

PENGANTAR CRUD

PENGANTAR CRUD

Topik: Mengimplementasikan CRUD Dasar dengan Framework

Isi Slide:

Pemrograman Web 2

Sub-CPMK: Mengimplementasikan CRUD

Materi: CRUD Dasar dengan Framework (Laravel / CodeIgniter / Express.js)

Penjelasan:

Perkenalkan konteks: CRUD merupakan kompetensi utama dalam pengembangan aplikasi web dinamis.

Slide 2 — Tujuan Pembelajaran

Isi Slide:

- Mahasiswa memahami konsep CRUD
- Mahasiswa mampu menerapkan operasi CRUD menggunakan framework
- Mahasiswa dapat menghubungkan CRUD dengan database & ORM

Penjelasan:

Target: mahasiswa dapat membuat fitur tambah, edit, tampil, hapus data.

Slide 3 — Konsep CRUD

Isi Slide:

CRUD = **Create, Read, Update, Delete**

Digunakan untuk mengelola data dalam sistem web.

Penjelasan:

CRUD adalah operasi dasar pada database yang digunakan hampir di semua aplikasi.

Slide 4 — Alur CRUD dalam Web

Isi Slide:

Client → Controller → Model → Database → View

Penjelasan:

Setiap request dari user diproses oleh controller, model mengakses database, view menampilkan data.

Slide 5 — Framework yang Digunakan

Isi Slide:

- Laravel (PHP, ORM Eloquent)
- CodeIgniter (PHP)
- Express.js (Node.js, ORM Sequelize)

Penjelasan:

Framework mempermudah routing, query, dan struktur kode.

Slide 6 — Struktur Folder Inti**Laravel:**

app/Models

app/Http/Controllers

resources/views

routes/web.php

Penjelasan:

Menunjukkan dimana model, controller, view, dan routing berada.

Slide 7 — Routing Create**Laravel contoh:**

```
Route::get('/product/create', [ProductController::class, 'create']);
```

```
Route::post('/product/store', [ProductController::class, 'store']);
```

Penjelasan:

Gunakan GET untuk form input, POST untuk simpan data.

Slide 8 — Form Tambah Data (View)**Blade contoh:**

```
<form action="/product/store" method="POST">
```

```
@csrf
```

```
<input type="text" name="name">
```

```
<button>Simpan</button>
```

```
</form>
```

Penjelasan:

@csrf diperlukan untuk keamanan form di Laravel.

Slide 9 — Controller Store

```
public function store(Request $request){  
    Product::create(['name' => $request->name]);  
    return redirect('/product');  
}
```

Penjelasan:

ORM Eloquent memudahkan perintah insert data.

Slide 10 — Controller Index

```
public function index(){  
    $data = Product::all();  
    return view('product.index', compact('data'));  
}
```

Penjelasan:

Mengambil semua data dari tabel *products*.

Slide 11 — Tampilan Data (View)

```
@foreach($data as $p)  
  
<tr>  
    <td>{{ $p->name }}</td>  
  
</tr>  
  
@endforeach
```

Penjelasan:

Loop data untuk ditampilkan ke tabel HTML.

Slide 12 — Routing Edit

```
Route::get('/product/{id}/edit', [ProductController::class, 'edit']);
```

```
Route::post('/product/{id}/update', [ProductController::class, 'update']);
```

Slide 13 — Form Edit

```
<input value="{{ $product->name }}" name="name">
```

Penjelasan:

Form diisi otomatis dengan data lama.

Slide 14 — Controller Update

```
$product->update(['name' => $request->name]);
```

Penjelasan:

ORM menangani perubahan data hanya dengan satu baris.

Slide 15 — Routing Delete

```
Route::get('/product/{id}/delete', [ProductController::class, 'delete']);
```

Slide 16 — Controller Delete

```
$product->delete();
```

```
return redirect('/product');
```

Slide 17 — Tombol Hapus

```
<a href="/product/{{ $p->id }}/delete">Hapus</a>
```

Penjelasan:

Pastikan lakukan konfirmasi hapus agar aman.

Slide 18 — Validasi Input

```
$request->validate([  
    'name' => 'required|min:3'  
]);
```

Penjelasan:

Validasi mencegah data kosong & garbage data.

Slide 19 — CSRF Protection

Isi Slide:

Laravel menggunakan @csrf

Express.js menggunakan middleware csrf

Penjelasan:

Keamanan wajib diterapkan dalam setiap form.

Slide 20 — Ringkasan & Tugas

Isi Slide:

- CRUD adalah inti aplikasi web dinamis
- Implementasi menggunakan MVC: Route → Controller → Model → View
- ORM mempermudah manipulasi database

Tugas Praktik:

- Buat CRUD sederhana untuk data *Mahasiswa* dengan framework yang kalian pilih.