

LAPORAN

PEMROGRAMAN WEB LANJUT

Dosen Pengampu :

Lintang Setyo Kurniawati S.Kom, M.Pd



DISUSUN OLEH :

BAYU KRISNA (2010651095)

TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH JEMBER
2022

BAB 1

Pengertian Laravel

Laravel merupakan sebuah framework PHP yang dibangun menggunakan konsep MVC atau *Model View Controller* agar memudahkan programmer dalam membangun sebuah aplikasi.

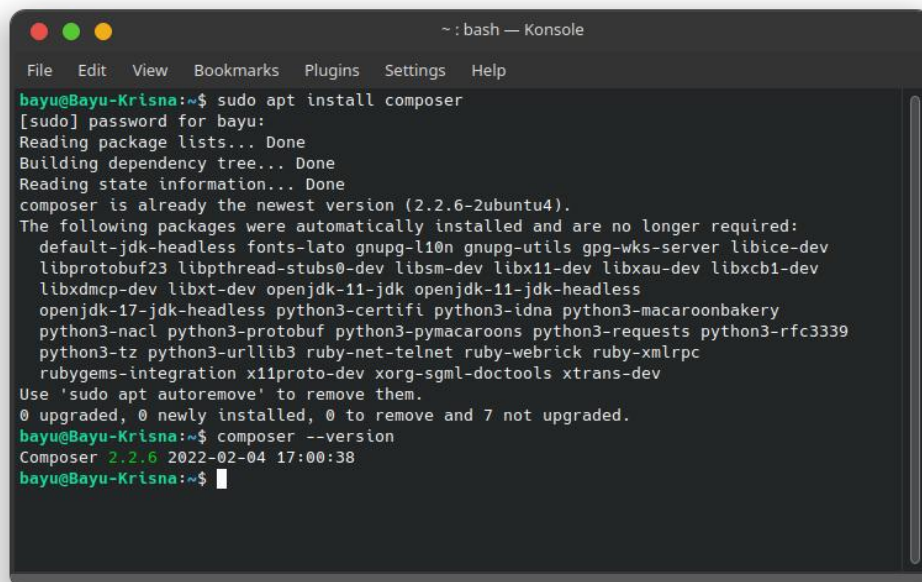
Cara Install Laravel

Untuk instalasi Laravel ada dua yaitu instalasi menggunakan composer dan instalasi lewat Laravel langsung tetapi kali ini cara yang saya gunakan adalah menggunakan composer untuk melakukan instalasinya.

Install Composer

Untuk menginstall laravel saya menggunakan composer. Composer merupakan sebuah dependency manager untuk PHP composer sendiri menyediakan library yang dibutuhkan pada aplikasi secara otomatis tanpa perlu mendownloadnya lagi. Disini saya menggunakan OS Linux dalam penginstalan composer. Adapun proses instalasi composer sebagai berikut :

1. Pastikan ada koneksi internet
2. Buka terminal / konsole
3. Ketik `sudo apt install composer` kemudian enter tunggu hingga proses instalasi selesai
4. Untuk cek composer sudah terinstall ketik `composer --version`
5. Jika berhasil akan muncul tulisan *Composer xxxxx*

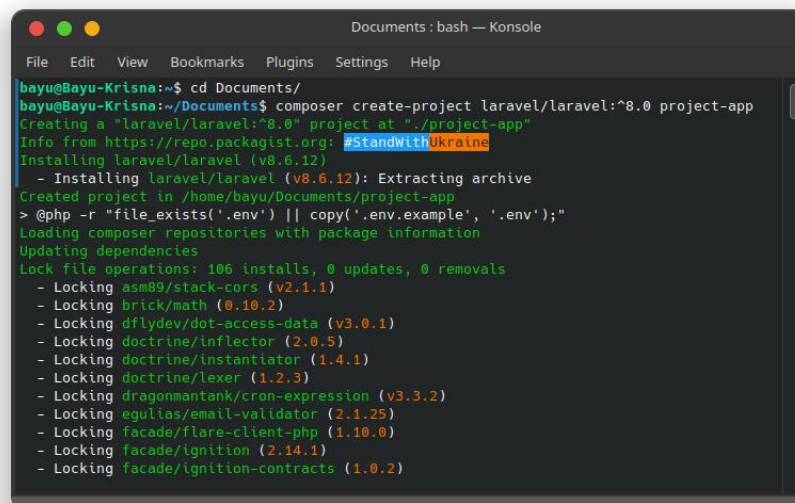


```
~ : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
bayu@Bayu-Krisna:~$ sudo apt install composer
[sudo] password for bayu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
composer is already the newest version (2.2.6-2ubuntu4).
The following packages were automatically installed and are no longer required:
 default-jdk-headless fonts-lato gnupg-l10n gnupg-utils gpg-wks-server libice-dev
 libprotobuf23 libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev
 libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless
 openjdk-17-jdk-headless python3-certifi python3-idna python3-macaronbakery
 python3-nacl python3-protobuf python3-pymacaroons python3-requests python3-rfc3339
 python3-tz python3-urllib3 ruby-net-telnet ruby-webrick ruby-xmlrpc
 rubygems-integration x11proto-dev xorg-sgml-doctools xtrans-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
bayu@Bayu-Krisna:~$ composer --version
Composer 2.2.6 2022-02-04 17:00:38
bayu@Bayu-Krisna:~$
```

