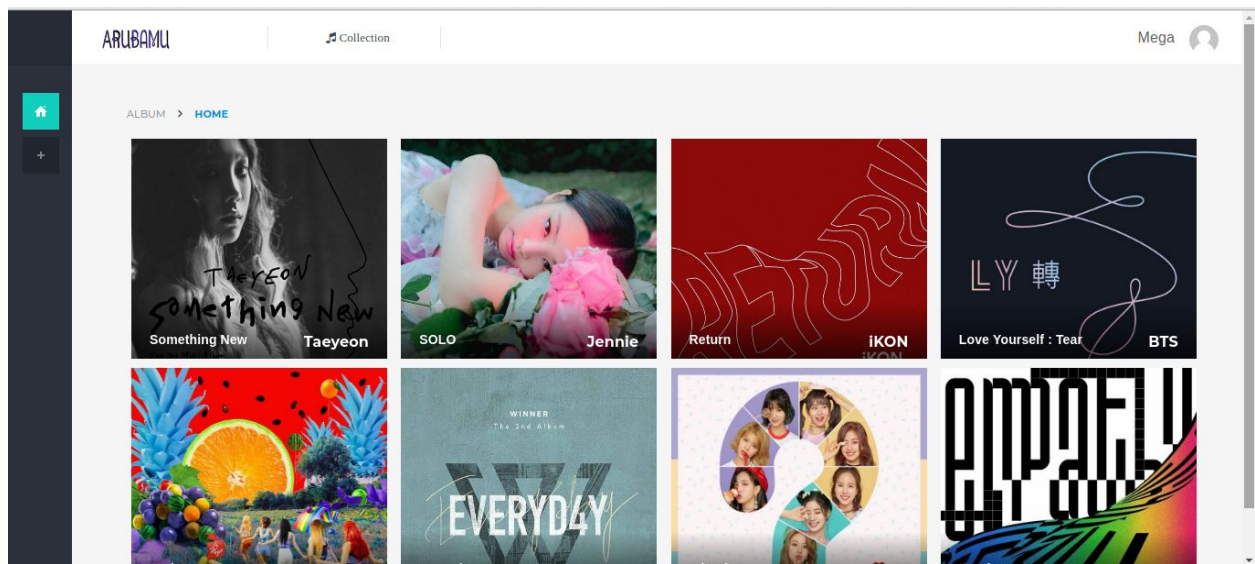


SMK Negeri 10

Jl. Mayjen Sutoyo
Jakarta Timur

Tutorial CRUD

ALBUM



By Mega Tri Handayani

-

DAFTAR ISI	2
Membuat Laravel	3
Membuat Database di MongoDB	3
Konfigurasi .env	3
Konek Laravel dengan MongoDB	4
Install Laravel MongoDB Package	5
Define Providers	5
Membuat Model	6
Membuat Controller dan Migration	6
Membuat CREATE & READ	6
Membuat UPDATE (edit)	9
Membuat DELETE	10
Application	11
Home	11
Index (READ)	11
CREATE	12
Edit (UPDATE)	12
DELETE	13
REPORT	13

Software yang perlu disiapkan :

- MongoDB
- Sublime text
- Google Chrome
- Terminal

Membuat Laravel

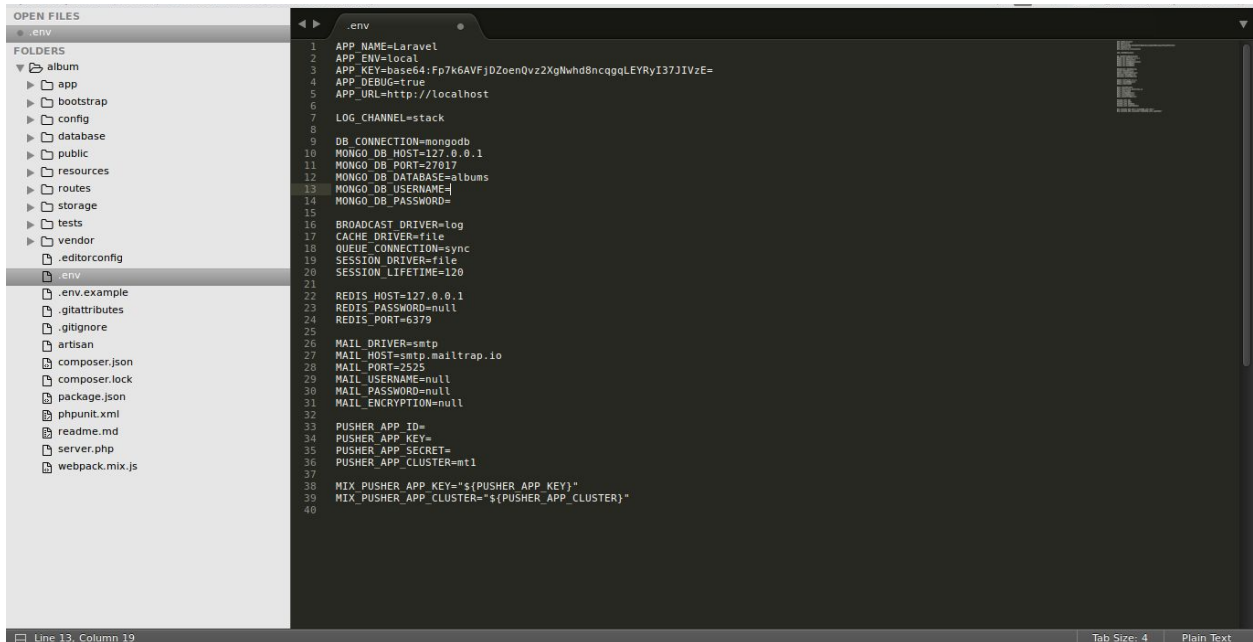
```
mega@mega-x550ze: ~/album  
mega@mega-x550ze:~$ composer create-project --prefer-dist laravel/laravel album
```

