

# Pertemuan 6

## Membuat fungsi Edit Kelas

buat untuk fungsi edit di route di web.php

```
Route::post('kelas/aksi_edit/{id}',[KelasController::class,'aksi_edit'])
->name('aksi_edit')
;
```

selanjutnya,buka file **edit.blade.php** di folder **kelas** lalu tambahkan action di form yang mengarah ke route **aksi\_edit**

```
@extends('layouts.index')
@section('content')
<div class="container">
  <h2>Edit kelas</h2>
  <form action="{{ route('aksi_edit',$dataKelas->id) }}" method="post">
    @csrf
    <div class="mb-3">
      <label class="form-label">Kelas</label>
      <input type="text" value="{{ $dataKelas->kelas }}" name="kelas" class="form-
control"
        placeholder="Masukan kelas">
    </div>
    <button type="submit" class="btn btn-primary">Edit</button>
  </form>
</div>
@endsection
```

Lanjut..buka file **KelasController.php** lalu buat fungsi **aksi\_edit** nya

```
public function aksi_edit(Request $request)
{
}
```

jika,sudah di buat coba kita debug data dari form nya dengan cara

```
public function aksi_edit(Request $request){
  dd($request->all());
}
```

Metode **all()** pada objek request (**\$request**) digunakan untuk mengambil semua input yang dikirimkan dengan request tersebut, baik itu data dari formulir HTML atau parameter URL.

jika kamu punya form seperti ini

```
@extends('layouts.index')
@section('content')
<div class="container">
  <h2>Edit kelas</h2>
  <form action="{{ route('aksi_edit',$dataKelas->id) }}" method="post">
    @csrf
    <div class="mb-3">
      <label class="form-label">Kelas</label>
      <input type="text" value="{{ $dataKelas->kelas }}" name="kelas" class="form-control"
        placeholder="Masukan kelas">
    </div>
    <button type="submit" class="btn btn-primary">Edit</button>
  </form>
</div>
@endsection
```

saat form tersebut dikirimkan, **\$request->all()** akan menghasilkan **array asosiatif** seperti:

```
array:2 [▼ // app\Http\Controllers\KelasController.php:63
  "_token" => "si28Xx5LY1L3LBQbR4uVbSaGebMaQoeScSrRFPbs"
  "kelas" => "10 pplg 2"
]
```

Fungsi **dd()** adalah singkatan dari "**dump and die.**"

setelah kita sudah memastikan bahwa data yang di input benar dan key nya sesuai dengan field di **table kelas**.

selanjutnya..kita ambil data dari form yang field nya **kelas** saja.karena key **\_token** adalah bawaan dari laravel dan kita tidak punya **field \_token di table kelas**.

untuk mengambil key kelas nya saja kita bisa gunakan ini

```
public function aksi_edit(Request $request){
  dd($request->only(['kelas']));
}
```

Ketika form ini dikirim, **\$request->only(['kelas'])** akan menghasilkan array asosiatif yang hanya berisi data dengan kunci 'kelas' seperti ini.

```
array:1 [▼ // app\Http\Controllers\KelasController.php:63
  "kelas" => "10 pplg 2"
]
```

selanjutnya buat fungsi update.

pertama,tambahkan parameter **\$id** setelah **\$request**

```
public function aksi_edit(Request $request,$id){  
    dd($request->only(['kelas']));  
}
```

lalu buat fungsi seperti ini

```
public function aksi_edit(Request $request,$id){  
    Kelas::where('id',$id)->update($request->only(['kelas']));  
}
```

- **Kelas::where('id', \$id):** Bagian ini membentuk query yang mencari ke dalam tabel Kelas di mana kolom **id** sama dengan nilai **\$id**.
- **->update(\$request->only(['kelas'])):** Bagian ini menjalankan operasi update yang ditemukan dengan data yang diambil dari permintaan. Data yang diambil hanya berisi kolom kelas.jadi data yang di update hanya field/kolum kelas saja.

