

TUGAS LAPORAN
MEMBUAT CRUD (Query Builder)
(Workshop Web Framework)



Dosen Pengampu :

Mukhamad Angga Gumilang, S. Pd., M. Eng.

Disusun Oleh :

Nama : Sang Bintang P. A

NIM : E41191570

Golongan : G/B Bondowoso

PRODI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFROMASI
POLITEKNIK NEGERI JEMBER

2021

PENDAHULUAN

1. Pengertian Laravel

PHP menjadi bahasa pemrograman yang sangat dinamis, tapi semenjak adanya Laravel, dia menjadi lebih powerful, cepat, aman, dan simpel. Setiap rilis versi terbaru, Laravel selalu memunculkan teknologi baru di antara framework PHP lainnya.

Laravel diluncurkan sejak tahun 2011 dan mengalami pertumbuhan yang cukup eksponensial. Di tahun 2015, Laravel adalah framework yang paling banyak mendapatkan bintang di Github. Sekarang framework ini menjadi salah satu yang populer di dunia, tidak terkecuali di Indonesia.

Laravel fokus di bagian end-user, yang berarti fokus pada kejelasan dan kesederhanaan, baik penulisan maupun tampilan, serta menghasilkan fungsionalitas aplikasi web yang bekerja sebagaimana mestinya. Hal ini membuat [developer](#) maupun perusahaan menggunakan framework ini untuk membangun apa pun, mulai dari proyek kecil hingga skala perusahaan kelas atas.

Laravel mengubah pengembangan website menjadi lebih elegan, ekspresif, dan menyenangkan, sesuai dengan jargonnya “*The PHP Framework For Web Artisans*”. Selain itu, Laravel juga mempermudah proses pengembangan website dengan bantuan beberapa fitur unggulan, seperti *Template Engine*, *Routing*, dan *Modularity*.

2. Pengertian Auth Laravel

Laravel auth adalah fitur esensial yang pasti ada di setiap aplikasi dan website yang dibuat memakai framework Laravel. Auth amat penting sebab memungkinkan user untuk melakukan verifikasi email, login akun, reset password dan lain sebagainya pada website maupun aplikasi.

Ada pula yang mendefinisikan Laravel auth (authentication) sebagai elemen security/perlindungan untuk melindungi halaman-halaman dan bagian penting dari sebuah web atau aplikasi.

Dengan adanya Laravel auth, maka hanya orang-orang tertentu saja yang bisa mengakses halaman penting tersebut. Orang-orang tersebut biasanya dikenal sebagai admin dan sudah mempunyai hak authentication khusus.

Berkat fungsi utamanya tersebut, Laravel auth sering kali digunakan pada fitur-fitur administrator website atau aplikasi. Sistem authentication juga bisa diaplikasikan pada software yang mempunyai menu user (pengguna) yang bisa registrasi dan login ke akun masing-masing, semisal pada aplikasi dan website e-commerce.

Membuat fungsi auth pada Laravel ini sangat mudah, sebab framework Laravel menyediakan fitur-fitur pembuatan auth siap pakai. Anda tidak perlu lagi repot melakukan instalasi modul khusus untuk membuat auth.

Cukup akses file konfigurasi otentikasi auth.php di bagian config pada Laravel, kemudian Anda bisa melakukan coding konfigurasi untuk membuat menu auth. Untuk membuat auth di Laravel, Anda bisa memakai php artisan auth dengan kode `$ php artisan make:auth`.

3. Pengertian Middleware Laravel

Middleware Laravel adalah penengah antara request yang masuk dengan controller yang dituju yang berpengaruh pada request dan respons. Misalnya kita ingin memverifikasi setiap request yang masuk seperti melakukan pengecekan status login, privilege, CSRF (Cross-Site Request Forgery) atau yang lainnya.

Dalam melakukan pengembangan aplikasi berbasis web, seringkali kita berkeinginan untuk memfilter setiap request yang masuk kedalam aplikasi kita. Hal ini dilakukan dengan tujuan untuk melakukan proses verifikasi terhadap setiap request yang masuk seperti misalnya melakukan pengecekan status login, privilege, atau bahkan melakukan pengecekan token CSRF (Cross-Site Request Forgery) untuk memastikan bahwa request yang masuk berasal dari komputer yang valid. Dengan menggunakan framework Laravel, kita dapat mengimplementasikan mekanisme tersebut dengan bantuan fitur Middleware yang telah disediakan oleh laravel-5 (pada laravel-4 fitur ini disebut sebagai filter dengan konsep yang sedikit berbeda).