Membuat Database di MongoDB

```
mega@mega-x550ze:~$ mongo  
MongoDB shell version: 3.2.21  
connecting to: test  
> use albums  
switched to db albums
```

Konfigurasi .env

Buka file .env



```
.env  
1 APP_NAME=Laravel  
2 APP_ENV=local  
3 APP_KEY=base64:Fp7k6AVfjDZoenQvzZXgNwhd8ncqgqLEYRyI37JIVzE=  
4 APP_DEBUG=true  
5 APP_URL=http://localhost  
6  
7 LOG_CHANNEL=stack  
8  
9 DB_CONNECTION=mongodb  
10 MONGO_DB_HOST=127.0.0.1  
11 MONGO_DB_PORT=27017  
12 MONGO_DB_DATABASE=albums  
13 MONGO_DB_USERNAME=  
14 MONGO_DB_PASSWORD=  
15  
16 BROADCAST_DRIVER=log  
17 CACHE_DRIVER=file  
18 QUEUE_CONNECTION=sync  
19 SESSION_DRIVER=file  
20 SESSION_LIFETIME=120  
21  
22 REDIS_HOST=127.0.0.1  
23 REDIS_PASSWORD=null  
24 REDIS_PORT=6379  
25  
26 MAIL_DRIVER=smtp  
27 MAIL_HOST=smtp.mailtrap.io  
28 MAIL_PORT=2525  
29 MAIL_USERNAME=null  
30 MAIL_PASSWORD=null  
31 MAIL_ENCRYPTION=null  
32  
33 PUSHER_APP_ID=  
34 PUSHER_APP_KEY=  
35 PUSHER_APP_SECRET=  
36 PUSHER_APP_CLUSTER=mt1  
37  
38 MIX_PUSHER_APP_KEY='${PUSHER_APP_KEY}'  
39 MIX_PUSHER_APP_CLUSTER='${PUSHER_APP_CLUSTER}'  
40
```

Konek Laravel dengan MongoDB

1. Buka terminal dan ketikkan command

```
mega@mega-x550ze:~$ cd /etc/php/7.2/apache2
mega@mega-x550ze:/etc/php/7.2/apache2$ sudo gedit php.ini
```

2. Tambahkan extension di php.ini

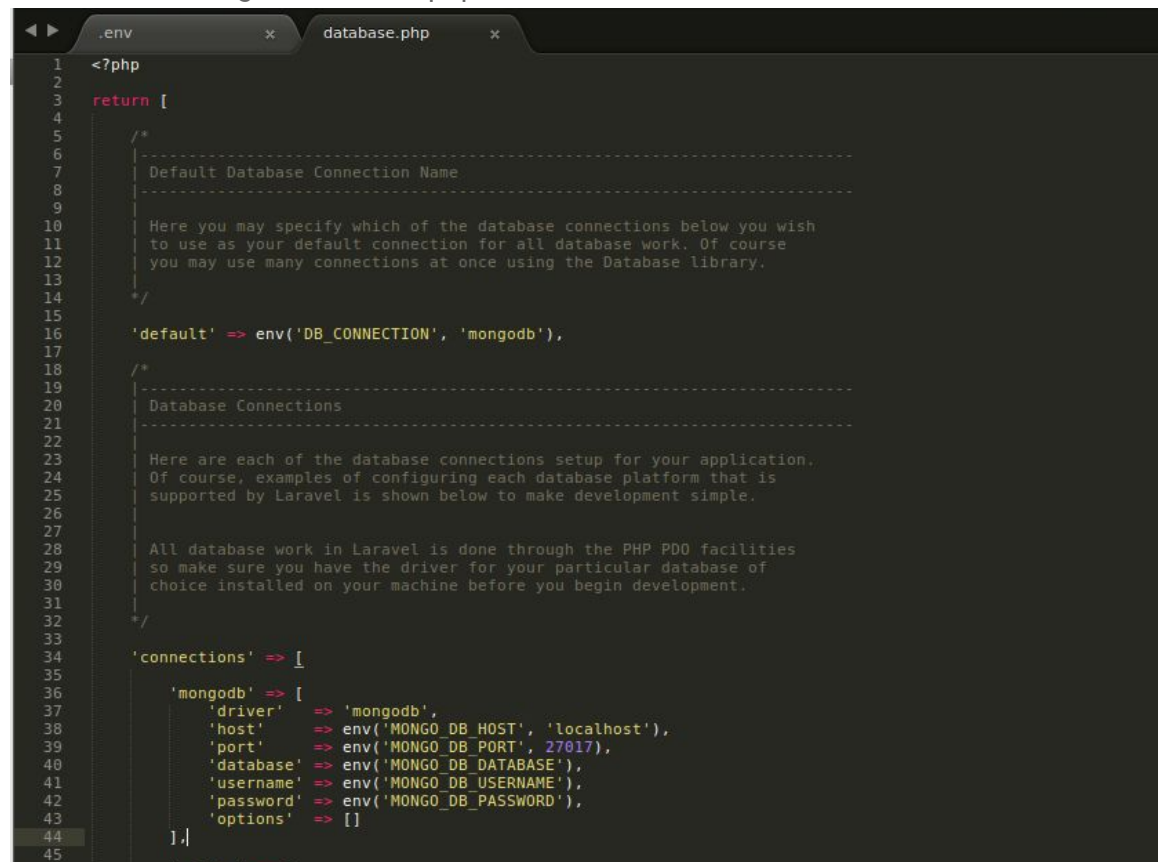


```
Open [icon] php.ini
/etc/php/7.2/apache2

;extension=soap
;extension=sockets
;extension=sqllite3
;extension=tidy
;extension=xmlrpc
;extension=xsl
;extension=php_mongo.dll
```

