

Nota Pelajar 13: Asas [Alpine.js](#) dalam Livewire

Objektif

- Faham asas Alpine.js & peranannya dalam projek Livewire.
- Belajar integrasi Alpine dalam komponen Livewire.
- Mampu **akses & kawal Livewire properties dari Alpine**.
- Faham cara **mutasi data, panggil methods, refresh & share state**.
- Tingkatkan interaktiviti projek Job Board dengan [Alpine.js](#).

Pengenalan [Alpine.js](#)

- **JavaScript framework ringan** → hanya ~10kb.
- Ditulis terus dalam HTML (inline directives).
- Guna prinsip “**Vue-like** syntax” tapi minimal.
- Sesuai untuk:
 - UI transition & animation
 - Dropdown, modal, tabs, toggle
 - Bekerja rapat dengan Livewire

Basic Alpine Component

```
<div x-data="{ open: false }">  
  <button @click="open = ! open">Toggle</button>  
  <div x-show="open">Content ditunjukkan bila open = true</div>  
</div>
```

- **x-data** → definisi state.
- **x-show** → kawal paparan.
- **@click** → event listener.

Menggunakan Alpine dalam Livewire

- Boleh campur **Blade + Livewire + Alpine**.
- Alpine urus UI, Livewire urus data dari server.

```
<div>
  <h1>{{ $post->title }}</h1>

  <div x-data="{ expanded: false }">
    <button type="button" x-on:click="expanded = ! expanded">
      <span x-show="! expanded">Show post content...</span>
      <span x-show="expanded">Hide post content...</span>
    </button>

    <div x-show="expanded">
      {{ $post->content }}
    </div>
  </div>
</div>
```

Mengawal Livewire dari Alpine (\$wire)

- ``$wire`` = akses terus kepada komponen Livewire.

```
<form wire:submit="save">
  <!-- ... -->

  <input wire:model="content" type="text">

  <small>
    Character count: <span x-text="$wire.content.length"></span>
  </small>

  <button type="submit">Save</button>
</form>
```

- Panggil method Livewire → `$wire.increment()`.
- Baca property Livewire → `$wire.count`.

Akses Livewire Properties

```
<div x-data>
  <span x-text="$wire.title"></span>
</div>
```

- `$wire.propertyName` → akses value property.

Mutasi Livewire Properties

```
<form wire:submit="save">
  <input type="text" wire:model="title">
  <button type="button" x-on:click="$wire.title = ''">Clear</button>

  <button>Save</button>
</form>
```

- Ubah property Livewire secara terus dari Alpine.

Panggil Livewire Methods

```
<form wire:submit="save">
  <input wire:model="title" type="text" x-on:blur="$wire.save()">

  <!-- ... -->

  <button type="submit">Save</button>
</form>
```

- Boleh invoke **public methods** Livewire.
- Berguna untuk trigger action dari Alpine.

Passing parameters

Boleh passkan parameters ke Livewire methods dengan passkan ke dalam `$wire` method call.

Di dalam component dengan method deletePost()

```
public function deletePost($postId)
{
    $post = Post::find($postId);

    auth()->user()->can('update', $post);

    $post->delete();
}
```

Sekarang boleh pass kan \$postId parameter ke dalam deletePost() method from Alpine like so:

```
<button type="button" x-on:click="$wire.deletePost(1)">
```

Ini sebagai contoh untuk passkan post id dengan menggunakan blade untuk Alpine passkan kepada deletePost().

```
@foreach ($posts as $post)
    <button type="button" x-on:click="$wire.deletePost({{ $post->id }})">
        Delete "{{ $post->title }}"
    </button>
@endforeach
```

Sekiranya posts ada tiga, blade template akan render kepada:

```
<button type="button" x-on:click="$wire.deletePost(1)">
    Delete "The power of walking"
</button>

<button type="button" x-on:click="$wire.deletePost(2)">
    Delete "How to record a song"
</button>

<button type="button" x-on:click="$wire.deletePost(3)">
```

```
Delete "Teach what you learn"
</button>
```

Blade Parameter Gotchas

Bila guna Blade + Alpine, kadang-kala syntax akan clash.

Contoh salah:

```
<!-- Warning: this is an example of problematic code... -->
<button
  type="button"
  x-on:click="$wire.deletePost({{ $post->uuid }})"
>
```

Contoh kod atas Blade template akan render menjadi:

```
<!-- Warning: this is an example of problematic code... -->
<button
  type="button"
  x-on:click="$wire.deletePost(93c7b04c-c9a4-4524-aa7d-39196011b81a)"
>
```

Cara betul (escape quotes):

```
<button
  type="button"
  x-on:click="$wire.deletePost('{{ $post->uuid }}')"
>
```

Contoh kod atas Blade template akan render menjadi:

```
<button
  type="button"
  x-on:click="$wire.deletePost('93c7b04c-c9a4-4524-aa7d-39196011b81a')"
>
```

Refreshing a Component

- Livewire component boleh refresh dari Alpine.

```
<button type="button" x-on:click="$wire.$refresh()">
```

- Berguna bila nak reload UI tanpa full page reload.

Sharing State dengan \$wire.entangle

- `entangle` = sync state antara Alpine & Livewire.

```
<div x-data="{ open: $wire.entangle('showForm') }">
  <button @click="open = ! open">Toggle</button>

  <div x-show="open">
    <form wire:submit.prevent="save">
      <input wire:model="title">
    </form>
  </div>
</div>

// Livewire component
class SomeForm extends Component
{
    public $showForm;
    public $title;
```

```
public function save()
{

}
}
```

- Bila Alpine open berubah → Livewire showForm juga berubah.
- Dua hala (two-way binding).

Kesimpulan

- Validation = penting untuk jaga data & UX.
- Livewire 3 guna #[Validate] → mudah & clean.
- Untuk form besar → guna Form Object.
- updated() + validateOnly → real-time validation.
- Custom messages buat sistem lebih mesra pengguna.
- rules() method → lebih flexible & maintainable.

Latihan

1. Gunakan `\$wire.entangle` untuk sync state modal dengan Livewire.
2. Cuba refresh komponen selepas simpan data baru.

Projek Mini

- Projek Job Board -> tambah Job Application Form.
- Gunakan Alpine untuk:
 - Modal transition (open/close).
 - Dropdown pilih lokasi kerja.
 - Convert kan semua componen (JobList, JobEdit, JobCreate, JobApply) untuk menggunakan Alpine.
- Gunakan `\$wire` & `entangle` untuk sync dengan Livewire.

Kelas Seterusnya: Lifecycle Hooks