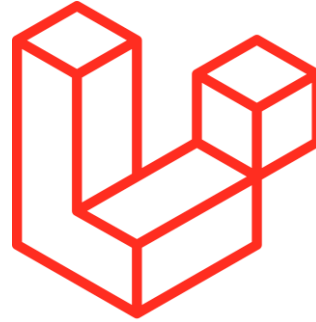


# Pemrograman Backend



## Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,  
Kec. Jombang, Kabupaten Jombang, Jawa Timur



# 13. Membangun Web Service REST API (Bagian 1)



# Web Service

- Web service adalah standar yang digunakan untuk melakukan pertukaran data antar aplikasi atau sistem, karena aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda. Contoh implementasi dari web service antara lain adalah SOAP dan REST.
- Web service menyediakan antarmuka layanan yang memungkinkan klien untuk berinteraksi dengan server dengan cara yang lebih umum daripada web browser lakukan.
- Klien mengakses operasi-operasi dalam antarmuka web service dengan cara meminta dan balasan diformat dalam **XML (eXtensible Markup Language)** atau **JSON (JavaScript Object Notation)**
- Biasanya ditransmisikan melalui HTTP.



# Manfaat Web Service

- Mendukung interoperabilitas di Internet global.
- Merupakan kunci penting dari integrasi bisnis-ke-bisnis.
- Memunculkan budaya 'mashup' yang memungkinkan pengembang pihak ketiga untuk mengembangkan perangkat lunak yang inovatif kreatif di atas layanan dasar yang sudah ada.
- Menyediakan middleware yang mendasari Grid computing dan Cloud computing.



# REST (Representational State Transfer)

- REST singkatan dari State Transfer Representasi, yang merupakan gaya arsitektur untuk aplikasi jaringan hypermedia, ini terutama digunakan untuk membangun Web service yang ringan, maintainable, dan scalable.
- Sebuah layanan berbasis REST disebut RESTful service atau REST API.
- REST tidak tergantung pada protokol, tapi hampir setiap RESTful service menggunakan HTTP sebagai protokol yang mendasarinya.
- REST merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.





# Arsitektur REST

- REST server menyediakan resources (sumber daya/data) dan REST klien mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya.
- Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID.
- Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.



# REST API

API atau Application Program Interface adalah sebuah antarmuka yang memungkinkan programmer untuk berkomunikasi dengan aplikasi. Cara kerja API serupa middleware, setiap kali ada request masuk, API akan mencegat request tersebut untuk diproses terlebih dahulu sebelum diteruskan ke Controller tujuan. Jika request valid, API akan merespon request dengan mengembalikan data yang diproses oleh Controller.

Penerapan API sangatlah luas, salah satunya adalah dengan arsitektur REST atau Representational State Transfer. Beberapa pedoman di dalam REST antara lain :

- **Client – Server** : klien (frontend) dan server (backend) dikembangkan secara terpisah, masing-masing dapat dibangun dengan teknologi yang berbeda-beda.
- **Stateless** : session state tidak disimpan di sisi server tapi di klien, tidak ada data klien yang perlu disimpan di server ketika request berlangsung.
- **Cacheable** : respon dari server dapat di-cache di sisi klien dalam kurun waktu tertentu agar klien tidak harus selalu melakukan request ke server untuk memperoleh data tertentu.





# Keuntungan REST API

Lebih sederhana/simpel untuk dikembangkan

Mudah dipelajari, tidak bergantung pada tools

Ringkas, tidak membutuhkan layer pertukaran pesan tambahan

Desain dan filosofi lebih dekat dengan web







# Kelemahan REST API

Mengasumsi model point-to-point komunikasi - tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara

Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri")

