

## LEMBAR JAWABAN UJIAN

NIM : 2019114033  
NAMA : Aulia Apriliani Mulyadi  
Mata Kuliah : Pemrograman Internet Lanjut  
Tanggal Ujian : Kamis, 25 November 2021  
Dosen : Reza Ilyasa, M.Kom.

---

Link Github : [https://github.com/auliaaprm/uts\\_pil\\_laravel.git](https://github.com/auliaaprm/uts_pil_laravel.git)

### 1. Penjelasan coding tambahan.

- a) Menambahkan variabel `$jumlah_anggota` pada bagian public function `show` di `GroupsController`. Dalam variabel ini, akan dipanggil tabel `friends` dengan kondisi dimana atribut `groups_id` sama dengan atribut `id` yang ada pada tabel `groups` kemudian menggunakan method `get` untuk mengambil nilainya dan menggunakan method `count` untuk menghitung jumlah nilai yang didapat. Tujuan tambahan ini adalah untuk menampilkan jumlah anggota saat ini yang ada pada grup.

```
public function show($id)
{
    $groups = Groups::where('id', $id)->first();
    $jumlah_anggota = Friends::where('groups_id', '=', $groups->id)->get()->count();
    return view('groups.show', ['group' => $groups, 'jumlah_anggota' => $jumlah_anggota]);
}
```

- b) Menambahkan model Riwayat dan tujuannya untuk memanggil tabel `riwayat` pada database. Didalam model ini terdapat variabel `$fillable` untuk melakukan insert atau update ke database, variabel `$timestamps` untuk menonaktifkan fitur mencatat waktu, public function `friends` dan public function untuk membuat relasi one to many.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Riwayat extends Model
{
    use HasFactory;
    protected $fillable = [
        'friends_id', 'groups_id', 'activity', 'date'
    ];

    protected $table = 'riwayat';
    public $timestamps = false;

    public function friends()
    {
        return $this->belongsTo(Friends::class, 'friends_id', 'id');
    }

    public function groups()
    {
        return $this->belongsTo(Groups::class, 'groups_id', 'id');
    }
}
```

- c) Menambahkan variabel riwayatats pada bagian public function *show* di CobaController. Dalam variabel ini, dipanggil tabel *riwayat* dan terdapat fungsi *with* untuk memanggil tabel *groups* dan *friends* untuk bergabung, dengan kondisi dimana atribut *friends\_id* pada tabel riwayat sama dengan atribut *id* pada tabel *friends*. Kemudian menggunakan method *get* untuk mengambil nilai. Tujuan tambahan ini adalah untuk mengambil nilai riwayat aktivitas anggota pada grup.

```
public function show($id)
{
    $friends = Friends::with('groups')->where('id', $id)->first();
    $riwayats = Riwayat::with('friends', 'groups')->where('friends_id', $id)->get();
    return view('friends.show', ['friend' => $friends, 'riwayats' => $riwayats]);
}
```

- d) Menambah source code pada model Friends. Ditambahkan variabel *\$fillable* untuk melakukan insert atau update ke database dan public function riwayatats untuk membuat relasi one to many antara tabel *riwayat* dengan tabel *groups*.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Friends extends Model
{
    use HasFactory;
    protected $fillable = [
        'nama', 'no_tlp', 'alamat', 'jenis_kelamin', 'instagram', 'groups_id'
    ];

    public function groups()
    {
        return $this->belongsTo(Groups::class, 'groups_id', 'id');
    }

    public function riwayatats()
    {
        return $this->hasMany(Riwayat::class);
    }
}
```

- e) Menambah source code pada model Groups. Ditambahkan variabel *\$guarded* untuk melakukan insert atau update ke database dan public function riwayatats untuk membuat relasi one to many antara tabel *riwayat* dengan tabel *friends*.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Groups extends Model
{
    use HasFactory;

    protected $guarded = ['name', 'total_users_join', 'total_users_leave'];

    public function friends()
    {
        return $this->hasMany(Friends::class);
    }

    public function riwayatats()
    {
        return $this->hasMany(Riwayat::class);
    }
}
```

- f) Menambahkan source code pada bagian public function *updateaddmember* di GroupsController. Ketika melakukan penambahan anggota baru pada group maka *groups\_id* akan menyesuaikan dengan *id groups*nya kemudian kolom *total\_users\_join* pada tabel *groups* akan bertambah 1. Untuk variabel *\$riwayat* berfungsi untuk membuat objek baru pada tabel *riwayat* dengan adanya comment penjelas pada kolom *activity* apabila '1 = masuk' dan *date* = Carbon untuk menangkap tanggal aktivitas terjadi.