Dalam konteks Laravel, Middleware merupakan sebuah Class khusus yang berperan sebagai "penengah" antara request yang masuk dengan Controller yang dituju. Secara umum, prinsip kerja Middleware adalah mencegat request yang masuk untuk kemudian diproses terlebih dahulu sebelum diberikan kepada Controller yang dituju atau diarahkan ke Controller yang lain. Dengan menggunakan fitur ini, kita dapat membuat komponen yang reusable untuk melakukan pekerjaan-pekerjaan tersebut.

4. Pengertian CRUD Laravel

CRUD Laravel adalah aneka operasi manipulasi dan pengolahan data di database yang bisa dilangsungkan dengan Laravel. CRUD Laravel juga merupakan singkatan (akronim) dari Create (buat), Read (baca / menampilkan), Update (memperbaharui), dan Delete (menghapus) data dari database.

Seluruh aktivitas ini merupakan kegiatan-kegiatan dasar dari pengolahan data di database, jadi, Anda pun wajib untuk bisa menguasainya. Dengan mempelajari tutorial CRUD Laravel, Anda bisa mengolah aneka data di database dan menampilkannya dengan lebih rapi. Data yang rapi pun akan lebih mudah untuk dicerna dan dipahami oleh user.

Fungsi Update di CRUD berguna untuk mengubah data yang sudah terdapat di server, misalnya pada untuk pengguna web ingin mengubah data alamat yang sudah tersimpan pada proses registrasi pada sebuah web, mengganti profile picture, mengedit blog post, dan berbagai aktivitas lainnya.

Seperti halnya pada fungsi Insert, pada fungsi Update juga dibutuhkan ketersediaan Form. Melalui form ini akan ditampilkan data yang sudah terdapat di server untuk kemudian pengguna dapat mengetikkan kalimat, memilih dari pilihan yang disediakan, atau misalnya dengan menekan tombol tertentu, dan apabila disimpan akan menggantikan data yang sebelumnya ada.

Delete merupakan salah satu fungsi yang tercakup dalam CRUD. Melalui fungsi ini data yang sudah tidak dibutuhkan dapat dihapus dari database. Contoh pemanfaatan

fitur ini misalnya adalah untuk menghapus blog post yang tidak perlu, menghapus transaksi yang salah, dan berbagai fungsi lainnya.

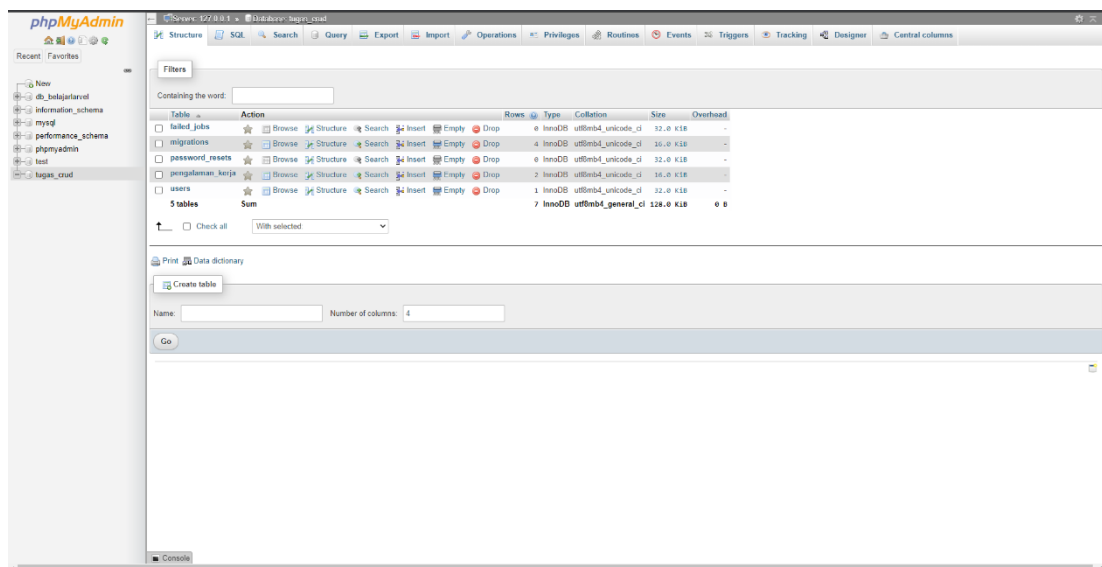
Namun demikian, untuk sebuah sistem yang baik perlu dipertimbangkan untuk pemanfaatan fitur delete. Pada transaksi misalnya, umumnya data tidak didelete secara fisik, namun diberikan sebuah field/kolom tertentu untuk menandai bahwa data dihapus serta dicatat tanggal penghapusannya, bahkan siapa pelakunya yang menghapus. Hal ini bertujuan untuk masalah kebutuhan ke depan yang serta untuk akuntabilitas data, dimana apabila dibutuhkan data masih bisa diakses.