jika di coba,maka data akan di terupdate,tapi tampilannya masih kosong.jadi sekarang redirect ke **route kelas** jika data sudah di update.

```
public function aksi_edit(Request $request,$id){  
    Kelas::where('id',$id)->update($request->only(['kelas']));  
    return redirect()->route('kelas');  
}
```

## Membuat validasi tambah kelas

**Validasi di Laravel** adalah proses memeriksa data yang masuk untuk memastikan bahwa data tersebut memenuhi aturan yang ditentukan sebelum digunakan dalam aplikasi. Laravel menyediakan fitur validasi yang kuat dan fleksibel yang memungkinkan developer untuk dengan mudah memvalidasi input dari user.

### Validasi Menggunakan validate Method

Laravel menyediakan metode validate pada objek Request untuk memvalidasi data yang masuk.

contoh buka file **KelasController.php** lalu tambahkan script ini

```

public function aksi_tambah(Request $request){
    $request->validate([
        'kelas' => 'required',
    ],
    );
    Kelas::insert([
        'kelas'=>$request->kelas
    ]);
    return redirect()->route('kelas');
}

```

Laravel menyediakan banyak aturan validasi yang bisa digunakan, seperti

- **required:** Field harus diisi.

## Menampilkan Pesan Kesalahan

Laravel otomatis menyediakan pesan kesalahan validasi yang sesuai, namun Anda dapat menyesuaikan pesan kesalahan jika diperlukan.

Menampilkan Pesan Kesalahan di Blade Template:

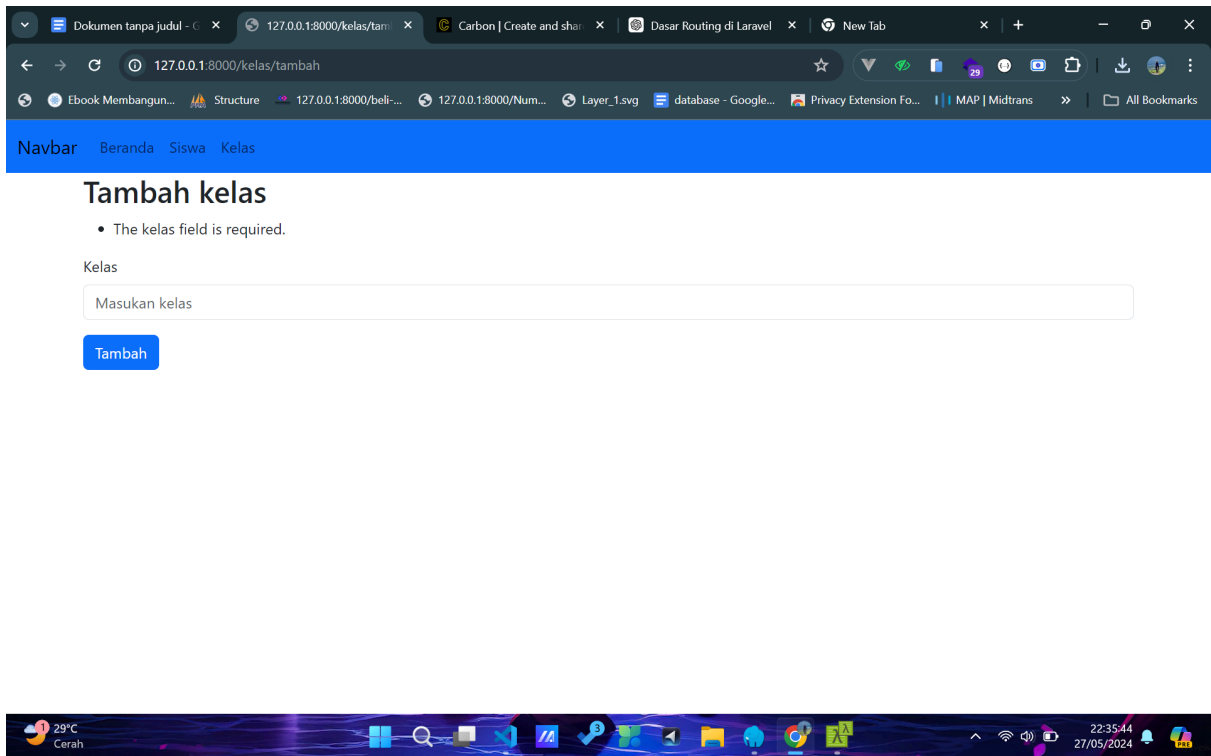
buka file **tambah.blade.php** di folder kelas

```

@extends('layouts.index')
@section('content')
<div class="container">
    <h2>Tambah kelas</h2>
    <!-- kode yang di tambahkan -->
    @if ($errors->any())
    <div>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
    @endif
    <!-- akhir -->
    <form action="{{ route('aksi_tambah') }}" method="post">
        @csrf
        <div class="mb-3">
            <label class="form-label">Kelas</label>
            <input type="text" name="kelas" class="form-control"
                placeholder="Masukan kelas">
        </div>
        <button type="submit" class="btn btn-primary">Tambah</button>
    </form>
    </div>
@endsection

