

MODUL 4

OPERASI *CREATE READ UPDATE DELETE (CRUD)*



CAPAIAN PEMBELAJARAN

1. Mampu membuat aplikasi Create, Read, Update, Delete (CRUD)



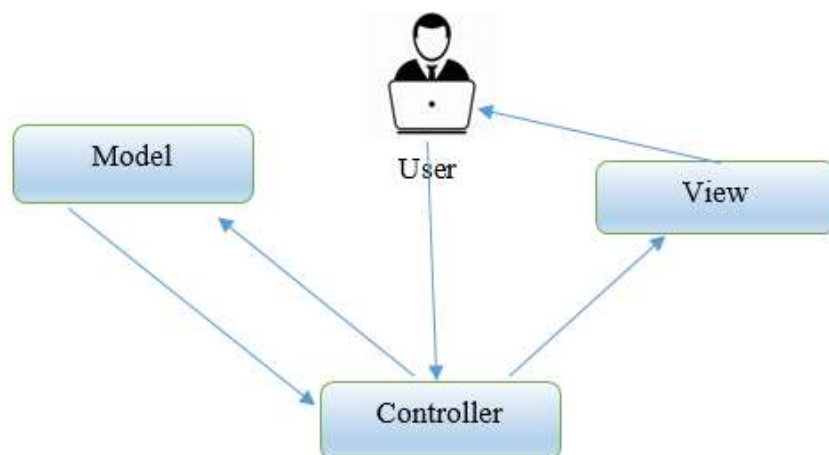
KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Sistem Operasi Linux/Window 10
2. PHP 5.x, diutamakan versi 7.x keatas
3. Composer
4. XAMPP (PHP, Apache, MySQL)
5. Text Editor Notepad++ atau Visual Studio Code



DASAR TEORI

1. *Create, Read, Update, Delete (CRUD)*, biasanya untuk manajemen rekaman tabel mater. Untuk membuat CRUD, diperlukan Model, Controller, dan views, seperti pada gambar berikut:



2. Untuk membuat membuat *Model-Views-Controllers* menggunakan command line artisan:

```
Php artisan make:model Kota
Php artisan make:model models\Kota (menyimpan ke subfolder models)
Php artisan make:controller KotaController --resource
```

3. Untuk Model Kota seperti pada skrip berikut:

```
<?php

namespace App\models;

use Illuminate\Database\Eloquent\Model;

class Kota extends Model
{
    // .....
}
```

4. Untuk KotaController seperti pada skrip berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class KotaController extends Controller
{

    public function index()
    {
        //-----
    }

    public function create()
    {
        //-----
    }

    public function store(Request $request)
    {
        //-----
    }
}
```

```

public function show($id)
{
    //-----
}

public function edit($id)
{
    //-----
}

public function update(Request $request, $id)
{
    //
}

public function destroy($id)
{
    //-----
}
}

```

5. View index.blade.php

```

.....
<table class="table table-striped table-bordered">
    <thead>
        <tr>
            <th>Id</th>
            <th>ID Propinsi </th>
            <th>Nama Kota</th>
            <th>Aksi</th>

        </tr>
    </thead>
    <tbody>
        @foreach($kota as $key => $value)
            <tr>
                <td>{{ $value->id }}</td>
                <td>{{ $value->id_propinsi }}</td>
                <td>{{ $value->nama_kota }}</td>
                <td>
                    <form action="{{ route('kota.destroy', $value->id) }}"
                        method="post">

```

```

        {{ csrf_field() }}
        {{ method_field('DELETE') }}
        <a href="{{ route('kota.edit',$value->id) }}"
        class=" btn btn-sm btn-primary">Edit</a>
        onclick="return confirm(
            'Yakin ingin menghapus kota :
            {{$value->nama_kota}}.....?')">Delete</button>
    </form>
</td>
</tr>
@endforeach
</tbody>
</table>
<?php echo str_replace('/?', '?', $kota->render()); ?>
.....

```



PRAKTIK

1. Bualah model

```
php artisan make:model models/Propinsi
```

2. Buka tambahkan file models/Propinsi.php seperti pada kode program berikut:

File Propinsi.php

```

<?php

namespace App\models;

use Illuminate\Database\Eloquent\Model;

class Propinsi extends Model
{
    protected $table = 'propinsi';
    protected $primaryKey = 'id';
    protected $fillable = ['id','propinsi'];

}

```

Akhir File Propinsi.php

3. Buatlah PropinsiController, dengan perintah berikut:

```
php artisan make:controller PropinsiController --resource
```

4. Buka tambahkan file `app/http/Controllers/PropinsiController.php` seperti pada kode program berikut:

```
File PropinsiController.php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\models\Propinsi;

class PropinsiController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $propinsi = Propinsi::orderBy('id', 'DESC')->paginate(5);
        return view('propinsi.index', compact('propinsi'));
    }
    .....
    .....
}
File PropinsiController.php
```

5. Menambahkan gaya tampilan untuk semua view tambahkan file `views/layouts/app.blade.php`, seperti pada skrip berikut:

```
File app.blade.php

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
```

```

<link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.
min.css" rel="stylesheet">
<title>Propinsi - CRUD</title>
<style>
    .martop-sm {margin-top: 15px;}
    .martop-lg {margin-top: 70px;}
</style>
</head>
<body>
    <div class="container martop-lg">
        <div class="panel panel-default">
            <div class="panel-body">@yield('content')</div>
        </div>
    </div>

    <script src=
"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js
">
</script>
</body>
</html>

```

Akhir file app.blade.php

6. Buatlah view index.blade.php simpan ke /resources/views/propinsi/.. seperti pada kode program berikut:

File index.blade.php

```

@extends('layouts.app')

@section('content')
    <h4>Menajemen Tabel Propinsi</h4>

    <a href="{{ route('propinsi.create') }}"
        class="btn btn-info btn-sm">Propinsi Baru</a>

    @if ($message = Session::get('message'))
        <div class="alert alert-success martop-sm">
            <p>{{ $message }}</p>
        </div>
    @endif

    <table class="table table-responsive martop-sm">
        <thead>
            <th>ID</th>
            <th>Propinsi</th>

```

```

        <th>Action</th>
    </thead>
    <tbody>
        @foreach ($propinsi as $p)
            <tr>
                <td>{{ $p->id }}</td>
                <td><a href="{{ route('propinsi.show', $p->id) }}">
                    {{ $p->propinsi }}</a></td>
                <td>
                    <form action="{{ route('propinsi.destroy', $p->id) }}"
                        method="post">
                        {{ csrf_field() }}
                        {{ method_field('DELETE') }}
                        <a href="{{ route('propinsi.edit', $p->id) }}"
                            class="btn btn-warning btn-sm">Ubah</a>
                        <button type="submit"
                            class="btn btn-danger btn-sm">Hapus</button>
                    </form>
                </td>
            </tr>
        @endforeach
    </tbody>
</table>
@endsection

```

Akhir file index.blade.php

Pengujian jalankan url <http://localhost:8000/propinsi>

7. Menambah rekaman baru tambahkan PropinsiController seperti pada skrip berikut:

```

Method function create() dan store()

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('propinsi.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)

```

```

    {
        $this->validate($request, [
            'propinsi' => 'required',

        ]);

        $propinsi = Propinsi::create($request->all());

        return redirect()->route('propinsi.index')
            ->with('message',
                'Propinsi baru berhasil ditambahkan!');
    }
}
.....
Method function create() dan store()

```

8. Tambahkan views create.blade.php seperti pada skrip program berikut:

```

File create.blade.php

@extends('layouts.app')

@section('content')
<h4>Propinsi Baru</h4>
<form action="{{ route('propinsi.store') }}" method="post">
    {{csrf_field()}}
    <div class="form-group {{ $errors->has('propinsi') ?
        'has-error' : '' }}">
        <label for="propinsi" class="control-label">Propinsi</label>
        <input type="text" class="form-control" name="propinsi"
            placeholder="Propinsi">
        @if ($errors->has('propinsi'))
            <span class="help-block">{{ $errors->first('propinsi') }}>
        @endif
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-info">Simpan</button>
        <a href="{{ route('propinsi.index') }}"
            class="btn btn-default">Kembali</a>
    </div>
</form>
@endsection

Akhir file create.blade.php

```

9. Mengubah rekaman tambah PropinsiController seperti pada kode berikut:

Method function edit(), dan update()

```
/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $propinsi = Propinsi::findOrFail($id);
    return view('propinsi.edit', compact('propinsi'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $this->validate($request, [
        'propinsi' => 'required',

    ]);

    $propinsi = Propinsi::findOrFail($id)
        ->update($request->all());

    return redirect()->route('propinsi.index')
        ->with('message', 'Propinsi baru berhasil diubah!');
}
```

Akhir method function edit(), dan update()

10. Tambahkan file edit.blade.php seperti pada kode program berikut:

File edit.blade.php