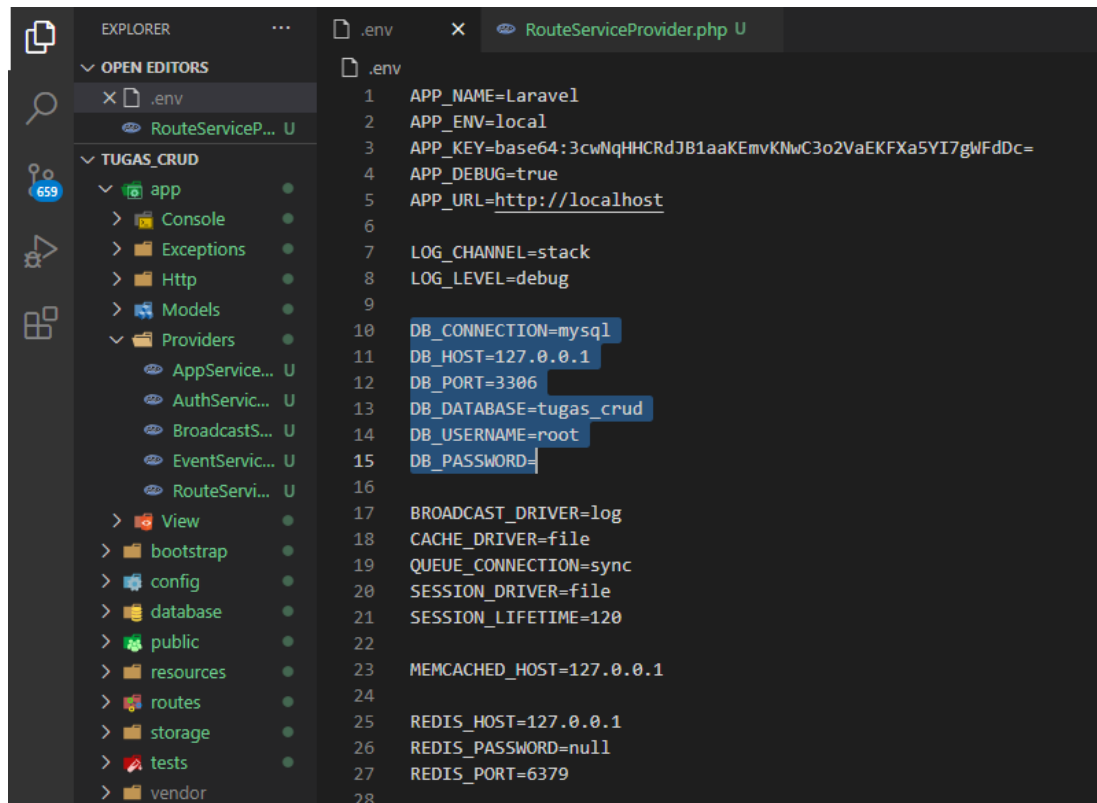
HASIL DAN PEMBAHASAN

Sebelumnya kita membuat AUTH Pada Laravel, setelah itu kita akan membuat CRUD pada Project Laravel kita. Tahap-tahapan seperti berikut :

A. Konfigurasi Database

Sebelumnya buat dulu database yang akan kalian pakai sesuai dengan database kalian (Saya asumsikan kalian sudah bisa membuat database-nya). Disini saya menggunakan MySQL pada phpMyAdmin. Setelah selesai membuat database, buka dile .env kalian, lalu setting sesuai nama, username dan password database-yang kalian buat.