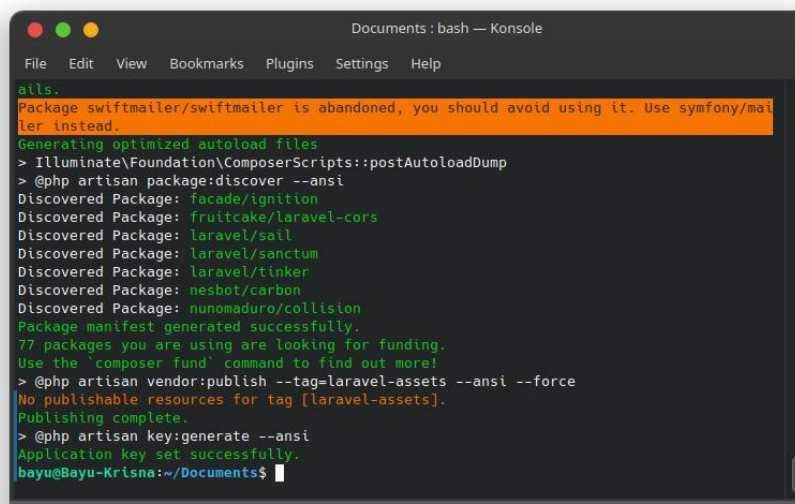
Instalasi framework laravel

Setelah composer terinstall composer, kemudian menginstall package laravel. Berikut cara instalasinya :

1. Buka terminal / konsole
2. Arahkan ke folder yang di inginkan
3. Kemudian ketik syntax ini *composer create-project laravel/laravel:^8.0 project-app* (yang bertulis tebal merupakan nama file laravel kita)
4. Tunggu hingga proses instalasi laravel selesai



```
Documents : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
bayu@Bayu-Krisna:~$ cd Documents/
bayu@Bayu-Krisna:~/Documents$ composer create-project laravel/laravel:^8.0 project-app
Creating a "laravel/laravel:^8.0" project at "./project-app"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v8.6.12)
- Installing laravel/laravel (v8.6.12): Extracting archive
Created project in /home/bayu/Documents/project-app
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 106 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.1.1)
- Locking brick/math (0.10.2)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.5)
- Locking doctrine/instantiator (1.4.1)
- Locking doctrine/lexer (1.2.3)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.10.0)
- Locking facade/ignition (2.14.1)
- Locking facade/ignition-contracts (1.0.2)
```



```
all.
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
77 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
No publishable resources for tag [laravel-assets].
Publishing complete.
> @php artisan key:generate --ansi
Application key set successfully.
bayu@Bayu-Krisna:~/Documents$
```

5. Kemudian masuk ke directory Laravel
6. Ketik syntax *php artisan serve*



7. Klik alamat ip yang muncul jika berhasil akan terbuka browser secara otomatis dan mengarah ke alamat server local laravel

BAB 2

Membuat Migration

Migration adalah sebuah fitur yang ada pada laravel. Dengan menggunakan migration laravel, memungkinkan kita untuk mengelola database dengan lebih mudah. Dengan menggunakan migration, kita bisa membuat table data dengan lebih mudah dan cepat. Pertama kita membuat database dengan phpmyadmin bernama **pwl_penjualan B** selanjutnya edit file dalam laravel yang bernama .env dan ubah syntax menjadi seperti berikut:

```
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=pwl_penjualan
15 DB_USERNAME=root
16 DB_PASSWORD=
```

Membuat Tabel Produk

Untuk membuat tabel kita menggunakan artisan dan tabel diberi nama `create_produk_table`. pertama buka console / terminal dan ketik syntax berikut *php artisan make:migration create_produk_table*

Mengubah script `create_produk_table`

Untuk membuat migration, kita menggunakan migration untuk menggunakannya, buka console lalu arahkan ke folder project Laravel yang sudah dibuat dan ketik syntax berikut *php artisan make:migration create_produk_table*. Dari perintah tersebut akan tercipta file migration baru dengan nama **2018_09_12_015121_create_produk_table** pada folder `project-app/database/migration`.