Kemudian save.

3. Buka file config => database.php



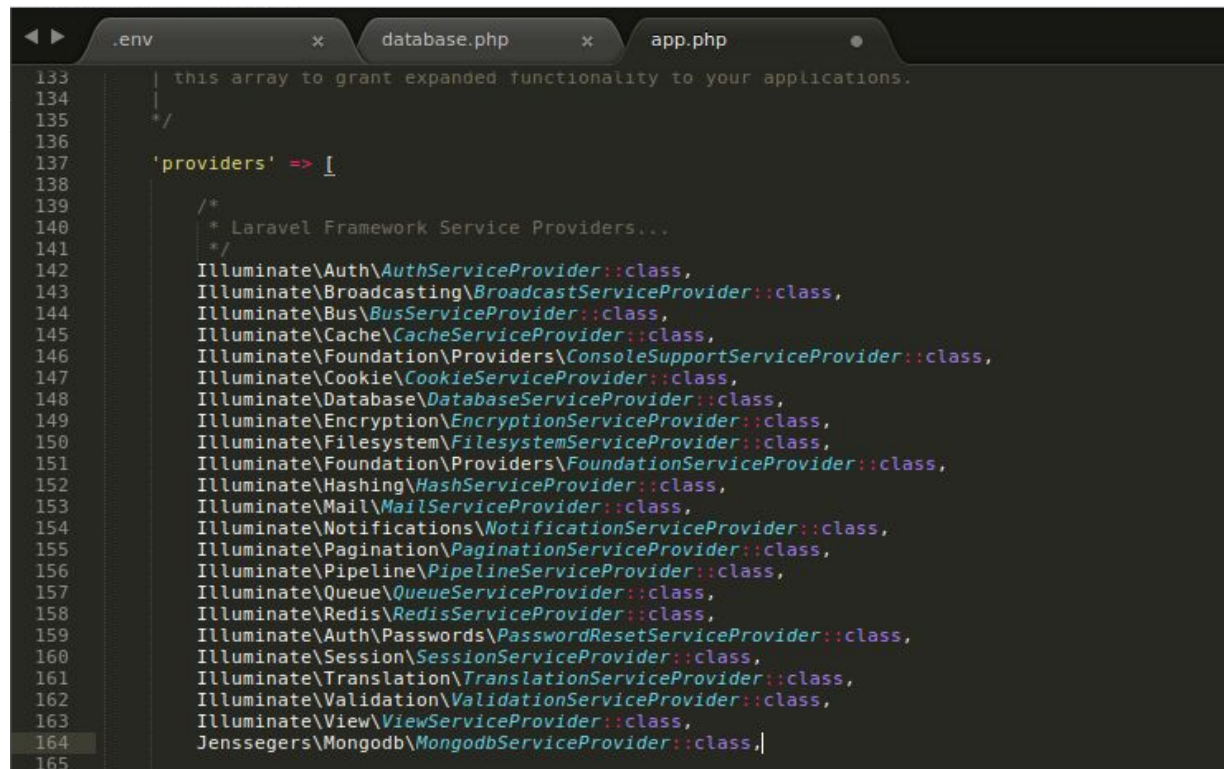
```
.env x database.php x
1 <?php
2
3 return [
4
5     /*
6     |-----
7     | Default Database Connection Name
8     |-----
9     */
10
11     | Here you may specify which of the database connections below you wish
12     | to use as your default connection for all database work. Of course
13     | you may use many connections at once using the Database library.
14
15     */
16     'default' => env('DB_CONNECTION', 'mongodb'),
17
18     /*
19     |-----
20     | Database Connections
21     |-----
22     */
23
24     | Here are each of the database connections setup for your application.
25     | Of course, examples of configuring each database platform that is
26     | supported by Laravel is shown below to make development simple.
27
28     | All database work in Laravel is done through the PHP PDO facilities
29     | so make sure you have the driver for your particular database of
30     | choice installed on your machine before you begin development.
31
32     */
33
34     'connections' => [
35
36         'mongodb' => [
37             'driver' => 'mongodb',
38             'host' => env('MONGO_DB_HOST', 'localhost'),
39             'port' => env('MONGO_DB_PORT', 27017),
40             'database' => env('MONGO_DB_DATABASE'),
41             'username' => env('MONGO_DB_USERNAME'),
42             'password' => env('MONGO_DB_PASSWORD'),
43             'options' => []
44         ],
45     ],
46 ];
```