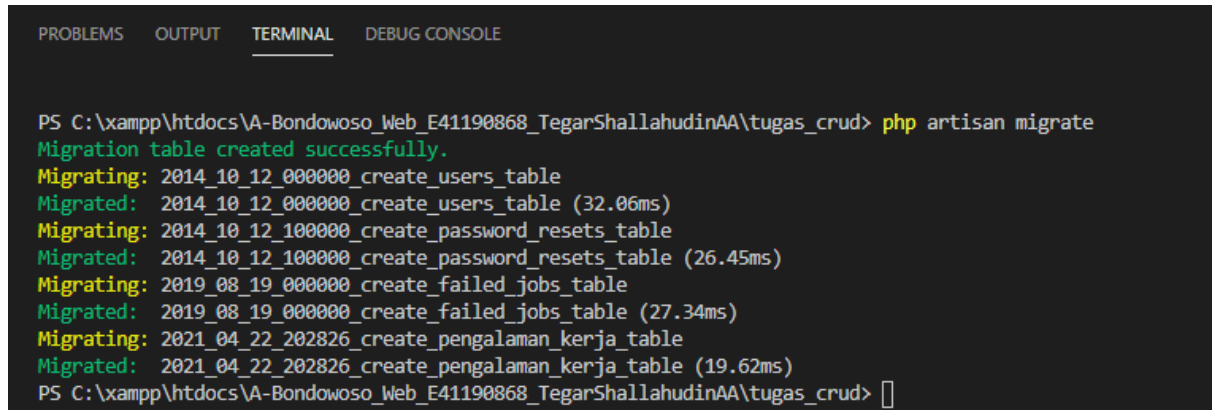




B. Membuat Migrasi pada database

Karena kita akan membuat fitur manajemen pengguna, maka kita langsung jalankan perintah migrasi saja, karena laravel sudah menyediakan file migrasi untuk tabel user/pengguna.

Sintax di Terminal : `php artisan migrate` atau `php artisan migrate:fresh`



```
PS C:\xampp\htdocs\A-Bondowoso_Web_E41190868_TegarShallahudinAA\tugas_crud> php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (32.06ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (26.45ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (27.34ms)
Migrating: 2021_04_22_202826_create_pengalaman_kerja_table
Migrated: 2021_04_22_202826_create_pengalaman_kerja_table (19.62ms)
PS C:\xampp\htdocs\A-Bondowoso_Web_E41190868_TegarShallahudinAA\tugas_crud> 
```