```

jika dicoba akan muncul text seperti ini jika ketika tambah data kelas tetapi inputnya kosong.

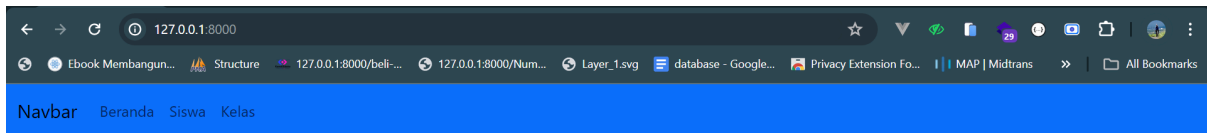


## Membuat dashboard

hapus codingan di dalam `@section('content')` , lalu buat desain card di file `index.blade.php` di folder `home` seperti ini

```
resources > views > home > index.blade.php > div.container.mt-3 > div.row > div.col-lg-3.mx-2 > div.card.bg-primary > div.card-body > h3.card-subt
2 @section( 'title', Home )
3 @section('content')
4 <div class="container mt-3">
5   <div class="row">
6     <div class="col-lg-3 mx-2">
7       <div class="card bg-primary " style="width: 18rem;">
8         <div class="card-body ">
9           <h2 class="card-title text-white">Total Kelas</h2>
10          <h3 class="card-subtitle mb-2 text-white">10</h3>
11        </div>
12      </div>
13    </div>
14    <div class="col-lg-3 mx-2">
15      <div class="card bg-primary " style="width: 18rem;">
16        <div class="card-body ">
17          <h2 class="card-title text-white">Total Siswa</h2>
18          <h3 class="card-subtitle mb-2 text-white">10</h3>
19        </div>
20      </div>
21    </div>
22  </div>
23 </div>
24 @endsection
```