Install Laravel MongoDB Package

```
mega@mega-x550ze: ~/album
mega@mega-x550ze:~$ cd album
mega@mega-x550ze:~/album$ composer require jenssegers/mongodb "3.4.*"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Installing mongodb/mongodb (1.4.2): Downloading (100%)
  - Installing jenssegers/mongodb (v3.4.5): Downloading (100%)
jenssegers/mongodb suggests installing jenssegers/mongodb-session (Add MongoDB session support to Laravel-MongoDB)
jenssegers/mongodb suggests installing jenssegers/mongodb-sentry (Add Sentry support to Laravel-MongoDB)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: jenssegers/mongodb
Discovered Package: laravel/nexmo-notification-channel
Discovered Package: laravel/slack-notification-channel
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
mega@mega-x550ze:~/album$
```

Define Providers

Buka file config => app.php



```
.env x database.php x app.php
133 | this array to grant expanded functionality to your applications.
134 |
135 | */
136 |
137 | 'providers' => [
138 |
139 |     /*
140 |      * Laravel Framework Service Providers...
141 |      */
142 |     Illuminate\Auth\AuthServiceProvider::class,
143 |     Illuminate\Broadcasting\BroadcastServiceProvider::class,
144 |     Illuminate\Bus\BusServiceProvider::class,
145 |     Illuminate\Cache\CacheServiceProvider::class,
146 |     Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
147 |     Illuminate\Cookie\CookieServiceProvider::class,
148 |     Illuminate\Database\DatabaseServiceProvider::class,
149 |     Illuminate\Encryption\EncryptionServiceProvider::class,
150 |     Illuminate\Filesystem\FilesystemServiceProvider::class,
151 |     Illuminate\Foundation\Providers\FoundationServiceProvider::class,
152 |     Illuminate\Hashing\HashServiceProvider::class,
153 |     Illuminate\Mail\MailServiceProvider::class,
154 |     Illuminate\Notifications\NotificationServiceProvider::class,
155 |     Illuminate\Pagination\PaginationServiceProvider::class,
156 |     Illuminate\Pipeline\PipelineServiceProvider::class,
157 |     Illuminate\Queue\QueueServiceProvider::class,
158 |     Illuminate\Redis\RedisServiceProvider::class,
159 |     Illuminate\Auth\Passwords\PasswordResetServiceProvider::class,
160 |     Illuminate\Session\SessionServiceProvider::class,
161 |     Illuminate\Translation\TranslationServiceProvider::class,
162 |     Illuminate\Validation\ValidationServiceProvider::class,
163 |     Illuminate\View\ViewServiceProvider::class,
164 |     Jenssegers\Mongodb\MongodbServiceProvider::class,
165 | ]
```


Membuat Model

```
mega@mega-x550ze: ~/album
mega@mega-x550ze:~/album$ php artisan make:model Album
Model created successfully.
```

Membuat Controller dan Migration

```
mega@mega-x550ze:~/album$ php artisan make:controller AlbumController
Controller created successfully.

mega@mega-x550ze:~/album$ php artisan make:migration create_albums_table
Created Migration: 2018_12_08_091318_create_albums_table
```

Jika ingin membuat controller dan migration bersamaan, bisa diketik seperti ini
`php artisan make:controller AlbumController -m`