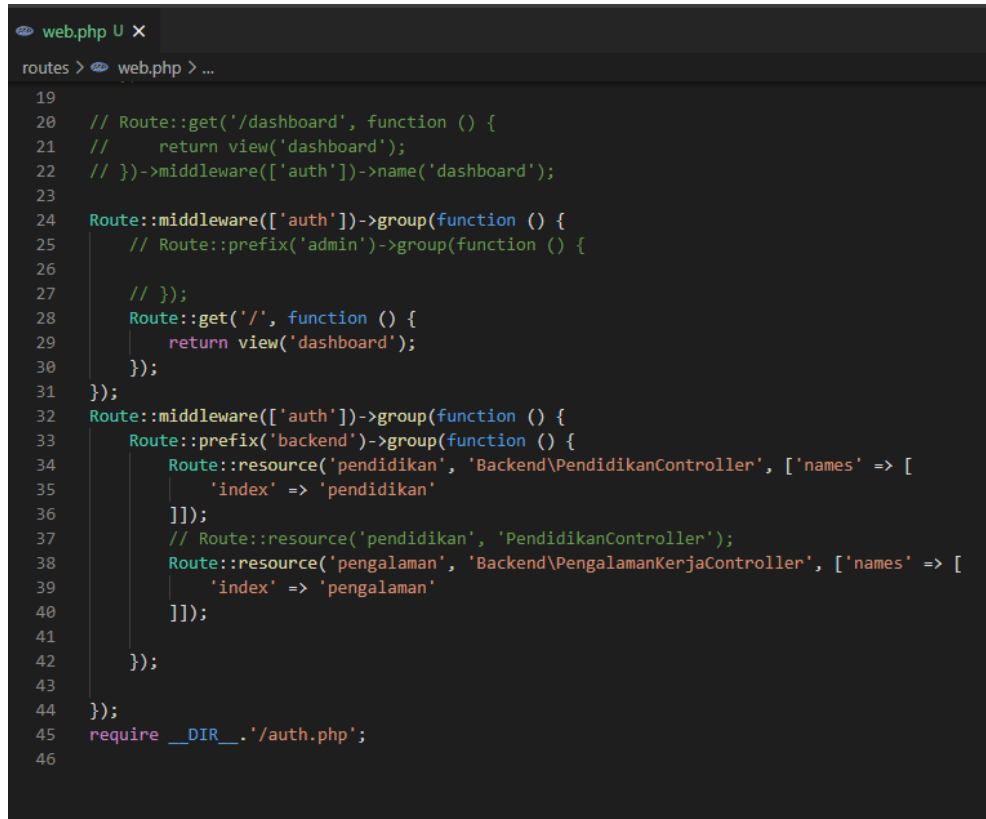

Membuat table pengalaman.

```
2021_04_22_202826_create_pengalaman_kerja_table.php U X
database > migrations > 2021_04_22_202826_create_pengalaman_kerja_table.php > ...
1 k?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreatePengalamanKerjaTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('pengalaman_kerja', function (Blueprint $table) {
17             $table->tinyIncrements('id');
18             $table->string('nama');
19             $table->string('jabatan');
20             $table->year('tahun_masuk');
21             $table->year('tahun_keluar');
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31     public function down()
32     {
33         Schema::dropIfExists('pengalaman_kerja');
34     }
35 }
36
```

Setelah itu anda tinggal menjalankan syntax *php artisan migrate* atau untuk memulai dari awal kalian bisa menggunakan syntax *php artisan migrate:fresh*.

C. Membuat Route

Disini lalu membuka folder **routes/web.php** dan membuat konfigurasi routes kalian sebagai berikut :



```
19
20 // Route::get('/dashboard', function () {
21 //     return view('dashboard');
22 // })->middleware(['auth'])->name('dashboard');
23
24 Route::middleware(['auth'])->group(function () {
25     // Route::prefix('admin')->group(function () {
26
27     // });
28     Route::get('/', function () {
29         return view('dashboard');
30     });
31 });
32 Route::middleware(['auth'])->group(function () {
33     Route::prefix('backend')->group(function () {
34         Route::resource('pendidikan', 'Backend\PendidikanController', ['names' => [
35             'index' => 'pendidikan'
36         ]]);
37         // Route::resource('pendidikan', 'PendidikanController');
38         Route::resource('pengalaman', 'Backend\PengalamanKerjaController', ['names' => [
39             'index' => 'pengalaman'
40         ]]);
41     });
42 });
43
44 });
45 require __DIR__.'/auth.php';
46
```

Pada file Controller yang kita dapat secara otomatis akan terdapat 7 fungsi berikut yang dapat kita gunakan untuk membuat operasi CRUD.

- index()
- create()
- store()
- show()
- edit()
- update()
- destroy()

Untuk melihat fungsi tersebut kalian dapat mengetikkan php artisan route:list

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure	auth api
	GET HEAD	backend/pendidikan	pendidikan	App\Http\Controllers\Backend\PendidikanController@index	auth:api web
	POST	backend/pendidikan	pendidikan.store	App\Http\Controllers\Backend\PendidikanController@store	auth web
	GET HEAD	backend/pendidikan/create	pendidikan.create	App\Http\Controllers\Backend\PendidikanController@create	auth web
	GET HEAD	backend/pendidikan/{pendidikan}	pendidikan.show	App\Http\Controllers\Backend\PendidikanController@show	auth web
	PUT PATCH	backend/pendidikan/{pendidikan}	pendidikan.update	App\Http\Controllers\Backend\PendidikanController@update	auth web
	DELETE	backend/pendidikan/{pendidikan}	pendidikan.destroy	App\Http\Controllers\Backend\PendidikanController@destroy	auth web
	GET HEAD	backend/pendidikan/{pendidikan}/edit	pendidikan.edit	App\Http\Controllers\Backend\PendidikanController@edit	auth web
	GET HEAD	backend/pengalaman	pengalaman	App\Http\Controllers\Backend\PengalamanKerjaController@index	auth web
	POST	backend/pengalaman	pengalaman.store	App\Http\Controllers\Backend\PengalamanKerjaController@store	auth web
	GET HEAD	backend/pengalaman/create	pengalaman.create	App\Http\Controllers\Backend\PengalamanKerjaController@create	auth web
	PUT PATCH	backend/pengalaman/{pengalaman}	pengalaman.update	App\Http\Controllers\Backend\PengalamanKerjaController@update	auth web
	DELETE	backend/pengalaman/{pengalaman}	pengalaman.destroy	App\Http\Controllers\Backend\PengalamanKerjaController@destroy	auth web
	GET HEAD	backend/pengalaman/{pengalaman}	pengalaman.show	App\Http\Controllers\Backend\PengalamanKerjaController@show	auth web