```
public function updateaddmember(Request $request, $id)
{
    $friend = Friends::where('id', $request->friend_id)->first();
    Friends::find($friend->id)->update([
        'groups_id' => $id
    ]);

    $group = Groups::find($id);
    $group->total_users_join = $group->total_users_join + 1;
    $group->save();

    $riwayat = new Riwayat();
    $riwayat->friends_id = $request->friend_id;
    $riwayat->groups_id = $id;
    $riwayat->activity = 1;
    $riwayat->date = Carbon::now();
    $riwayat->save();

    return redirect('/groups');
}
```

- g) Menambahkan source code pada bagian public function *deleteaddmember* di GroupsController. Ketika melakukan penghapusan anggota pada group, kolom *total\_users\_leaves* pada tabel *groups* akan bertambah 1. Untuk variabel *\$riwayat* berfungsi untuk membuat objek baru pada tabel *riwayat* dengan adanya comment penjelas pada kolom *activity* apabila '0 = keluar' dan *date* = Carbon untuk menangkap tanggal aktivitas terjadi.

```
public function deleteaddmember(Request $request, $id)
{
    //dd($id);

    $friend = Friends::find($id);

    $group = Groups::find($friend->groups_id);
    $group->total_users_leave = $group->total_users_leave + 1;
    $group->save();

    $riwayat = new Riwayat();
    $riwayat->friends_id = $id;
    $riwayat->groups_id = $friend->groups_id;
    $riwayat->activity = 0;
    $riwayat->date = Carbon::now();
    $riwayat->save();

    $friend->groups_id = 0;
    $friend->save();

    return redirect('/groups');
}
```

- h) Menambahkan source code pada file *show.blade* di Friends untuk menampilkan nilai group saat ini dan riwayat aktivitas group pada views Friends.