Membuat CREATE & READ

1. Buat folder **album** di **resources => views** dan didalamnya buat file **add.blade.php** (Tampilan untuk create)

```
add.blade.php
1 |@extends('layouts.apps')
2 |
3 |@section('content')
4 |<ul class="breadcrumb">
5 |<li>
6 |<a href="{{ url('home') }}">Album</a>
7 |</li>
8 |<li>
9 |<a class="active">Add Album</a>
10|</li>
11|</ul>
12|<button class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">
13|Add New Album
14|</button>
15|<a href="{{ url('album/pdf') }}">
16|<button class="btn btn-primary btn-lg pull-right">
17|PDF
18|</button>
19|</a>
20|<!-- START PAGE CONTENT -->
21|
22|<!-- MODAL STICK UP -->
23|<div class="modal fade stick-up in" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModallabel" aria-hidden="true">
24|<div class="modal-dialog">
25|<div class="modal-content">
26|<div class="modal-header clearfix text-left">
27|<button type="button" class="close" data-dismiss="modal" aria-hidden="true"><i class="pg-close fs-14"></i>
28|</button>
29|<h5>Add <span class="semi-bold">New Album</span></h5>
30|<p>Enjoy for your experience</p>
31|</div>
32|<div class="modal-body">
33|<form role="form" method="POST" autocomplete="off" action="{{ route('save') }}" enctype="multipart/form-data">
34|{{ csrf_field() }}
35|
36|<div class="form-group-attached">
37|<div class="row">
38|<div class="col-sm-12">
39|<div class="form-group form-group-default">
40|<label>Name of Album</label>
41|<input type="text" class="form-control" name="name" required="">
42|</div>
43|</div>
44|<div class="row">
45|<div class="col-sm-8">
46|<div class="form-group form-group-default">
47|<label>Artist</label>
48|<input type="text" class="form-control" name="artist">
49|</div>
```

```

49         <input type="text" class="form-control" name="artist">
50     </div>
51 </div>
52 <div class="col-sm-4">
53     <div class="form-group form-group-default form-group-default-select2">
54         <label>Category Artist</label>
55         <select class="full-width select2-offscreen" placeholder-data="" data-init-plugin="select2" tabindex="-1" title=""
56             name="category_artist">
57             <option value="boyband">Boyband</option>
58             <option value="girlband">Girlband</option>
59             <option value="solo">Solo</option>
60         </select>
61     </div>
62 </div>
63 <div class="row">
64     <div class="col-sm-5">
65         <div class="form-group form-group-default input-group col-sm-10">
66             <label>Release Date</label>
67             <input type="date" class="form-control" id="datepicker-component2" name="year">
68         </div>
69     </div>
70     <div class="col-sm-7">
71         <div class="form-group form-group-default form-group-default-select2">
72             <label>Genre Music</label>
73             <select class="full-width select2-offscreen" placeholder-data="" data-init-plugin="select2" tabindex="-1" title=""
74                 name="genre_music">
75                 <option>Pope</option>
76                 <option>Hip Hop</option>
77                 <option>Rock</option>
78                 <option>RnB</option>
79                 <option>Jazz</option>
80                 <option>Other</option>
81             </select>
82         </div>
83     </div>
84 <div class="row">
85     <div class="col-sm-12">
86         <div class="form-group form-group-default input-group col-sm-10">
87             <label>Main Track</label>
88             <input type="text" class="form-control" name="track"></input>
89             <span class="input-group-addon">
90                 <i class="fa fa-add"></i>
91             </span>
92         </div>
93     </div>
94 </div>
95 <div class="row">
96     <input type="file" name="img"></input>
97 </div>
98 </div>
99 <div class="row">
100     <div class="col-sm-4 m-t-10 sm-m-t-10 pull-right">
101         <button type="submit" class="btn btn-primary btn-block m-t-5">Submit</button>
102     </div>
103 </div>
104 </div>
105 </div>
106 <!-- /.modal-content -->
107 </div>
108 <!-- /.modal-dialog -->
109 </div>
110 <!-- END MODAL STICK UP -->
111

```

(Tampilan untuk Read)

```

132 <!-- Index -->
133 <div class="table-responsive">
134     <div id="basicTable_wrapper" class="dataTables_wrapper form-inline no-footer">
135         <table class="table table-hover dataTable no-footer" id="basicTable" role="grid">
136             <thead>
137                 <tr role="row">
138                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Name of Album</th>
139                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Artist</th>
140                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Category Artist</th>
141                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Release Date</th>
142                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Genre Music</th>
143                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Main Track</th>
144                     <th style="width: 182px;" class="sorting_desc" tabindex="0" aria-controls="basicTable" rowspan="1" colspan="1" aria-sort="descending" aria-label="Title: activate to sort column ascending">Action</th>
145                 </tr>
146             </thead>
147             <tbody>
148                 <tr role="row" class="odd">
149                     <td class="v-align-middle sorting_1">
150                         <p>{{ $key->name }}</p>
151                     </td>
152                     <td class="v-align-middle">
153                         <p>{{ $key->artist }}</p>
154                     </td>
155                     <td class="v-align-middle">
156                         <p>{{ $key->category_artist }}</p>
157                     </td>
158                     <td class="v-align-middle">
159                         <p>{{ $key->year }}</p>
160                     </td>
161                     <td class="v-align-middle">
162                         <p>{{ $key->genre_music }}</p>
163                     </td>
164                     <td class="v-align-middle">
165                         <p>{{ $key->track }}</p>
166                     </td>
167                 </tr>
168             </tbody>
169         </table>
170     </div>
171