D. Membuat Models

Setelah itu kita akan menghubungkan table yang ada pada database dengan membuka folder pada **app/models**. Untuk membuat models baru kita dapat mengetikkan pada terminal yaitu : `php artisan make:model PengalamanKerja`. Dan menuliskan syntax sebagai berikut :

```

PengalamanKerja.php U X
app > Models > PengalamanKerja.php > ...
1  k?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class PengalamanKerja extends Model
9  {
10     use HasFactory;
11     protected $table = 'pengalaman_kerja';
12 }
13

```

E. Membuat syntax menambahkan data dan merubah data

Pertama, kita membuat tampilan tambah data dan mengubah data yang akan menjadi satu file , membuat tampilan menjadi lebih mudah pada folder **resources/views/backend/pengalaman/create.blade.php**, dan membuat **index.blade.php** untuk syntax dan tampilan sebagai berikut :

a. Syntax pada view

Seperti dibawah :

[illegible]

```

                <input type="text" class="form-
control" value="{{ isset($data) ? $data-
>jabatan : '' }}" id="jabatan" name="jabatan" minlength="2" required>
            </div>
        </div>
        <div class="form-group">
            <label for="curl" class="control-label col-lg-
2">Tahun Masuk<span class="required">*</span></label>
            <div class="col-lg-10">
                <input type="text" class="form-
control" value="{{ isset($data) ? $data-
>tahun_masuk : '' }}" id="tahun_masuk" name="tahun_masuk" minlength="4" requi
red>
            </div>
        </div>
        <div class="form-group">
            <label for="curl" class="control-label col-lg-
2">Tahun Selesai<span class="required">*</span></label>
            <div class="col-lg-10">
                <input type="text" class="form-
control" value="{{ isset($data) ? $data-
>tahun_keluar : '' }}" id="tahun_keluar" name="tahun_keluar" minlength="4" re
quired>
            </div>
        </div>
        <div class="form-group">
            <div class="col-lg-offset-2 col-lg-10">
                <button class="btn btn-
primary" type="submit">Simpan</button>
                <a href="{{ route('pengalaman') }}">
                    <button class="btn btn-
default" type="button">Kembali</button>
                </a>
            </div>
        </div>
    </form>
</div>
</div>
</section>
</div>
</div>
<!-- page end-->
</section>
</section>
<!--main content end-->

```

b. Sintax pada controller

- Menambah Data

```
public function store(Request $request)
{
    $validate = $request->validate([
        'nama' => 'required|max:100',
        'jabatan' => 'required',
        'tahun_masuk' => 'required',
        'tahun_keluar' => 'required'
    ],
    [
        'required' => ':Data Harus Terisi',
    ]
    );
    $pengalaman = new Pengalamankerja;
    $pengalaman->nama = $request->nama;
    $pengalaman->jabatan = $request->jabatan;
    $pengalaman->tahun_masuk = $request->tahun_masuk;
    $pengalaman->tahun_keluar = $request->tahun_keluar;
    $pengalaman->save();
    return redirect()->route('pengalaman')->with('success','Data Pengalaman kerja baru telah berhasil disimpan');
}
```

- Merubah Data

```
<a class="btn btn-warning" href="{{ route('pengalaman.edit',$item->id) }}"><i class="icon_check_alt2"></i></a>
```

PengalamanKerjaController.php U X

app > Http > Controllers > Backend > PengalamanKerjaController.php > ...



```
32         'jabatan' => 'required',
33         'tahun_masuk' => 'required',
34         'tahun_keluar' => 'required'
35     ],
36     [
37         'required' => ':Data Harus Terisi',
38     ]
39 );
40 $pengalaman = new PengalamanKerja;
41 $pengalaman->nama = $request->nama;
42 $pengalaman->jabatan = $request->jabatan;
43 $pengalaman->tahun_masuk = $request->tahun_masuk;
44 $pengalaman->tahun_keluar = $request->tahun_keluar;
45 $pengalaman->save();
46 return redirect()->route('pengalaman')->with('success','Data Pengalaman kerja baru telah berhasil disimpan');
47 }
48
49 public function show($id)
50 {
51     //
52 }
53
54
55 public function edit($id)
56 {
57     $this->param['title'] = 'Mengubah';
58     $this->param['data'] = PengalamanKerja::findOrFail($id);
59     return view('backend.pengalaman.create', $this->param);
60 }
61
62 public function update(Request $request, $id)
63 {
64     $updatePengalaman = PengalamanKerja::findOrFail($id);
65     $updatePengalaman->nama = $request->nama;
66     $updatePengalaman->jabatan = $request->jabatan;
67     $updatePengalaman->tahun_masuk = $request->tahun_masuk;
68     $updatePengalaman->tahun_keluar = $request->tahun_keluar;
69     $updatePengalaman->save();
70     return redirect()->route('pengalaman')->with('success', 'Pengalaman Kerja Berhasil Diperbaharui');
71 }
72
73 public function destroy($id)
74 {
75     $deletePengalaman = PengalamanKerja::findOrFail($id);
76     $deletePengalaman->delete();
77     return redirect()->route('pengalaman')->with('success','Data Pengalaman Kerja berhasil dihapus');
78 }
79 }
80
```