Selanjutnya buka file yang sudah dibuat menggunakan artisan lalu mengubah script menjadi seperti dibawah ini

```
public function up()
{
    Schema::create('produk', function (Blueprint $table) {
        $table->increments('id');
        $table->string('nama');
        $table->integer('id_kategori');
        $table->integer('qty');
        $table->integer('harga_beli');
        $table->integer('harga_jual');
        $table->timestamps();
    });
}
```

Konfigurasi database

Sebelum menjalankan migration perlu kita sedikit memodifikasi pada file project-app/config/database.php pada array strict dari **true** menjadi **false** seperti gambar berikut.

```
'mysql' => [
    'driver' => 'mysql',
    'url' => env('DATABASE_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => 'utf8mb4',
    'collation' => 'utf8mb4_unicode_ci',
    'prefix' => '',
    'prefix_indexes' => true,
    'strict' => false,
    'engine' => null,
    'options' => extension_loaded('pdo_mysql') ? array_filter([
        PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
    ]) : [],
],
```

Setelah selesai memodifikasi kita ketik perintah *ph artisan migrate* untuk membuat database tanpa perlu membuat dari phpMyAdmin.


```
project-app: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
bayu@Bayu-Krisna:~/Documents/Kuliah/Semester 5/Web Lanjut/project-app$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (11.11ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (9.81ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (9.47ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (14.53ms)
Migrating: 2022_10_14_093947_create_produk_table
Migrated: 2022_10_14_093947_create_produk_table (6.02ms)
bayu@Bayu-Krisna:~/Documents/Kuliah/Semester 5/Web Lanjut/project-app$
```

Perintah tersebut akan menjalankan method run(). Kemudian lihat di phpMyAdmin maka database pwl_penjualan akan terlihat seperti dibawah ini.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
<input type="checkbox"/> produk	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-

Rolling Back Migration

Rolling back migration merupakan perintah untuk mengembalikan satu operasi terakhir yang telah dilakukan sebelumnya. Perintah untuk melakukan rollback adalah *php artisan migrate:rollback*

```
project-app: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
bayu@Bayu-Krisna:~/Documents/Kuliah/Semester 5/Web Lanjut/project-app$ php artisan migrate:rollback
Rolling back: 2022_10_14_093947_create_produk_table
Rolled back: 2022_10_14_093947_create_produk_table (48.35ms)
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back: 2019_12_14_000001_create_personal_access_tokens_table (4.13ms)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back: 2019_08_19_000000_create_failed_jobs_table (3.44ms)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table (3.65ms)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table (3.36ms)
bayu@Bayu-Krisna:~/Documents/Kuliah/Semester 5/Web Lanjut/project-app$
```

Modifikasi tabel

Sebelumnya kita telah membuat tabel dengan migration menggunakan perintah `schema::create()`. Selain membuat tabel, kita juga dapat mengubah nama tabel menggunakan perintah `schema::rename()`. Untuk mencobanya, kita harus membuat migration dengan nama `ubah_tabel_produk` dengan script artisan sebagai berikut: `php artisan make:migration ubah_tabel_produk`. Selanjutnya ubah function `up` menjadi seperti dibawah ini

```
public function up()
{
    Schema::rename('produks', 'barang')
}
```

Kemudian ketikan perintah `php artisan migrate` sekarang table produk akan berubah menjadi barang

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> barang	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
6 tables	Sum	6	InnoDB	utf8mb4_general_ci	176.0 KiB	0 B

Membuat model

Untuk membuat model kita ketik perintah `php artisan make:model barang`. Hasil perintah tersebut tersimpan di folder `project-app/app/barang.php`. kemudian ubah script seperti dibawah ini

```
class barang extends Model
{
    protected $table = 'barang';
    use HasFactory;
}
```

Membuar seeder

Seeder merupakan data dummy yang kita buat ketika kita membuat sebuah aplikasi berbasis web, data hasil seeder dapat kita gunakan sebagai data awal untuk keperluan testing ataupun untuk keperluan lain. Untuk membuat seeder, gunakan perintah artisan pada konsolet sebagai berikut: `php artisan make:seeder produkTableSeeder` Perintah tersebut akan menghasilkan satu file baru pada folder

database/seeds dengan nama produkTableSeeder.php. selanjutnya ubah script pada function run() seperti dibawah ini

```
public function run()
{
    DB::table('produk')->insert(array(['nama'=>'Meja', 'id_kategori'=>'1', 'qty'=>'12',
    'harga_beli'=>'50000', 'harga_jual'=>'540000'], ['nama'=>'Kursi', 'id_kategori'=>'1',
    'qty'=>'12', 'harga_beli'=>'40000', 'harga_jual'=>'450000']));
}
```

Setelah itu jalankan perintah artisan *php artisan db:seed*. Setelah menjalankan cek database di phpmyadmin pada table barang akan ada isi yang telah kita buat di file produkTableSeeder.php

				id	nama	id_kategori	qty	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	Meja		12	50000	540000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	Kursi		12	40000	450000	NULL	NULL