Berkaitan dengan model transport HTTP





# Metode HTTP dalam Arsitektur REST API

**GET**

- menyediakan hanya akses baca pada *resource*

**PUT**

- digunakan untuk menciptakan *resource* baru

**DELETE**

- digunakan untuk menghapus *resource*

**POST**

- digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru

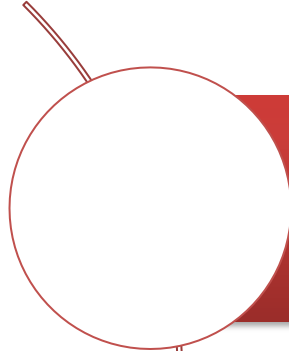
**OPTIONS**

- digunakan untuk mendapatkan operasi yang disupport pada *resource*

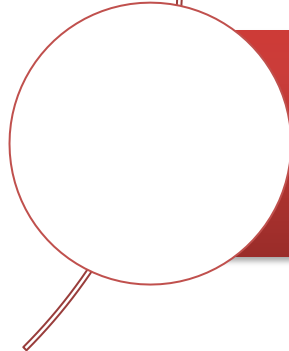




# Cara Kerja REST API



Sebuah client mengirimkan sebuah data atau request melalui **HTTP Request** dan kemudian server merespon melalui **HTTP Response**



*Web service* yang berbasis arsitektur REST kemudian dikenal sebagai **RESTful web services**. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST





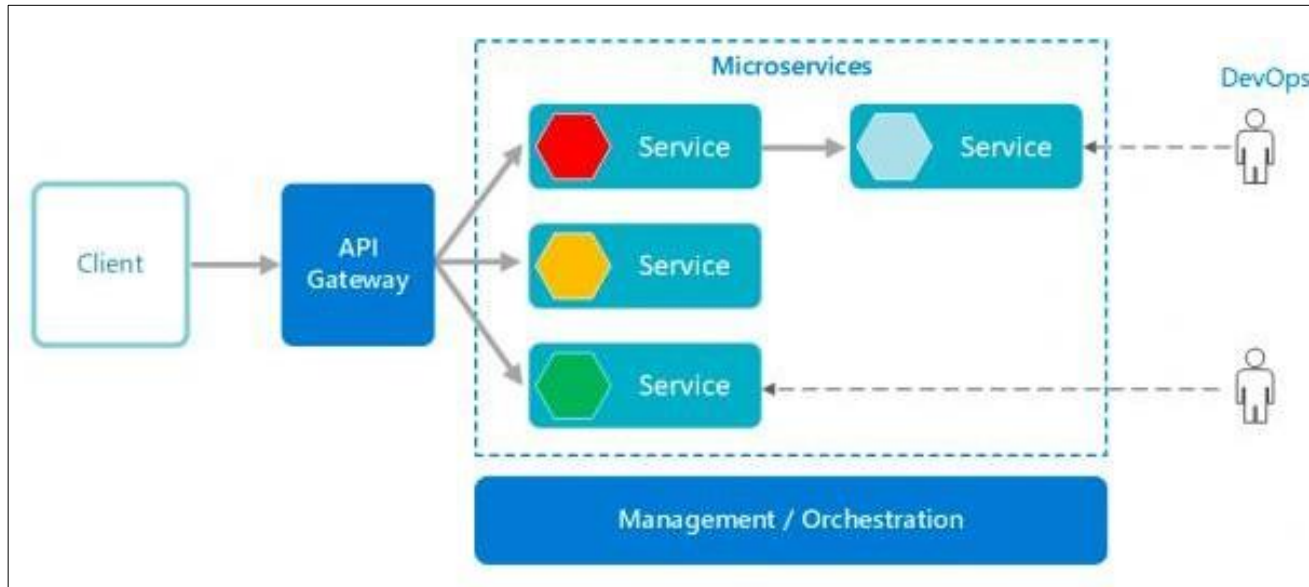
# Microservices

- Microservices berarti membagi aplikasi menjadi layanan yang lebih kecil dan saling terhubung.
- Microservices memungkinkan tiap fitur pada aplikasi mengalami pengembangan tersendiri.
- Pola arsitektur microservices secara signifikan memengaruhi hubungan antara aplikasi dan database.
- Di dalam microservices, masing-masing services atau layanan memiliki skema database tersendiri. Selain itu, services dapat menggunakan jenis database dan bahasa pemrograman yang paling sesuai dengan keperluan. Dengan begitu, tiap-tiap layanan akan lebih optimal.
- Microservices memungkinkan aplikasi menjadi lebih padat dan kompleks namun tetap ringan.
- Microservices adalah metode dengan membagi services ke bagian yang lebih kecil namun tetap berkaitan.