c. Tampilan Tambah dan Merubah data

The screenshot shows the 'Menambah Pengalaman Kerja' (Add Work Experience) form in the NICE ADMIN system. The form is located under the 'RIWAYAT HIDUP' (Resume) section. It contains four input fields: 'Nama Perusahaan*' (Company Name), 'Jabatan*' (Position), 'Tahun Masuk*' (Start Year), and 'Tahun Selesai*' (End Year). Below the fields are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

Dan jika berhasil menambah data akan sebagai berikut :

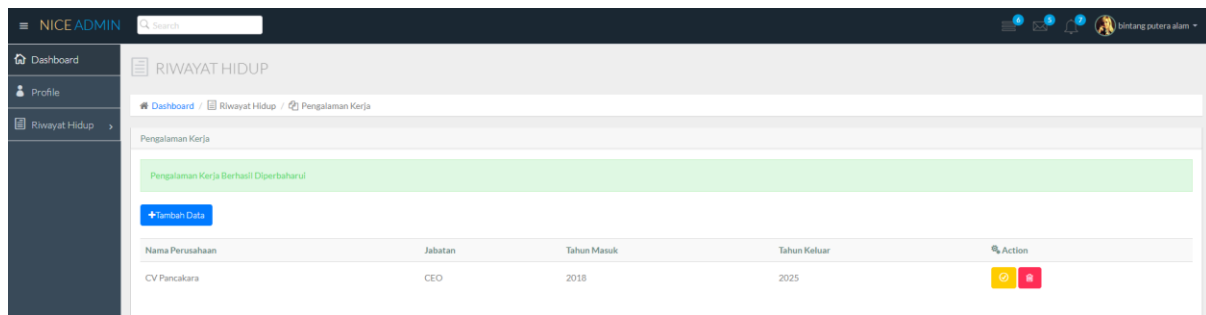
The screenshot shows the 'Pengalaman Kerja' (Work Experience) table in the NICE ADMIN system. A green message bar at the top indicates 'Data Pengalaman kerja baru telah berhasil disimpan' (New work experience data has been successfully saved). Below the message is a '+ Tambah Data' (Add Data) button. The table has five columns: 'Nama Perusahaan', 'Jabatan', 'Tahun Masuk', 'Tahun Keluar', and 'Action'. The first row of data shows 'CV Pancakara' as the company, 'CEO' as the position, '2018' as the start year, and '2021' as the end year. The 'Action' column contains a yellow edit button and a red delete button.

Nama Perusahaan	Jabatan	Tahun Masuk	Tahun Keluar	Action
CV Pancakara	CEO	2018	2021	 

Untuk merubah data klik button bewarna kuning pada salah satu data dan tampilan sebagai berikut :

The screenshot shows the 'Mengubah Pengalaman Kerja' (Edit Work Experience) form in the NICE ADMIN system. The form is located under the 'RIWAYAT HIDUP' (Resume) section. It contains four input fields: 'Nama Perusahaan*' (Company Name), 'Jabatan*' (Position), 'Tahun Masuk*' (Start Year), and 'Tahun Selesai*' (End Year). Below the fields are two buttons: 'Simpan' (Save) and 'Kembali' (Back). The form is pre-filled with the data from the table in the previous screenshot: 'CV Pancakara', 'CEO', '2018', and '2021'.

Dan jika berhasil merubah data tampilan akan sebagai berikut :



