**# Penjelasan**

Untuk bisa menjalankan web servernya laravel, cukup tulis nama foldernya di url browser.

untuk bisa melakukan hal tersebut, dibutuhkan **valet**, untuk valet windows bisa didapatkan di **pakagist.org**, search aja **valet-windows**, nanti ikutin aja prosedurnya. abis itu ketika valetnya udh diinstall, kita hanya perlu menuliskan valet park di cmd atau git bash kita (didalam folder yg ada folder isi laravelnya) agar laravelnya bisa dijalankan tanpa harus command **php artisan serve**.

**Penting!** = localhost:8080 -> menjalankan XAMPP

**kita bisa mengirimkan data lewat route**

Route::get('/about', function () {

return view('about', [

"name" => "Muhammad Yazid Akbar",

"age" => 17

]);

});

Jadi di route about ini menyimpan data array assoc, yg barisi name dan age. ini bisa kita panggil dengan cara

<h1><?= $name ;?></h1>

<p><?= $age;?></p>

nanti itu akan menampilkan data name yg ada di route about, begitupun dengan age

**# Database, migration, eloquent**

**php artisan migrate**

perintah ini akan menjalankan fungsi up yg ada di folder migrations>2014\_10\_12\_00000\_create\_users\_table

jadi nanti akan dibuatkan table secara otomatis lewat codingannya tanpa harus buat table manual di database mysql nya.

**php artisan migrate:rollback**

perintah ini akan merollback atau mengembalikan atau drop table yg sudah dibuat di function up yg ada di migration tadi (table migrationsnya tidak akan hilang, karena itu mirip dengan .git jadi dengan adanya migratios ini, kita bisa lihat perubahan apa yg terjadi)

**php artisan migrate:fresh**

perintah ini akan menghapus table dan membuat table yg baru ketika ada perubahan pada function up nya

ada yg namanya **APP\_ENV=local** di dalam file **.env**

jadi itu artinya environmentnya masih ada di local, jika di ganti jadi production, ketika kita melakukan command php artisan migrate:fresh, maka nanti akan muncul warning yg isinya **"Application in production!"** dan akan ada perintah confirm apakah command migrate itu akan dijalankan atau tidak.

**Eloquent ORM (object-relational mapper)**

sebuah fungsi untuk memetakan tiap2 data yg ada di dalam **table** atau **database** nanti ke dalam sebuah **object**.

ketika menggunakan **eloquent**, setiap table di dalam database itu terhubung ke dalam sebuah **"model"** untuk bisa kita interaksi dengan tablenya. selain kita mengambil record/data dari database kita, **eloquent model ini memungkinkan kita juga untuk melakukan CRUD lewat codingan**

**active record pattern** adalah pendekatan untuk mengakses data di dalam database. jadi table di dalam database kita itu dibungkus dalam sebuah class.jika sudah ada class yg mempresentasikan sebagai table, tiap data (row) itu terhubung ke dalam instance dari classnya (object). setelah membuat object/instance dari classnya, maka baris baru ditablenya akan ditambahkan ketika kita melakukan save. **Active record pattern ini biasanya digunakan dalam ORM**.

masuk ke terminal lalu ketik **php artisan tinker**, jika sudah masuk dan ingin membuat user baru, ketik **$user** yg berisi instance dari class users sebagai model kita.

**# Post Model**

**Route Model Binding (**

pada saat kita menyuntikkan sebuah ID dari model kita ke dalam Route kita atau ke dalam Controller, yg biasa kita lakukan adalah kita akan query record yg kita cari berdasarkan ID tadi.

Tugas Route Model Binding adalah nge-skip apa yg kita lakukan tadi, jadi si laravelnya akan langsung nge query-in supaya dapet langsung data sesuai dengan apa yang kita cari tanpa harus kirimin ID terus kita query sendiri, jadi secara otomatis mengirimkan instances dari model yg sesuai.

Contoh :

*// ini namanya Route Model Binding, jadi kita tulis nama modelnya disini*

*// variabel yg menerimanya harus sama dengan variabel yg dikirim di Route web.php nya, yg ini {post}*

*public* function show(Post **$post**)

    {

*// ini akan menampilkan single data dari kelas Post yg ada di folder model*

**return** view("post", [

            "title" **=>** "single post",

            "post" **=>** $post

        ]);

    }

**# Database Seeder (folder: database->seeder->DatabaseSeeder.php)**

kita bisa mengisi otomatis table kita ketika kita buat, jadi kita panen isi tablenya. Laravel memiliki kemampuan melakukan seed di dalam database kita dengan data testing caranya dengna menggunakan class seed.

Data yg masuk ke dalam database dengan menggunakan seeder ini adalah data dummy (bisa dipakai ketika development) untuk memudahkan ketika ingin membuat data dummy secara banyak

**php artisan migrate:fresh --seed**

jadi perintah diatas ini akan menghapus dan membuat kembali table yg ada di database sekalian mengirimkan seed/data yg telah dibuat di DatabaseSeeder

**# Factory & Faker**

untuk documentasi dari materi faker ini, bisa dilihat di website fakerphp.github.io dan bisa dilihat menunya, ada formatters, locales, dll

intinya adalah dengan menggunakan factory & faker, bisa mempercepat dalam pengisian data field dalam table. dan datanya adalah data dummy/fake, ini untuk proses development aja