```
<div class="card-body">
  <h4 class="small font-weight-bold mb-4">Nomor Telepon: <span class="float-right">{{ $friend['no_tlp'] }}</span></h4>
  <h4 class="small font-weight-bold mb-4">Alamat: <span class="float-right">{{ $friend['alamat'] }}</span></h4>
  <h4 class="small font-weight-bold mb-4">Jenis Kelamin: <span class="float-right">{{ $friend['jenis_kelamin'] }}</span></h4>
  <h4 class="small font-weight-bold mb-4">Instagram: <span class="float-right">{{ $friend['instagram'] }}</span></h4>
  <h4 class="small font-weight-bold mb-4">Anggota Grup Saat Ini: <span class="float-right">{{ $friend->groups->name ?? 'Tidak Ada' }}</span></h4>
  <div class="table-responsive">
    <table class="table">
      <thead>
        <tr>
          <th>No</th>
          <th>Grup</th>
          <th>Keterangan</th>
          <th>Tanggal</th>
        </thead>
      </thead>
      <tbody>
        <@foreach ($riwayats as $riwayat)>
          <tr>
            <td>{{ $loop->iteration }}</td>
            <td>{{ $riwayat->groups->name }}</td>
            <td>
              @if ($riwayat->activity==1)
                Masuk
              @else
                Keluar
              @endif
            </td>
            <td>{{ $riwayat->date }}</td>
          </tr>
        </foreach>
      </tbody>
    </table>
  </div>
</div>
</div>
```

- i) Menambahkan source code pada file *show.blade* di Groups untuk menampilkan nilai jumlah anggota saat ini, jumlah anggota masuk, dan jumlah anggota keluar pada views Groups.

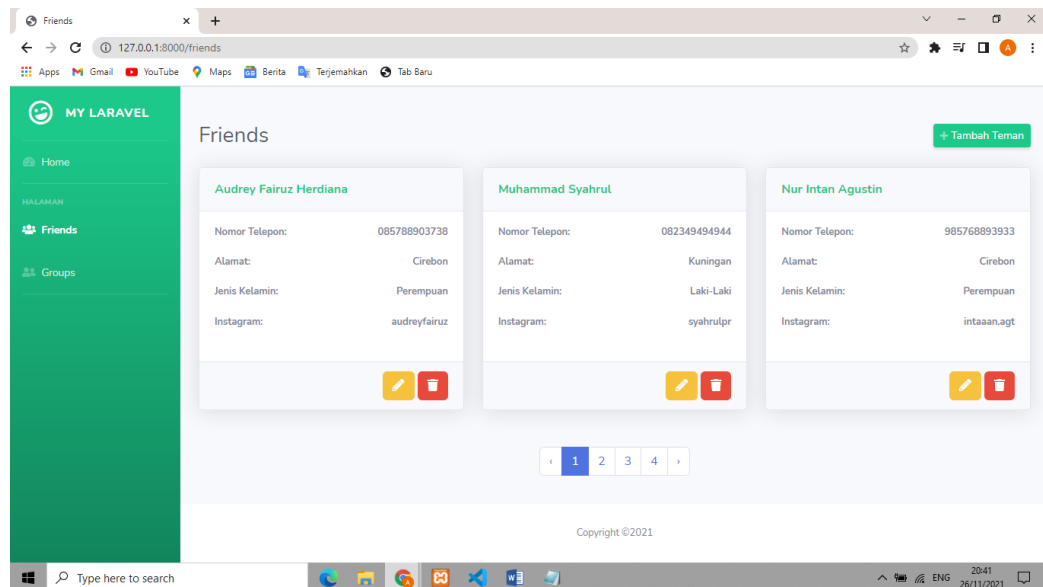
```
<div class="card-body">
  <h6>Jumlah Anggota Saat Ini: {{ $jumlah_anggota }} </h6>
  <h6>Jumlah Anggota Masuk : {{ $group['total_users_join'] }}</h6>
  <h6>Jumlah Anggota Keluar : {{ $group['total_users_leave'] }}</h6>
</div>
```

- j) Menambahkan source code sb-admin-2 agar tampilan lebih rapih dan menarik pada file *app.blade* dan beberapa views file yang dibutuhkan.

## 2. Tampilan UI

### a) Tampilan Friends

Pada view ini akan ditampilkan berupa pagination yang menampilkan daftar list teman sesuai dengan data-data yang ada pada tabel *friends* di database.



### b) Tampilan Tambah Teman

Pada view ini menampilkan form untuk menambahkan data teman baru. Terdapat lima jenis data yang dapat ditambahkan yaitu nama, nomor telepon, alamat, jenis kelamin, dan instagram. Data yang ditambahkan akan secara otomatis juga akan masuk ke tabel *friends* di database.

Tambah Teman

Nama

No Telepon

Alamat

Jenis Kelamin

Instagram

### c) Tampilan Edit Teman

Pada view ini menampilkan form untuk mengedit data teman yang sudah ada. Sama seperti form tambah data, terdapat lima jenis data yang dapat diedit yaitu nama, nomor telepon, alamat, jenis kelamin, dan instagram. Data yang diedit akan secara otomatis juga akan masuk ke tabel *friends* di database.

MY LARAVEL

Home

HALAMAN

Friends

Groups

### Edit Teman

Nama

Audrey Fairuz Herdiana

No Telepon

085788903738

Alamat

Cirebon