F. Membuat Tampilan Data

Untuk membuat tampilan data kita harus membuat tampilan pada view dan membuat fungsi pada controller, untuk syntax dan tampilan sebagai berikut :

```
@include('layouts.template')
<!--main content start-->
<section id="main-content">
    <section class="wrapper">
        <div class="row">
            <div class="col-lg-12">
                <h3 class="page-
header"><i class="icon_document_alt"></i>Riwayat Hidup</h3>
                <ol class="breadcrumb">
                    <li><i class="fa fa-
home"></i><a href="{{ url('/') }}">Dashboard</a></li>
                    <li><i class="icon_document_alt"></i>Riwayat Hidup</li>
                    <li><i class="fa fa-files-o"></i>Pengalaman Kerja</li>
                </ol>
            </div>
        </div>
        <!-- page start-->
        <div class="row">
            <div class="col-lg-12">
                <section class="panel">
                    <header class="panel-heading">
                        Pengalaman Kerja
                    </header>
                    <div class="panel-body">
                        @if ($message = Session::get('success'))
                            <div class="alert alert-success">
                                <p>{{ $message }}</p>
                            </div>
                        @endif
                        <a href="{{ route('pengalaman.create') }}">
```

```

        <button class="btn btn-primary"><i class="fa fa-
plus"></i>Tambah Data</button>
    </a>

    <div style="margin-top: 20px">
        <table class="table table-striped table-advance table-
hover">

            <tbody>
                <tr>
                    <th>Nama Perusahaan</th>
                    <th>Jabatan</th>
                    <th>Tahun Masuk</th>
                    <th> Tahun Keluar</th>
                    <th><i class="icon_cogs"></i> Action</th>
                </tr>
                @foreach ($data as $item)
                    <tr>
                        <td> {{ $item->nama }} </td>
                        <td> {{ $item->jabatan }} </td>
                        <td> {{ $item->tahun_masuk }} </td>
                        <td> {{ $item->tahun_keluar }} </td>
                        <td>
                            <div class="btn-group">
                                <form action=" {{ route('pengalaman.de
stroy', $item->id) }} " method="POST">
                                    <a class="btn btn-
warning" href=" {{ route('pengalaman.edit',$item-
>id) }} "><i class="icon_check_alt2"></i></a>
                                    @csrf
                                    @method('DELETE')
                                    <button type="submit" class="btn b
tn-
danger" onclick="return confirm('Apakah Anda Yakin ingin menghapus data ini ?'
)"><i class="fa fa-trash-o"></i></button>
                                    {{-- <a class="btn btn-
danger" href="#"><i class="icon_close_alt2"></i></a> --}}
                                </form>
                            </div>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</div>
</section>
</div>
</div>

```

```

        <!-- page end-->
    </section>
</section>
<!--main content end-->

```



Untuk syntax pada controller :

```

public function index()
{
    $this->param['data'] = PengalamanKerja::all();
    return view('backend.pengalaman.index', $this->param);
}

```

Dan untuk tampilannya sebagai berikut :

Nama Perusahaan	Jabatan	Tahun Masuk	Tahun Keluar	Action
CV Pancakara	CEO	2018	2025	 

G. Membuat Hapus data

Untuk menghapus hampir sama seperti edit, tetapi ada sedikit perbedaan dengan penambahan javascript, sehingga untuk syntax dan tampilan sebagai berikut :

syntax :

```

<form action="{{ route('pengalaman.destroy', $item->id) }}" method="POST">
    <a class="btn btn-warning" href="{{ route('pengalaman.edit',$item->id) }}"><i class="icon_check_alt2"></i></a>
    @csrf
    @method('DELETE')
    <button type="submit" class="btn btn-danger" onclick="return confirm('Apakah Anda Yakin ingin menghapus data ini ?')"><i class="fa fa-trash-o"></i></button>
    {{-- <a class="btn btn-danger" href="#"><i class="icon_close_alt2"></i></a> --}}
</form>





```

Untuk controller :

```
public function destroy($id)
{
    $deletePengalaman = PengalamanKerja::findOrFail($id);
    $deletePengalaman->delete();
    return redirect()->route('pengalaman')-
>with('success','Data Pengalaman Kerja berhasil dihapus');
}
```

Sehingga jika berhasil menghapus data sebagai berikut :

The screenshot shows the NICE ADMIN dashboard. A confirmation dialog box is displayed at the top right, asking "Apakah Anda Yakin ingin menghapus data ini ?" with "OK" and "Cancel" buttons. Below the dialog, a green message box states "Data Pengalaman kerja baru telah berhasil di simpan". The main content area shows a table of work experience data.

Nama Perusahaan	Jabatan	Tahun Masuk	Tahun Keluar	Action
CV Pancakara	CEO	2018	2025	 
CV Bintang Terang	CEO	2019	2026	 

The screenshot shows the NICE ADMIN dashboard after deleting data. A green message box states "Data Pengalaman Kerja berhasil dihapus". The main content area shows a table of work experience data with only one row.

Nama Perusahaan	Jabatan	Tahun Masuk	Tahun Keluar	Action
CV Pancakara	CEO	2018	2025	