```
@extends('layouts.app')

@section('content')
<h4>Ubah Propinsi</h4>
<form action="{{ route('propinsi.update', $propinsi->id) }}"
method="post">
    {{csrf_field()}}
    {{ method_field('PUT') }}
    <div class="form-group {{ $errors->has('propinsi') ?
        'has-error' : '' }}">
        <label for="propinsi" class="control-label">Propinsi</label>
```

```

        <input type="text" class="form-control" name="propinsi"
            placeholder="Propinsi" value="{{ $propinsi->propinsi }}">
        @if ($errors->has('propinsi'))
            <span class="help-block">{{ $errors->first('propinsi') }}</span>
        @endif
    </div>

    <div class="form-group">
        <button type="submit" class="btn btn-info">Simpan</button>
        <a href="{{ route('propinsi.index') }}"
            class="btn btn-default">Kembali</a>
    </div>
</form>
@endsection
Akhir File edit.blade.php

```

11. Menampilkan satu rekaman tambahan PropinsiController seperti pada kode program berikut:

```

Method function show($id)

public function show($id)
{
    $propinsi = Propinsi::findOrFail($id);
    return view('propinsi.show', compact('propinsi'));
}

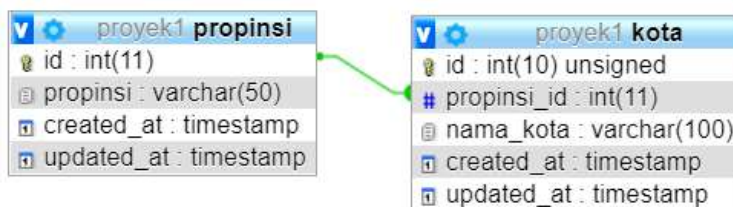
Akhir method function show($id)

```

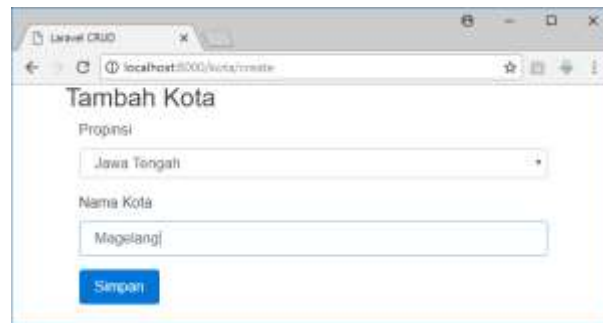


LATIHAN

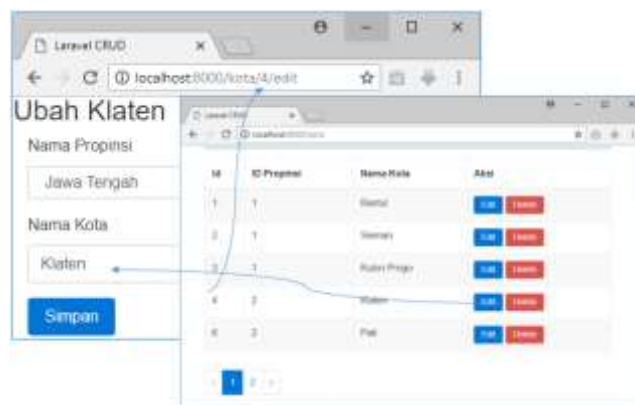
1. Buatlah CRUD untuk mengelola tabel kota



2. Buatlah model Kota relasi dengan Model Propinsi
3. Buatlah controller KotaController,
4. View create seperti pada gambar berikut:



5. View index dan update seperti pada gambar berikut:



5. Tambahkan 5 rekaman baru



TUGAS

1. Jelaskan method `create()`, `save()`, dalam kelas Model
2. Apa yang dimaksud use `app/model/Propinsi`?
3. Apa maksud perintah `compact('kota')`?
4. Apa yang dimaksud perintah `store(Request $request)` dan `$request->All()`?



REFERENSI

1. Laravel 5.7 CRUD (Create Read Update Delete) Tutorial Example, <https://itsolutionstuff.com/post/laravel-57-crud-create-read-update-delete-tutorial-example-example.html>
2. Laravel 5.8 Tutorial: Build your First CRUD App with Laravel and MySQL (PHP 7.1+), <https://www.techiediaries.com/php-laravel-crud-mysql-tutorial/>