# Arsitektur Microservices

Setiap services pada microservices akan bertanggung jawab atas tugas yang berbeda-beda antar satu dan lainnya. Namun, setiap microservices dapat berkomunikasi dengan services lainnya untuk menyelesaikan kebutuhana bisnis tertentu.





# Lumen

- Lumen adalah micro framework dari Laravel yang difokuskan untuk membuat REST API.
- Pada Lumen banyak komponen – komponen Laravel yang dihilangkan, biasanya komponen – komponen Laravel yang berhubungan dengan View aplikasi.
- Lumen versi ramping Laravel yang tidak disertai fitur templating dan dukungan session state.
- Lumen memangkas library yang ada di laravel, sehingga kecepatan lumen lebih cepat dari pada Laravel.
- Lumen merupakan ekosistem Laravel yang khusus dirancang untuk layanan microservices.



# Lingkungan Kerja Lumen

- Sebelum menggunakan Lumen kita harus menyiapkan terlebih dahulu lingkungan development kita (environment).
- Setelah itu kita akan belajar membuat project lumen baru dan melakukan konfigurasi awal terhadapnya.
- Langkah pertama yang kita lakukan adalah menyiapkan lingkungan development kita.
- Hal ini terkait dengan persyaratan minimal sistem yang dibutuhkan oleh Lumen dan instalasi tool-tool pendukung sesuai dengan sistem operasi kita.



# Kebutuhan Software Pendukung Lumen

- Apache Web Server (XAMPP, Laragon)
- PHP versi 7.2 atau lebih
- Ekstensi PHP OpenSSL
- Ekstensi PHP PDO
- Ekstensi PHP Mbstring
- Database (MySQL, MariaDB, PostgreSQL dan lain-lain)
- Composer
- Web Browser
- Good Editor (Visual Code, Sublime Text, IDE Netbeans)







# Instalasi Lumen



Untuk instalasi Lumen, langkahnya buka terminal/CMD, lalu masuk ke folder Document Root web server Anda, misal di windows di E:\xampp\htdocs. Lalu ketikkan perintah seperti di bawah ini:

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\nasrul>e:

E:\>cd xampp\htdocs

E:\xampp\htdocs>composer create-project --prefer-dist laravel/lumen lumen
Creating a "laravel/lumen" project at "./lumen"
Installing laravel/lumen (v8.1.1)
- Downloading laravel/lumen (v8.1.1)
- Installing laravel/lumen (v8.1.1): Extracting archive
Created project in E:\xampp\htdocs\lumen
```





# Instalasi Lumen

Setelah instalasi Lumen selesai, langkah selanjutnya melalui terminal/CMD, pindah ke folder project Lumen di Document Root, untuk running server Lumen tidak bisa menggunakan perintah `php artisan serve`, tapi menggunakan built-in php dev servernya, seperti perintah di bawah ini:

```
E:\xampp\htdocs>cd lumen  
  
E:\xampp\htdocs\lumen>php -S localhost:8000 -t public  
[Sun Jun 20 19:56:42 2021] PHP 7.4.2 Development Server (http://localhost:8000) started
```

Setelah itu buka web browser, pada url ketikkan: `http://localhost:8000`





# Konfigurasi Dasar Lumen



Setelah server Lumen dijalankan, runningnya terasa sangat ringan. Jika Anda menggunakan kekuatan yang ada di laravel (seperti eloquent, facades, middleware, dan lain-lain) untuk dipakai pada Lumen, Anda harus mengaktifkannya di file **bootstrap/app.php**. Uncomment baris kode dibawah ini:

```
//aktifkan(uncomment) kode di bawah yang sebelumnya dicomment
$app->withFacades();
$app->withEloquent();
$app->routeMiddleware([
    'auth' => App\Http\Middleware\Authenticate::class,
]);
$app->register(App\Providers\AppServiceProvider::class);
$app->register(App\Providers\AuthServiceProvider::class);
$app->register(App\Providers\EventServiceProvider::class);
```



# Composer: Lumen Vs Laravel

- Ketika Anda mengetik php artisan di terminal, pada Laravel akan tampil banyak opsi.
- Sebaliknya di Lumen Anda hanya akan menemukan sedikit. karena Lumen ini lebih banyak menggunakan teknik manual daripada otomatisasi seperti di Laravel.
- Jika di laravel Anda bisa membuat model dan controller melalui artisan, di Lumen Anda tidak bisa melakukan hal tersebut. Jadi, di Lumen kita harus membuat filenya secara manual.



## Route: Lumen Vs Laravel

Di Laravel, kita bisa mengetikkan route seperti ini:

```
Route::get('post', 'PostController@index');
```

Atau menggunakan resource:

```
Route::resource('post', 'PostController');
```

Tetapi di Lumen berbeda, seperti dibawah ini:

```
$router->get('post', 'PostController@index');
```

Dan satu lagi, di Lumen Anda tidak bisa menggunakan fitur resource.





# Migration pada Lumen

Setelah berhasil menginstall dan menjalankan aplikasi Lumen, langkah selanjutnya adalah membuat sebuah database baru untuk memigrasi database. Silahkan buat database baru, misalnya dengan nama: **dblumen**. Setelah database berhasil dibuat, langkah selanjutnya adalah mengkonfigurasi koneksi database yang ada di aplikasi Lumen kita. Silahkan buka file **.env** dan cari kode dibawah ini :

```
DB_CONNECTION=mysql //koneksi databasenya ke database mysql
DB_HOST=localhost //hostnya berada di lokal
DB_PORT=3306 //nomor port database mysql
DB_DATABASE=dblumen //nama basis data
DB_USERNAME=root //user basis data
DB_PASSWORD=rahasia //password user basis data
```



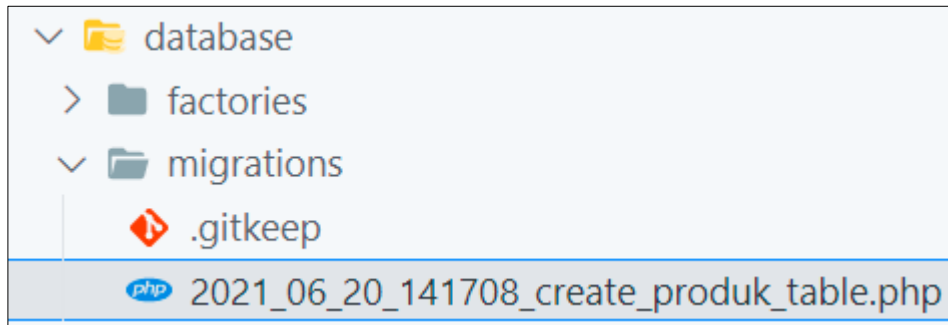


# Membuat File Migration

Pertama kita perlu membuat sebuah file Migration, file ini nantinya bisa dieksekusi agar dapat mengubah database sesuai perintah yang kita tuliskan di file tersebut. Untuk membuat file Migration kita gunakan perintah di bawah ini:

```
E:\xampp\htdocs\lumen>php artisan make:migration create_produk_table  
Created Migration: 2021_06_20_141708_create_produk_table
```

Setelah itu akan muncul file migration yang kita buat:





# Isi File Migration