hasilnya akan seperti ini



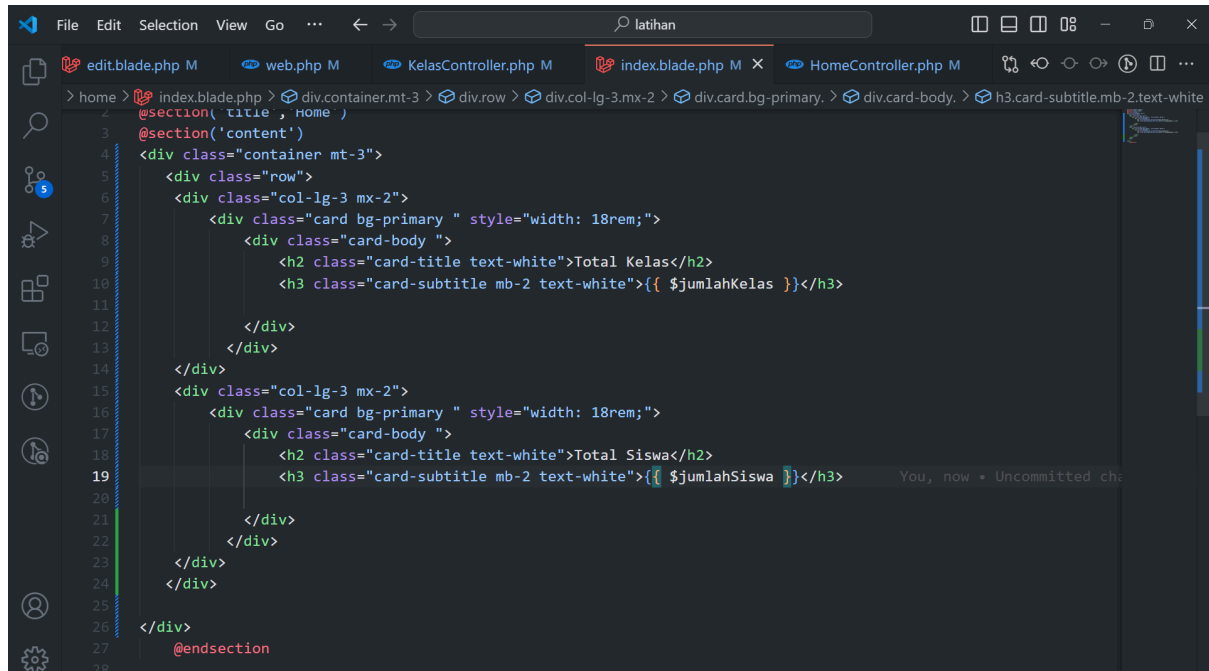
selanjutnya..buat fungsi untuk menghitung jumlah siswa dan kelas.

buka file **HomeController.php** di **method index** lalu ubah menjadi seperti ini

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Kelas; //tambahkan model Kelas nya juga
6 use App\Models\Siswa; //tambahkan model Siswa nya juga
7 use Illuminate\Http\Request;
8
9 You, 27 seconds ago | 1 author (You)
10 class HomeController extends Controller
11 {
12     public function index(){
13         $jumlahKelas=Kelas::count();// Menghitung jumlah kelas
14         $jumlahSiswa=Siswa::count(); // Menghitung jumlah siswa
15         return view('home.index',
16             ['jumlahKelas'=>$jumlahKelas,'jumlahSiswa'=>$jumlahSiswa]);
17     }
18 }
```

- **Kelas::count()** : Metode count digunakan untuk menghitung jumlah total baris dalam tabel kelas.
- Variabel **\$jumlahKelas** akan menyimpan hasil dari perhitungan jumlah kelas.
- **Siswa::count()** : Method count digunakan untuk menghitung jumlah total baris dalam tabel siswa.
- Variabel **\$jumlahSiswa** akan menyimpan hasil dari perhitungan jumlah siswa

kemudian buka file **index.blade.php** di folder home dan tampilkan variabel **\$jumlahKelas** dan **\$jumlahSiswa** seperti ini



```
1 <@section('title', Home)>
2 </@section>
3 <@section('content')>
4 <div class="container mt-3">
5 <div class="row">
6 <div class="col-lg-3 mx-2">
7 <div class="card bg-primary " style="width: 18rem;">
8 <div class="card-body ">
9 <h2 class="card-title text-white">Total Kelas</h2>
10 <h3 class="card-subtitle mb-2 text-white">{{ $jumlahKelas }}</h3>
11 </div>
12 </div>
13 </div>
14 <div class="col-lg-3 mx-2">
15 <div class="card bg-primary " style="width: 18rem;">
16 <div class="card-body ">
17 <h2 class="card-title text-white">Total Siswa</h2>
18 <h3 class="card-subtitle mb-2 text-white">{{ $jumlahSiswa }}</h3>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 </div>
26 </@section>
27 </div>
```

hasilnya akan seperti ini, Total Kelas dan Total Siswa akan sama dengan total data di table kelas dan siswa

