

# PEMBUATAN SISTEM FORM (Pertanyaan)

OLEH : LALU BAYU ALAMI

# sisnopsis

- Membuat sistem form seperti form :
- Sekolahkoding.com
- Ubuntu indoesia

**Cerita tentang sistem yang akan kita buat :**  
**“User hanya bisa melihat post, tapi jika di sudah terdaftar  
dibisa mengomentari dan mengepost pertanyaan -  
pertanyaan”**

- Membuat auth pada laravel :
- Membuat CRUD FORM
- Membuat sistem komentar
- Membuat notifkasi

# Membuat auth pada laravel

- Php artisan make:auth
- Setelah itu jalankan perintah migrate, jika terjadi error :

- ```
Illuminate\Database\QueryException : SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 767 bytes (SQL: alter table `users` add unique `users_email_unique`(`email`))
```

- Maka tambahkan aturlah provider

*use Illuminate\Support\Facades\Schema;*

*Schema::defaultStringLength(191);*

# Membuat CRUD FORM

- Pertama buatlah tabelnya dengan migration :
- Karna kita menggunakan tabel relasi:

```
$table->unsignedBigInteger('user_id');
```

```
$table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
```

- Memuat model form.
- Mengatur model : 1 to Many :
- Model user :

*return \$this->belongsToMany('App\Models\Form');*

- Model Form

*return \$this->belongsTo('App\Models\User');*

- Mengatur middleware pada route :

```
“Route::group(['middleware' => 'auth'], function(){  
    //sintak  
});”
```

- Membuat slug

```
$slug = str_slug($request->title, '-');
```



- Membatasi huruf yang muncul

- Membuat sistem pencarian :

```
$cari_b = urldecode($request->cari);
```

```
if (!empty($cari_b)) {
```

```
Forms = Form::where('title','like','%'.$cari_b.'%')->get();
```

```
}
```

- Membuat sistem komentar
- //model comment

```
public function isOwner(){  
    if (Auth::guest()) {  
        return false;  
    }  
    return Auth::user()->id == $this->user->id;  
}
```

```
$form = form::findOrFail($id);  
    $comment = Comment::create([  
        'subject' => $request->subject,  
        'form_id' => $id,  
        'user_id' => Auth::user()->id  
    ]);
```

- Membuat owner pada post

- Sistem edit dan delete

# notifikasi

- Membuat sistem notifikasi

```
if ($form->user->id != Auth::user()->id) {  
    Notification::create([  
        'subject' => 'Ada komentar dari '. Auth::user()->name,  
        'user_id' => $buku->user->id,  
        'form_id' => $id  
    ]);  
}
```

- `==// forms`
- `public function comments()`
- `{`
- `return $this->hasMany('App\Models\Comment');`
- `}`
-

# notifikasi

- //user

```
public function comments(){  
    return $this->hasMany('App\Models\Comment');  
}
```

- 
- //coment
-



# notifikasi

//coment

```
public function user() { return $this->belongsTo('App\Models\User'); }
```

```
public function book() { return $this->belongsTo('App\Models\Book'); }
```

# **BERKENALAN SEDIKIT DENGAN API**