```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateProdukTable extends Migration
{
    public function up()
    {
        Schema::create('produk', function (Blueprint $table) {
            $table->id();
            $table->string('namaProduk');
            $table->longText('deskripsiProduk');
            $table->integer('hargaProduk');
            $table->enum('kategoriProduk',
                ['makanan', 'minuman', 'perlengkapan mandi', 'kosmetik']);
            $table->timestamps();
        });
    }
}
```







# Eksekusi Migration

Setelah membuat kolom-kolom pada tabel produk, Langkah selanjutnya adalah mengeksekusi migration. Buka terminal dan masuk ke root path aplikasi Lumen kita, jalankan perintah di bawah ini:

```
E:\xampp\htdocs\lumen>php artisan migrate  
Migration table created successfully.  
Migrating: 2021_06_20_141708_create_produk_table  
Migrated: 2021_06_20_141708_create_produk_table (262.00ms)
```

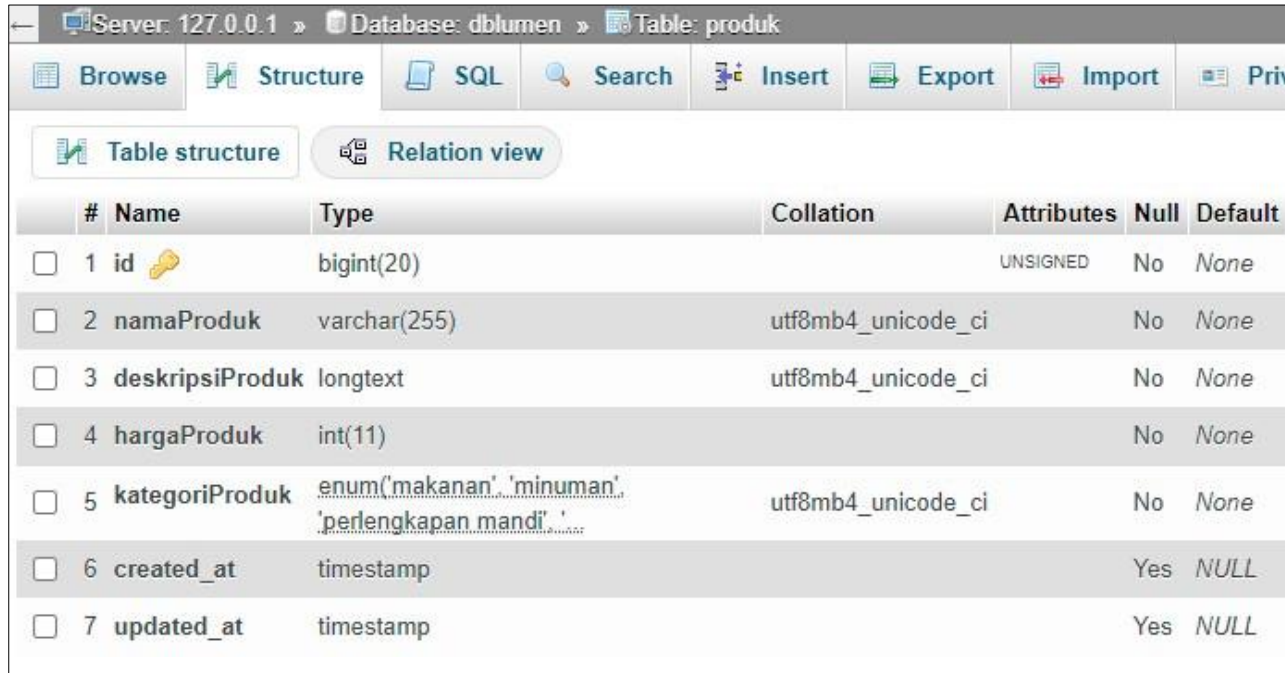
Dengan perintah di atas, maka file-file Migration yang berisi perintah pengubahan database akan dijalankan, hanya kode di method up() yang dieksekusi. Untuk melihat perubahannya, kita bisa melihat langsung tabel di database kita. Coba jalankan perintah migrate di atas setelah kita berhasil menggenerate file migration.






# Hasil Eksekusi Migration

Setelah berhasil menjalankan perintah migration, coba buka PHPMyadmin untuk mengecek apakah benar tabel sudah berhasil dibuat.



Server: 127.0.0.1 » Database: dbumen » Table: produk

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/> 1	id 	bigint(20)		UNSIGNED	No	None
<input type="checkbox"/> 2	namaProduk	varchar(255)	utf8mb4_unicode_ci		No	None
<input type="checkbox"/> 3	deskripsiProduk	longtext	utf8mb4_unicode_ci		No	None
<input type="checkbox"/> 4	hargaProduk	int(11)			No	None
<input type="checkbox"/> 5	kategoriProduk	enum('makanan', 'minuman', 'perlengkapan mandi', '...')	utf8mb4_unicode_ci		No	None
<input type="checkbox"/> 6	created_at	timestamp			Yes	NULL
<input type="checkbox"/> 7	updated_at	timestamp			Yes	NULL

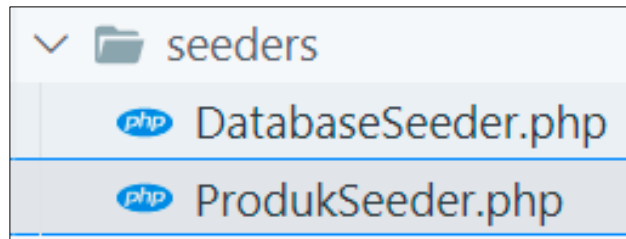


# Membuat Sedding

Untuk membuat sedding dari tabel pegawai yang sudah ada, caranya adalah buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

