TAMING TECH SDN. BHD.
321A, Tingkat 1, Lorong Selangor, Taman Melawati,
53100 Kuala Lumpur.
Tel: +60 13-446 4601 email: hq@tamingtech.my

Nota Pelajar 14: Livewire Lifecycle Hooks

Objektif

Selepas habis kelas ini, anda akan dapat:

1. Faham maksud lifecycle dalam Livewire.
2. Bezakan antara hooks utama dalam Livewire (PHP).
3. Guna hooks seperti `mount()`, `updating()`, `updated()`, `hydrate()`, `dehydrate()` dengan betul.
4. Faham bila hook dipanggil (sekali, setiap request, atau bila property berubah).
5. Apply lifecycle hooks untuk projek Job Board.

Apa itu Lifecycle Hooks?

- Setiap komponen Livewire ada kitaran hidup: mula (mount) → berinteraksi (update) → hantar balik data (dehydrate).
- Lifecycle hooks adalah “pintu masuk” untuk developer masukkan logik tambahan dalam fasa tertentu.
- Bayangkan lifecycle ini macam checkpoint dalam satu perjalanan.

Kenapa penting?

- Dengan hooks, kita boleh automasi banyak perkara:
    - Auto ambil data bila component mula.
    - Validasi atau manipulasi data sebelum update.
    - Log perubahan untuk debugging.
    - Tambah loading indicator di UI.

Hooks Utama (Server-Side)

TAMING TECH SDN. BHD.
321A, Tingkat 1, Lorong Selangor, Taman Melawati,
53100 Kuala Lumpur.
Tel: +60 13-446 4601 email: hq@tamingtech.my

1. mount()

- Dipanggil **sekali sahaja** bila komponen mula diload.
- Biasa digunakan untuk:
    - Ambil data dari DB.
    - Set default value.
    - Check authorization (boleh masuk page ke tak).

```
public function mount()
{
    $this->jobs = Job::latest()->take(10)->get();
}
```

Use case dalam Job Board: bila buka Job Listing, terus paparkan senarai job terbaru.

2. hydrate()

- Dipanggil setiap kali Livewire menghidupkan semula komponen selepas request.
- Biasanya digunakan untuk reset sesuatu state kecil.

```
public function hydrate()
{
    logger('Komponen sedang dihidupkan semula...');
}
```

Contoh: nak log berapa kali user trigger action pada komponen.

3. dehydrate()

- Dipanggil sebelum Livewire hantar response ke browser.
- Boleh guna untuk:
    - Padam data sensitif.
    - Transform data sebelum render.

```
public function dehydrate()
{
    unset($this->secretToken);
```

TAMING TECH SDN. BHD.
321A, Tingkat 1, Lorong Selangor, Taman Melawati,
53100 Kuala Lumpur.
Tel: +60 13-446 4601 email: hq@tamingtech.my

```
}
```

Contoh: pastikan token API tak “bocor” ke frontend.

4. `updating($property, $value)`

- Dipanggil **sebelum** sesuatu property dikemas kini.
- Sesuai untuk:
    - Validasi awal.
    - Bersihkan input sebelum disimpan.

```
public function updating($property, $value)
{
    logger("Property {$property} akan diupdate kepada {$value}");
}
```

Contoh: bila user update `search`, kita boleh bersihkan whitespace.

5. `updated($property, $value)`

- Dipanggil selepas property berjaya dikemas kini.
- Boleh trigger logik lain, contohnya update senarai carian.

```
public function updated($property, $value)
{
    logger("Property {$property} telah dikemas kini: {$value}");
}
```

Contoh: bila user isi “Laravel Developer” dalam search, senarai job terus di-refresh.

Kegunaan Lifecycle Hooks dalam Projek Job Board

1. Gunakan `mount()`
    - Load job listings awal-awal bila buka page.
2. Gunakan `updated()`
    - Refresh hasil carian setiap kali user ubah input search.
3. Gunakan `dehydrate()`

TAMING TECH SDN. BHD.
321A, Tingkat 1, Lorong Selangor, Taman Melawati,
53100 Kuala Lumpur.
Tel: +60 13-446 4601 email: hq@tamingtech.my

    - Pastikan data sensitif (contoh internal note HR) tak dihantar ke frontend.
4. Gunakan `hydrate()`
    - Untuk log atau reset state setiap kali request baru.

Tugasan

Dalam projek kali ini, kamu perlu siapkan beberapa fungsi utama:

1. Admin Panel

- Bina satu panel khas untuk Admin supaya Admin boleh urus sistem dengan mudah.

2. Job Application (Permohonan Kerja)

- Buat paparan (view) untuk senarai permohonan kerja.

3. Job Listing (Senarai Jawatan Kosong)

- Sediakan fungsi penuh CRUD (Create, Read, Update, Delete) supaya jawatan kosong boleh ditambah, dilihat, dikemaskini, dan dipadam.

4. User Authentication & Authorization (Log Masuk & Kebenaran Pengguna)

- Buat sistem login/logout.
- Tetapkan dua jenis role:
    - Admin – boleh urus semua fungsi dalam sistem.
    - Guest – hanya boleh lihat dan hantar permohonan kerja.

5. Policy (Dasar Akses)

- Tetapkan polisi untuk pastikan hanya pengguna dengan role tertentu boleh buat tindakan tertentu:
    - Job Listings – hanya Admin boleh urus CRUD.
    - Job Application – Guest boleh hantar permohonan, Admin boleh semak permohonan.

Kelas Seterusnya: Listing Lanjutan