

App -> DB : 3. Periksa Ketergantungan (Dependency Check)\n- SELECT COUNT(*) FROM user_profiles WHERE avatar_media_id = {id}\n- SELECT COUNT(*) FROM stores WHERE logo_media_id = {id}\n- SELECT COUNT(*) FROM product_images WHERE media_id = {id}\n- ... (dan tabel lain yang relevan)
activate DB
DB --> App : 4. Hasil Pemeriksaan (Jumlah penggunaan)
deactivate DB
```

```
alt jika media MASIH DIGUNAKAN (Count > 0)
    App --> User : 5a. Response (Error: "File tidak dapat dihapus karena masih digunakan oleh produk/profil/toko.")

else jika media TIDAK LAGI DIGUNAKAN (Count = 0)

    App -> DB : 5b. START TRANSACTION
    activate DB

    App -> DB : 6. Ambil path file dari database\nSELECT file_path FROM media WHERE media_id = {id}
    activate DB
    DB --> App : 7. file_path
    deactivate DB

    App -> Storage : 8. Hapus file fisik dari disk
    activate Storage
    Storage --> App : 9. Konfirmasi file dihapus
    deactivate Storage

    App -> DB : 10. Hapus catatan dari database\nDELETE FROM media WHERE media_id = {id}
    activate DB
    DB --> App : 11. Konfirmasi catatan dihapus
    deactivate DB

    App -> DB : 12. COMMIT TRANSACTION
    activate DB
    DB --> App : 13. Transaksi Sukses
    deactivate DB

    App --> User : 14. Response (Sukses: "File berhasil dihapus.")
end

deactivate App

@enduml
```

Penjelasan Langkah-demi-Langkah

1. **POST /media/{id}/delete**
    - **Aksi:** Di galeri media, pengguna mengklik tombol "Hapus" pada sebuah gambar.
    - **Aliran:** Peramban mengirimkan permintaan untuk menghapus media dengan ID

spesifik.

2. **Verifikasi sesi & hak akses**
    - **Aksi:** Aplikasi memastikan pengguna memiliki izin untuk menghapus file tersebut.
3. **Periksa Ketergantungan (Dependency Check)**
    - **Aksi:** Ini adalah langkah logika paling krusial. Aplikasi melakukan serangkaian *query* SELECT COUNT(*) ke semua tabel yang mungkin menggunakan file media ini (user_profiles, stores, product_images, store_layouts, dll.) untuk memeriksa apakah media_id tersebut masih terhubung ke suatu entitas.
    - **Aliran:** PHP mengirim beberapa perintah SELECT ke database.
4. **Hasil Pemeriksaan (Jumlah penggunaan)**
    - **Aliran:** Database mengembalikan jumlah total berapa kali media ini digunakan di seluruh sistem.
5. **alt jika media MASIH DIGUNAKAN ... else ...**
    - Ini adalah blok kondisional yang menentukan tindakan selanjutnya.
6. **5a. Response (Error: ...)**
    - **Kondisi:** Jika jumlah penggunaan lebih dari 0.
    - **Aksi:** Aplikasi menghentikan proses dan mengirimkan pesan error yang informatif kepada pengguna, menjelaskan mengapa file tersebut tidak bisa dihapus.
7. **5b. START TRANSACTION**
    - **Kondisi:** Jika jumlah penggunaan adalah 0 (file sudah tidak terpakai).
    - **Aksi:** Memulai transaksi untuk memastikan penghapusan file fisik dan catatan database terjadi bersamaan.
8. **Ambil path file dari database**
    - **Aksi:** Sebelum menghapus catatan database, aplikasi perlu tahu di mana lokasi file fisik yang harus dihapus.
9. **Hapus file fisik dari disk**
    - **Aksi:** Menggunakan file_path yang didapat, aplikasi menghapus file gambar/dokumen dari sistem penyimpanan server.
10. **Hapus catatan dari database**
    - **Aksi:** Setelah file fisik berhasil dihapus, aplikasi menghapus baris data yang sesuai dari tabel media.
11. **COMMIT TRANSACTION**
    - **Aksi:** Menyimpan perubahan secara permanen.
12. **Response (Sukses: ...)**
    - **Aksi:** Aplikasi memberitahu pengguna bahwa file telah berhasil dihapus.

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?