

Sequence Diagram - Alur Pengguna Memperbarui Profil Sendiri

Diagram ini memvisualisasikan urutan interaksi sistem saat seorang pengguna yang sudah login memperbarui informasi profil mereka (seperti nama, bio, atau foto profil) melalui halaman pengaturan akun.

Aktor dan Komponen yang Terlibat

1. **User (Sudah Login):** Pengguna yang melakukan perubahan di dasbor.
2. **main-app (PHP):** Aplikasi backend yang memproses logika.
3. **File Storage (Host):** Sistem file di server tempat avatar baru disimpan.
4. **postgres-db (Database):** Tempat memperbarui data profil dan media.

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 Pengguna Memperbarui Profil Sendiri
```

```
title Alur Proses Pengguna Memperbarui Profil Sendiri
```

```
actor "User (Sudah Login)" as User
```

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

```
participant "File Storage (Host)" as Storage
```

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

```
User -> App : 1. POST /account/profile/update (full_name, bio, file_avatar opsional)  
activate App
```

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

```
App -> App : 3. Ambil user_id dari sesi pengguna
```

```
opt jika ada file avatar baru yang diunggah
```

```
App -> Storage : 4a. Simpan file avatar baru ke disk
```

```
activate Storage
```

```
Storage --> App : 5a. Path file tersimpan
```

```
deactivate Storage
```

```
App -> DB : 6a. Masukkan data avatar baru (media)\nINSERT INTO media ...
```

activate DB
DB --> App : 7a. avatar_media_id baru
deactivate DB
end

App -> DB : 8. Perbarui data profil (user_profiles)\nUPDATE user_profiles SET full_name=?,
bio=?, avatar_media_id=? WHERE user_id=?
activate DB
DB --> App : 9. Konfirmasi profil diperbarui
deactivate DB

App --> User : 10. Response (Sukses: "Profil berhasil diperbarui.")
deactivate App

@enduml

Penjelasan Langkah-demi-Langkah

1. **POST /account/profile/update**
 - **Aksi:** Di halaman pengaturan profil, pengguna mengubah nama, bio, dan mungkin memilih untuk mengunggah foto profil baru, lalu mengklik "Simpan".
 - **Aliran:** Peramban mengirimkan data teks dan file (jika ada) ke aplikasi PHP.
2. **Validasi data input**
 - **Aksi:** Aplikasi memastikan data yang diterima valid (misalnya, nama tidak kosong).
3. **Ambil user_id dari sesi pengguna**
 - **Aksi:** Aplikasi mengidentifikasi pengguna mana yang sedang melakukan perubahan ini berdasarkan data sesi yang aktif.
4. **opt jika ada file avatar baru ...**
 - Ini adalah blok opsional. Langkah 4a hingga 7a hanya akan dijalankan jika pengguna benar-benar mengunggah file avatar baru.
5. **4a. Simpan file avatar baru ke disk**
 - **Aksi:** Aplikasi memproses file gambar yang diunggah dan menyimpannya ke lokasi permanen di server.
6. **6a. Masukkan data avatar baru (media)**
 - **Aksi:** Aplikasi membuat catatan baru di tabel media untuk avatar yang baru diunggah.
7. **7a. avatar_media_id baru**
 - **Aliran:** Database mengembalikan ID dari media yang baru saja ditambahkan. Aplikasi akan menggunakan ID ini saat memperbarui profil.
8. **Perbarui data profil (user_profiles)**
 - **Aksi:** Aplikasi memperbarui satu baris di tabel user_profiles yang terkait dengan user_id pengguna. Ia akan memperbarui full_name, bio, dan jika ada, kolom avatar_media_id dengan ID yang baru dari langkah #7a.

- **Aliran:** PHP mengirim perintah UPDATE ke tabel user_profiles.

9. Konfirmasi profil diperbarui

- **Aliran:** Database mengkonfirmasi bahwa pembaruan berhasil.

10. Response (Sukses: ...)

- **Aksi:** Aplikasi mengirimkan respons kembali ke peramban pengguna, memberitahukan bahwa profil mereka telah berhasil diperbarui.

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?