Jenis Kelamin

Perempuan

Instagram

audreyfairuz

Submit

Copyright ©2021

### d) Tampilan Friend Detail

Pada view ini menampilkan detail data dari setiap teman. Data yang dapat dilihat pada tampilan ini adalah nama, nomor telepon, alamat, jenis kelamin, instagram, group saat ini, dan riwayat group. Data group saat ini didapat dari *groups\_id* pada tabel *friends*, sedangkan data riwayat group didapatkan dari tabel *riwayat*.

MY LARAVEL

Home

HALAMAN

Friends

Groups

### Detail Teman

Melani Yanti

Nomor Telepon: 087564443001

Alamat: Kuningan

Jenis Kelamin: Perempuan

Instagram: melaniyanti

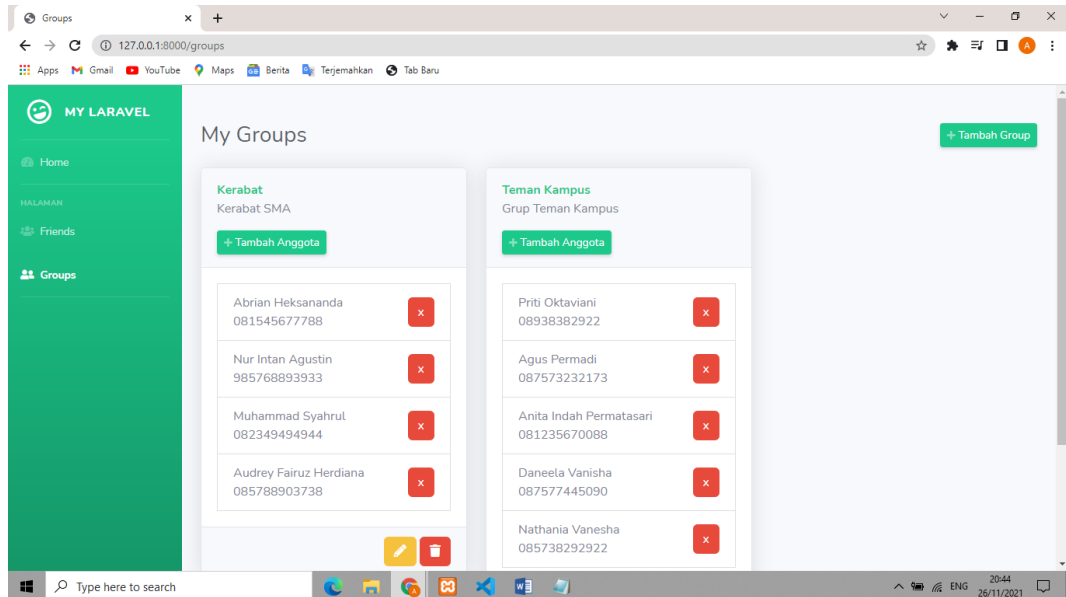
Anggota Grup Saat Ini: Tidak Ada

Riwayat Group

No	Grup	Keterangan	Tanggal
1	Kerabat	Masuk	2021-11-26 08:29:19
2	Kerabat	Keluar	2021-11-26 08:31:10

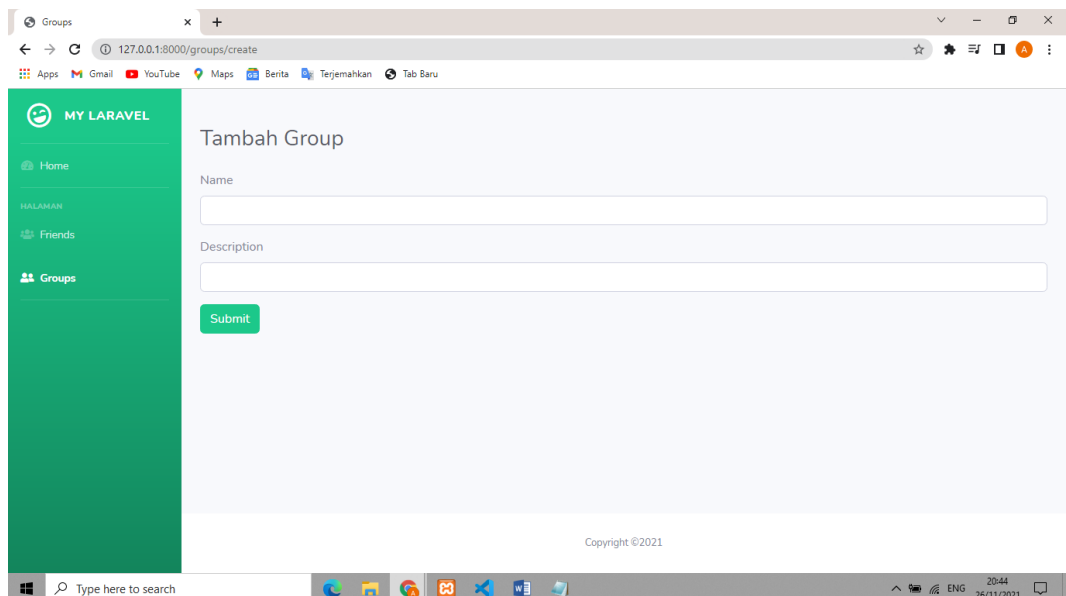
e) Tampilan Groups

Pada view ini akan ditampilkan berupa pagination yang menampilkan daftar list grup sesuai dengan data-data yang ada pada tabel groups di database.



f) Tampilan Tambah Group

Pada view ini menampilkan form untuk menambahkan data group baru. Terdapat dua jenis data yang dapat ditambahkan yaitu nama group dan deskripsi. Data yang ditambahkan akan secara otomatis juga akan masuk ke tabel *groups* di database.



### g) Tampilan Edit Group

Pada view ini menampilkan form untuk mengedit data group yang sudah ada. Sama seperti form tambah data, terdapat dua jenis data yang dapat diedit yaitu nama group dan deskripsi. Data yang diedit akan secara otomatis juga akan masuk ke tabel *groups* di database.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/groups/3/edit`. The page has a green sidebar on the left with the logo 'MY LARAVEL' and navigation links: Home, HALAMAN, Friends, and Groups. The main content area is titled 'Edit Group' and contains two text input fields: 'Name' with the value 'Kerabat' and 'Description' with the value 'Kerabat SMA'. Below these fields is a green 'Submit' button. The footer of the page says 'Copyright ©2021'.

### h) Tampilan Group Detail

Pada view ini menampilkan detail data dari setiap group. Data yang dapat dilihat pada tampilan ini adalah jumlah anggota saat ini, jumlah anggota yang masuk dan jumlah anggota yang keluar.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/groups/3`. The page has a green sidebar on the left with the logo 'MY LARAVEL' and navigation links: Home, HALAMAN, Friends, and Groups. The main content area is titled 'Detail Group' and displays a card for the group 'Kerabat' with the description 'Kerabat SMA'. Below the card, the following statistics are listed: 'Jumlah Anggota Saat Ini: 4', 'Jumlah Anggota Masuk : 5', and 'Jumlah Anggota Keluar : 1'. The footer of the page says 'Copyright ©2021'.



### i) Tampilan Tambah Anggota

Dalam view ini ditampilkan berupa form untuk menambahkan anggota baru kedalam sebuah group. Data teman yang ada pada form berasal dari hasil relasi one to many pada model Groups dan Friends.