```

```

172 <td class="v-align-middle">
173 <a href="{{ route('edit', $key->id) }}"><button class="btn btn-info"><i class="fa fa-search"></i></button></a>
174
175 <button class="btn btn-danger" data-toggle="modal" data-target="#modal"><i class="fa fa-trash-o"></i></button></button>
176 <div class="modal fade bd-example-modal-sm id="modal" tabindex="-1" role="dialog" aria-labelledby="mySmallModalLabel" aria-hidden="true"
177 >
178 <div class="modal-dialog modal-sm">
179 <div class="modal-content">
180 <div class="modal-header">
181 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
182 <span aria-hidden="true"><span></span></button>
183 <div class="modal-title">Confirmation</div>
184 <div class="modal-body text-center m-t-20">
185 <div class="no-margin p-b-10" style="color: red;">Are You Sure Want To Delete?</div>
186 <div class="modal-footer">
187 <button type="button" class="btn btn-default pull-left" data-dismiss="modal">Cancel</button>
188 <a href="{{ url('delete/'.$key->id) }}">
189 <button type="button" class="btn btn-primary">Yes</button>
190 </a>
191 </div>
192 </div>
193 </div>
194 </div>
195 </td>
196 </tr>
197 </tbody>
198 @endforeach
199 </table>
200 </div>
201 </div>
202 </div>

```

2. Buka AlbumController.php

```

AlbumController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4 use Jenssegers\Mongodb\Eloquent\Model as Eloquent;
5 use Illuminate\Http\Request;
6 use Storage;
7 use App\Album;
8 use PDF;
9
10 class AlbumController extends Controller
11 {
12
13     public function add()
14     {
15         $data['album'] = Album::get();
16         return view('album.add')->with($data);
17     }
18
19     public function save(Request $r)
20     {
21         $file = $r->file('img');
22         $album = new \App\Album;
23         $album->name = $r->get('name');
24         $album->artist = $r->get('artist');
25         $album->category_artist = $r->get('category_artist');
26         $album->year = $r->get('year');
27         $album->genre_music = $r->get('genre_music');
28         $album->track = $r->get('track');
29         $album->img = $file->getClientOriginalName();
30         $file->move('foto/', $file->getClientOriginalName());
31
32         $album->save();
33         return redirect(url('add'));
34     }
35 }

```

3. Buka route => web.php

```

25 Route::get('add', 'AlbumController@add')->name('add');
26 Route::post('save', 'AlbumController@save')->name('save');

```


Membuat UPDATE (edit)

1. Buat file di folder **Album** dengan nama **edit.blade.php**

```
1 |@extends('layouts.apps')
2
3 |@section('content')
4 |<ul class="breadcrumb">
5 |    <li><a href="{{ url('home') }}">Album</a>
6 |    </li>
7 |    <li class="active">Update Album</li>
8 |</ul>
9
10 |</ul>
11
12 |<div class="row">
13 |    <div class="col-sm-4">
14 |        
15 |    </div>
16 |    <div class="col-sm-8">
17 |        <form class="form-horizontal" method="POST" role="form" action="{{ route('update', $album->id) }}">
18 |            {{ csrf_field() }}
19 |            <div class="form-group">
20 |                <label for="position" class="col-sm-3 control-label">Name of Album</label>
21 |                <div class="col-sm-9">
22 |                    <input type="text" name="name" class="form-control" {{ $errors->has('name') ? ' is-invalid' : '' }} id="position" value="{{ $album->name }}" required="" aria-required="true">
23 |                    @if ($errors->has('name'))
24 |                        <span class="invalid-feedback" role="alert">
25 |                            <strong>{{ $errors->first('name') }}</strong>
26 |                        </span>
27 |                    @endif
28 |                </div>
29 |            </div>
30 |            <div class="form-group">
31 |                <label for="position" class="col-sm-3 control-label">Artist</label>
32 |                <div class="col-sm-9">
33 |                    <input type="text" name="artist" class="form-control" id="position" value="{{ $album->artist }}" required="" aria-required="true">
34 |                </div>
35 |            </div>
36 |            <div class="form-group">
37 |                <label for="position" class="col-sm-3 control-label">Category Artist</label>
38 |                <div class="col-sm-9">
39 |                    <div class="select2-container full-width" id="s2id_autogen1">
40 |                        <select class="full-width select2-offscreen" name="category_artist" data-init-plugin="select2" tabindex="-1" title="" value="{{ $album->category_artist }}">
41 |                            <option value="boyband">Boyband</option>
42 |                            <option value="girlband">Girlband</option>
43 |                            <option value="solo">Solo</option>
44 |                        </select>
45 |                    </div>
46 |                </div>
47 |            </div>
48 |            <div class="form-group">
49 |                <label for="position" class="col-sm-3 control-label">Release Date</label>
50 |                <div class="col-sm-9">
51 |                    <input type="date" class="form-control" name="year" id="position" value="{{ $album->year }}" required="" aria-required="true">
52 |                </div>
53 |            </div>
54 |            <div class="form-group">
55 |                <label for="position" class="col-sm-3 control-label">Genre Music</label>
56 |                <div class="col-sm-9">
57 |                    <div class="select2-container full-width" id="s2id_autogen1">
58 |                        <select class="full-width select2-offscreen" name="genre_music" data-init-plugin="select2" tabindex="-1" title="" value="{{ $album->genre_music }}">
59 |                            <option>Pop</option>
60 |                            <option>Hip Hop</option>
61 |                            <option>Rock</option>
62 |                            <option>RnB</option>
63 |                            <option>Jazz</option>
64 |                            <option>Other</option>
65 |                        </select>
66 |                    </div>
67 |                </div>
68 |            </div>
69 |            <div class="form-group">
70 |                <label for="position" class="col-sm-3 control-label">Main Track</label>
71 |                <div class="col-sm-9">
72 |                    <input type="text" class="form-control" name="track" id="position" value="{{ $album->track }}" required="">
73 |                </div>
74 |            </div>
75 |        </div>
76 |        <div class="row">
77 |            <button class="btn btn-info pull-right" type="submit">Submit</button>
78 |        </div>
79 |    </div>
80 |</div>
81
82 |@endsection
```