```
E:\xampp\htdocs\lumen>php artisan make:seeder ProdukSeeder  
Seeder created successfully.
```

Setelah perintah di atas berhasil dieksekusi, lihatlah di project Anda di folder seeders ada file ProdukSeeder.php, seperti tampak gambar di bawah ini:





# Mengisi Data Sedding

Untuk mengisi/mengenerate data di tabel produk, caranya adalah buka file ProdukSeeder.php yang berada di folder seeders, lalu isi data seperti kode program di bawah ini:

```
database > seeders >  ProdukSeeder.php >  ProdukSeeder >  run
1  <?php
2  namespace Database\Seeders; use Illuminate\Database\Seeder;
3  use DB; //tambahan
4  class ProdukSeeder extends Seeder{
5      public function run()
6      {
7          DB::table('produk')->insert(
8              [
9                  [
10                     'namaProduk'=>'Bakwan', 'deskripsiProduk'=>'Bakwan Udang',
11                     'hargaProduk'=>2000, 'kategoriProduk'=>'makanan',
12                     'created_at' => new \DateTime, 'updated_at' => null,
13                 ],
14                 [
15                     'namaProduk'=>'Martabak', 'deskripsiProduk'=>'Martabak Telor',
16                     'hargaProduk'=>25000, 'kategoriProduk'=>'makanan',
17                     'created_at' => new \DateTime, 'updated_at' => null,
18                 ],
19             ]
20         );
21     }
}
```



# Eksekusi Data Seeding

Setelah mengisi data produk sesuai dengan kolom-kolomnya, langkah selanjutnya adalah mengeksekusi data seedingnya. Buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

```
E:\xampp\htdocs\lumen>php artisan db:seed --class=ProdukSeeder  
Database seeding completed successfully.
```

Dengan perintah di atas, maka tabel pegawai akan terisi data produk yang baru. Anda bisa cek di phpMyAdmin.

id	namaProduk	deskripsiProduk	hargaProduk	kategoriProduk	created_at	updated_at
1	Bakwan	Bakwan Udang	2000	makanan	2021-06-20 22:33:50	NULL
2	Martabak	Martabak Telor	25000	makanan	2021-06-20 22:33:50	NULL



**TERIMA KASIH  
ATAS SEGALA PERHATIAN  
SEMOGA BERMANFAAT...**

