

# Sequence Diagram - Alur Registrasi User Pemilik Baru

Diagram ini memvisualisasikan urutan interaksi sistem saat seorang calon pemilik bisnis mendaftarkan akun baru. Alur ini kompleks karena melibatkan pembuatan beberapa catatan di tabel yang berbeda secara bersamaan dalam satu transaksi.

## Aktor dan Komponen yang Terlibat

1. **User (Calon Pemilik):** Pengguna yang mengisi formulir registrasi.
2. **main-app (PHP):** Aplikasi backend yang memproses logika.
3. **postgres-db (Database):** Tempat menyimpan semua data.

## 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 Registrasi User Pemilik Baru
```

```
title Alur Proses Registrasi User Pemilik Baru
```

```
actor "User (Calon Pemilik)" as User
```

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

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

```
User -> App : 1. POST /register (nama, email, password, nama_toko)  
activate App
```

```
App -> App : 2. Validasi data (email unik, password kuat, dll.)
```

```
App -> DB : 3. Periksa keunikan email\nSELECT COUNT(*) FROM users WHERE email = [email]  
activate DB
```

```
DB --> App : 4. Hasil (Count = 0)  
deactivate DB
```

```
App -> App : 5. Hash password (password_hash)
```

```
App -> DB : 6. START TRANSACTION  
activate DB
```

App -> DB : 7. Buat Pengguna (users)\nINSERT INTO users (email, password\_hash, role='store\_owner')  
activate DB  
DB --> App : 8. user\_id  
deactivate DB

App -> DB : 9. Buat Profil Pengguna (user\_profiles)\nINSERT INTO user\_profiles (user\_id, full\_name)  
activate DB  
DB --> App : 10. Konfirmasi profil dibuat  
deactivate DB

App -> DB : 11. Buat Toko (stores)\nINSERT INTO stores (owner\_id, store\_name)  
activate DB  
DB --> App : 12. store\_id  
deactivate DB

App -> DB : 13. Buat Layout Toko Default (store\_layouts)\nINSERT INTO store\_layouts (store\_id, ...)  
activate DB  
DB --> App : 14. Konfirmasi layout dibuat  
deactivate DB

App -> DB : 15. Update user dengan store\_id\nUPDATE users SET store\_id = [store\_id] WHERE user\_id = [user\_id]  
activate DB  
DB --> App : 16. Konfirmasi user diupdate  
deactivate DB

App -> DB : 17. COMMIT TRANSACTION  
activate DB  
DB --> App : 18. Transaksi Sukses  
deactivate DB

App --> User : 19. Response (Redirect ke halaman login atau dasbor)  
deactivate App

@enduml

## Penjelasan Langkah-demi-Langkah

### 1. POST /register

- **Aksi:** Pengguna baru mengisi formulir registrasi dengan nama, email, password,

dan nama toko yang diinginkan.

- **Aliran:** Peramban mengirimkan data ini ke aplikasi PHP.

## 2. Validasi data input

- **Aksi:** Aplikasi memastikan semua data valid. Misalnya, email memiliki format yang benar, password memenuhi syarat kekuatan, dan nama toko tidak kosong.

## 3. Periksa keunikan email

- **Aksi:** Aplikasi harus memastikan tidak ada pengguna lain yang sudah terdaftar dengan email yang sama.
- **Aliran:** PHP mengirim *query* SELECT COUNT(\*) ke database.

## 4. Hasil (Count = 0)

- **Aliran:** Database mengkonfirmasi bahwa email tersebut belum ada. Jika sudah ada, proses akan berhenti dan menampilkan error.

## 5. Hash password

- **Aksi:** Aplikasi mengenkripsi kata sandi pengguna menggunakan algoritma *hashing* yang kuat (seperti password\_hash() di PHP) sebelum menyimpannya.

## 6. START TRANSACTION

- **Aksi:** Memulai transaksi database. Ini sangat penting karena kita akan menulis ke 4 tabel berbeda. Jika salah satu gagal, semua harus dibatalkan.

## 7. Buat Pengguna (users)

- **Aksi:** Membuat catatan baru di tabel users dengan peran 'store\_owner'.
- **Aliran:** INSERT ke tabel users.

## 8. user\_id

- **Aliran:** Database mengembalikan ID dari pengguna yang baru dibuat.

## 9. Buat Profil Pengguna (user\_profiles)

- **Aksi:** Membuat catatan di tabel user\_profiles yang terhubung dengan user\_id.

## 10. Buat Toko (stores)

- **Aksi:** Membuat catatan di tabel stores, menghubungkannya dengan owner\_id (yang merupakan user\_id dari langkah #8).

## 11. store\_id

- **Aliran:** Database mengembalikan ID dari toko yang baru dibuat.

## 12. Buat Layout Toko Default (store\_layouts)

- **Aksi:** Membuat catatan di tabel store\_layouts dengan store\_id yang baru. Aplikasi bisa mengisi beberapa nilai default untuk judul, footer, dll.

## 13. Update user dengan store\_id

- **Aksi:** Ini adalah langkah penting untuk melengkapi hubungan dua arah. Aplikasi memperbarui catatan pengguna yang dibuat di langkah #7 untuk menambahkan store\_id yang didapat di langkah #12.

## 14. COMMIT TRANSACTION

- **Aksi:** Setelah semua operasi berhasil, aplikasi menyimpan semua perubahan secara permanen.

## 15. Response (Redirect ...)

- **Aksi:** Aplikasi memberitahu pengguna bahwa registrasi berhasil, biasanya dengan mengarahkan mereka ke halaman login untuk masuk pertama kali.

Apakah ada alur kerja lain yang menurut Anda penting untuk kita visualisasikan?