Tentu, mari kita pecah tugas ini dan kerjakan satu per satu.

**6.3 Latihan Pembelajaran**

1.

2.

| Course | Type | Rate |
| --- | --- | --- |
| Matematika | Diskrit | 4 |
| Matematika | Aljabar Linear | 3 |
| Basis Data | DDL | 2 |
| Bhs Inggris | Speaker | 1 |

Untuk mengubahnya menjadi bentuk multiple array (atau array multidimensi), kita bisa merepresentasikannya sebagai array of objects atau array of arrays.

Contoh dalam PHP (umum digunakan di backend):

Sebagai Array of Objects (paling umum dan mudah dibaca):

PHP

$academicSkills = [

[

'Course' => 'Matematika',

'Type' => 'Diskrit',

'Rate' => 4

],

[

'Course' => 'Matematika',

'Type' => 'Aljabar Linear',

'Rate' => 3

],

[

'Course' => 'Basis Data',

'Type' => 'DDL',

'Rate' => 2

],

[

'Course' => 'Bhs Inggris',

'Type' => 'Speaker',

'Rate' => 1

]

];

Sebagai Array of Arrays (jika urutan kolom selalu sama):

PHP

$academicSkills = [

['Matematika', 'Diskrit', 4],

['Matematika', 'Aljabar Linear', 3],

['Basis Data', 'DDL', 2],

['Bhs Inggris', 'Speaker', 1]

];

3.Untuk bagian ini, kita akan membuat contoh sederhana dengan Laravel (framework PHP yang populer) karena disebutkan "blade profile" dan "blade layout".

Asumsi Struktur Proyek Laravel:

* app/Http/Controllers/ProfileController.php
* resources/views/layouts/app.blade.php
* resources/views/profile.blade.php
* routes/web.php

Langkah-langkah:

a. Definisikan Data Array di Controller (app/Http/Controllers/ProfileController.php):

PHP

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProfileController extends Controller

{

public function showProfile()

{

$academicSkills = [

[

'Course' => 'Matematika',

'Type' => 'Diskrit',

'Rate' => 4

],

[

'Course' => 'Matematika',

'Type' => 'Aljabar Linear',

'Rate' => 3

],

[

'Course' => 'Basis Data',

'Type' => 'DDL',

'Rate' => 2

],

[

'Course' => 'Bhs Inggris',

'Type' => 'Speaker',

'Rate' => 1

]

];

return view('profile', compact('academicSkills'));

}

}

b. Buat Blade Layout (resources/views/layouts/app.blade.php):

Blade

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Profile Page</title>

<style>

body { font-family: sans-serif; margin: 20px; }

.container { max-width: 800px; margin: 0 auto; padding: 20px; border: 1px solid #ccc; border-radius: 8px; }

table { width: 100%; border-collapse: collapse; margin-top: 20px; }

th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }

th { background-color: #f2f2f2; }

</style>

</head>

<body>

<div class="container">

@yield('content')

</div>

</body>

</html>

c. Buat Blade Profile (resources/views/profile.blade.php):

Blade

@extends('layouts.app')

@section('content')

<h1>Academic Skills Profile</h1>

<table>

<thead>

<tr>

<th>Course</th>

<th>Type</th>

<th>Rate</th>

</tr>

</thead>

<tbody>

@foreach ($academicSkills as $skill)

<tr>

<td>{{ $skill['Course'] }}</td>

<td>{{ $skill['Type'] }}</td>

<td>{{ $skill['Rate'] }}</td>

</tr>

@endforeach

</tbody>

</table>

@endsection

d. Definisikan Route (routes/web.php):

PHP

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\ProfileController;

Route::get('/profile', [ProfileController::class, 'showProfile'])->name('profile');

Cara Menjalankan (setelah setup Laravel):

1. Pastikan Anda memiliki Laravel terinstal.
2. Buat file-file di atas di lokasi yang sesuai.
3. Jalankan php artisan serve di terminal.
4. Buka browser Anda dan navigasi ke http://127.0.0.1:8000/profile (atau port yang diberikan oleh artisan serve).

4. Dari hasil kode nomor 3, silahkan anda buat tampilan hasil rendernya menjadi seperti gambar dibawah ini:

Sekarang kita akan memodifikasi profile.blade.php untuk menampilkan rating dalam bentuk bintang (★) dan bintang kosong (☆).

Modifikasi Blade Profile (resources/views/profile.blade.php):

Blade

@extends('layouts.app')

@section('content')

<h1>Academic Skills Profile</h1>

<table>

<thead>

<tr>

<th>Course</th>

<th>Type</th>

<th>Rate</th>

</tr>

</thead>

<tbody>

@foreach ($academicSkills as $skill)

<tr>

<td>{{ $skill['Course'] }}</td>

<td>{{ $skill['Type'] }}</td>

<td>

@php

$fullStars = $skill['Rate'];

$emptyStars = 5 - $fullStars;

@endphp

@for ($i = 0; $i < $fullStars; $i++)

<span style="color: gold;">★</span>

@endfor

@for ($i = 0; $i < $emptyStars; $i++)

<span style="color: lightgray;">☆</span>

@endfor

</td>

</tr>

@endforeach

</tbody>

</table>

@endsection

Penjelasan Modifikasi:

* Di dalam loop @foreach, kita menambahkan blok @php untuk menghitung fullStars (bintang penuh) dan emptyStars (bintang kosong).
* $fullStars langsung diambil dari nilai Rate.
* $emptyStars dihitung sebagai 5 - $fullStars karena diasumsikan rating maksimal adalah 5.
* Kemudian, dua loop @for digunakan:
  + Loop pertama mencetak bintang penuh (★) sejumlah fullStars. Saya tambahkan style="color: gold;" untuk membuatnya terlihat seperti bintang.
  + Loop kedua mencetak bintang kosong (☆) sejumlah emptyStars. Saya tambahkan style="color: lightgray;" untuk membedakannya.

Dengan perubahan ini, ketika Anda mengakses /profile, Anda akan melihat tabel dengan kolom "Rate" yang menampilkan bintang sesuai dengan nilai rating yang diberikan di data.