2. Buka AlbumController

```
36     public function edit($id)
37     {
38         $data ['album'] = Album::findorFail($id);
39         return view('album.edit')->with($data);
40     }
41
42     public function update(Request $request, $id)
43     {
44         $album = Album::find($id);
45         $album->name = $request->input('name');
46         $album->artist = $request->input('artist');
47         $album->category_artist = $request->input('category_artist');
48         $album->year = $request->input('year');
49         $album->genre_music = $request->input('genre_music');
50         $album->track = $request->input('track');
51
52
53
54         $album->save();
55         return redirect(url('add'));
56     }
57 }
```

3. Buka web.php

```
27
28 Route::get('{id}', 'AlbumController@edit')->name('edit');
29 Route::post('{id}', 'AlbumController@update')->name('update');
30
```

Membuat DELETE

1. Buka AlbumController

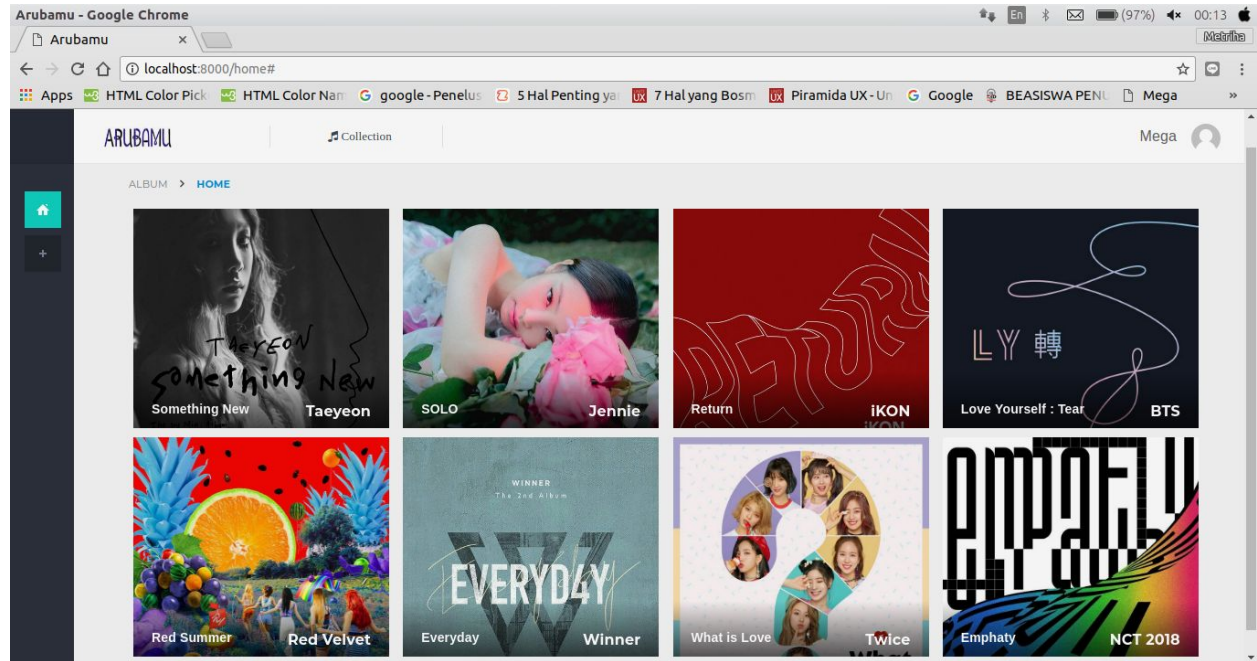
```
57
58     public function delete($id)
59     {
60         $album = \App\Album::find($id);
61         $album->delete();
62
63         return redirect('add');
64     }
65 }
```

2. Buka web.php

```
31
32 Route::get('delete/{id}', 'AlbumController@delete');
33
```

Application

Home



Index (READ)

The screenshot shows the 'Index (READ)' page of the Arubamu application. The browser address bar indicates the URL is localhost:8000/add. The page features a table of albums with the following data:

NAME OF ALBUM	ARTIST	CATEGORY ARTIST	REALASE DATE	GENRE MUSIC	MAIN TRACK	ACTION
Something New	Taeyeon	solo	2018-06-18	Other	Something New	Search Delete
SOLO	Jennie	boyband	2018-11-12	Pop	Solo	Search Delete
Return	iKON	boyband	2018-01-25	Pop	Love Scenario	Search Delete
Love Yourself : Tear	BTS	boyband	2018-05-18	Pop	Fake Love	Search Delete
Red Summer	Red Velvet	girlband	2017-07-08	Pop	Red Flavor	Search Delete

At the top of the table, there is a button 'Add New Album' and a 'PDF' button. At the bottom right, there is a 'Trash' button.

CREATE

Arubamu - Google Chrome

localhost:8000/add

Arubamu

ALBUM > ADD ALBUM

Add New Album

NAME OF ALBUM

ARTIST

CATEGORY ARTIST
Boyband

RELEASE DATE
mm/dd/yyyy











GENRE MUSIC
Pop

MAIN TRACK

Choose File No file chosen

Submit

PDF

NAME OF ALBUM	ARTIST	MAIN TRACK	ACTION
Something New	Taeyeon	Something New	 
SOLO	Jennie	Solo	 
Return	IKON	Love Scenario	 
Love Yourself : Tear	BTS	Fake Love	 
Red Summer	Red Velvet	Red Flavor	 

Edit (UPDATE)

Arubamu - Google Chrome


localhost:8000/5c111fcc20fab72c6c71ac28

Arubamu

ALBUM > UPDATE ALBUM

Collection

Mega



NAME OF ALBUM
SOLO

ARTIST
Jennie

CATEGORY ARTIST
Boyband

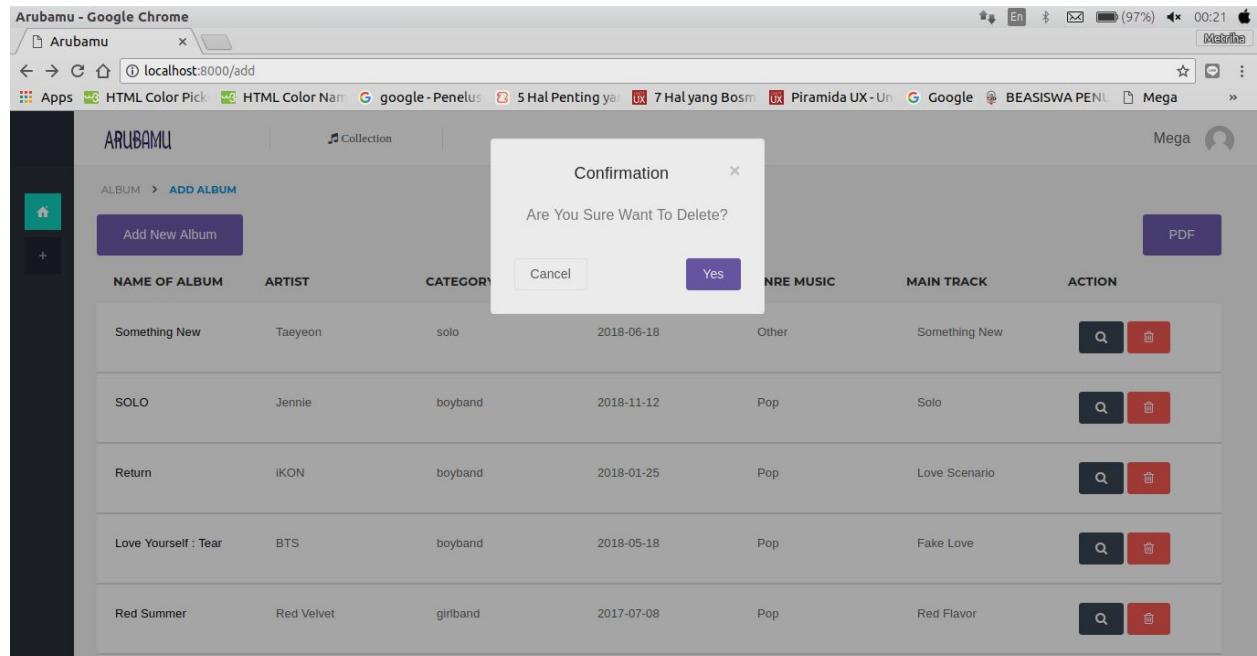
RELEASE DATE
11/12/2018

GENRE MUSIC
Pop

MAIN TRACK
Solo

Submit

DELETE



REPORT

