

## Merancang Arsitektur dalam Web API 1

### 1. API

API atau *Application Programming Interface* adalah sebuah software interface yang digunakan untuk melakukan komunikasi antar program dalam 2 komputer atau lebih. API mengizinkan komunikasi antar dengan berbagai bahasa pemrograman yang berbeda. Sehingga pengembang aplikasi tidak perlu menyediakan semua data sendiri karena cukup mengambil data yang dibutuhkan dari platform lain dengan API.

API juga bisa disebut sebagai kontrak antar aplikasi dengan dokumentasi sebagai kesepakatan antar aplikasi. Jika antar aplikasi mengirimkan request dengan spesifikasi dokumen yang sesuai dengan dokumentasi, maka perangkat lain yang menerima request akan memberikan respon balasan.

API terdiri dari dua komponen, yaitu :

- Spesifikasi teknis yang menggambarkan pilihan pertukaran data antara solusi dengan spesifikasi dilakukan dalam bentuk permintaan pemrosesan dan pengiriman data.
- Interface pada aplikasi ditulis dengan spesifikasi yang mewakili isinya.

Contoh sederhananya, ketika Anda membuat aplikasi pemesanan tiket pesawat atau kamar hotel. Sebagai developer, Anda hanya perlu menggunakan API maskapai penerbangan, hotel dan sistem pembayaran.

Aplikasi lain mengintegrasikan fungsi yang diminta oleh aplikasi sebelumnya dan interface yang digunakan kedua aplikasi ini untuk berkomunikasi yaitu menggunakan API. API akan memberikan data yang Anda minta seperti ketersediaan tiket dan jumlah harga kemudian mengembalikan data tersebut kepada Anda. Developer tidak perlu lagi repot membuat program baru dengan coding yang rumit dari awal, API sangat membantu untuk mengambil data dan informasi. Developer dapat menambahkan fungsi lain seperti rekomendasi penerbangan lain atau hotel lain dan pemrosesan pembayaran dari aplikasi lain ke solusi yang tersedia atau membuat aplikasi baru menggunakan layanan pihak ketiga.

Tidak hanya developer, bagi user atau pengguna juga tidak perlu keluar dari halaman aplikasi atau website yang sedang mereka kunjungi untuk mendapatkan data. Hanya dalam

satu aplikasi, user dapat mendapatkan apa yang mereka cari dan mendapatkan pengalaman yang menyenangkan. Setiap API berisi data dan diimplementasikan oleh function calls, yaitu pernyataan yang meminta aplikasi untuk melakukan tindakan dan layanan tertentu. Function calls dijelaskan dalam dokumentasi API.

### Cara Kerja API

Untuk mengetahui cara kerja sebuah API bisa dilihat gambar berikut:



1. Aplikasi mengakses API

Bagian pertama adalah ketika pengguna mengakses alamat API atau biasa disebut endpoint sesuai dengan fungsi/ fitur. Saat mengakses endpoint API, aplikasi juga akan mengirimkan parameter/ data.

2. API melakukan request ke server

Setelah berhasil mengakses endpoint, permintaan tersebut akan diteruskan ke server sesuai bersamaan dengan data yang dikirimkan.

3. Server memberikan response ke API

Ketika menemukan data atau proses di server selesai, maka server akan menghubungi kembali API. Data yang dikembalikan umumnya berupa status code, message dan data.

4. API menyampaikan response ke aplikasi

Selanjutnya API akan meneruskan informasi dari server ke aplikasi.

## Komponen RESTful API

RESTful API memiliki 4 komponen penting yaitu:

### a. URL Design

RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut endpoint dalam pemanggilannya.

### b. HTTP Verbs

Setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di request client:

#### - GET

GET adalah metode HTTP Request yang paling mudah, metode ini digunakan untuk membaca atau mendapatkan data dari sumber.

#### - POST

POST adalah metode HTTP Request yang digunakan untuk membuat data baru dengan menyisipkan data dalam body saat request dilakukan.

#### - PUT

PUT adalah metode HTTP Request yang biasanya digunakan untuk melakukan update data resource.

#### - DELETE

DELETE adalah metode HTTP Request yang digunakan untuk menghapus suatu data pada resource.

### c. HTTP Response Code

HTTP Response Code adalah kode standarisasi dalam menginformasikan hasil request kepada client. Secara umum terdapat 3 kelompok yang biasa kita jumpai pada RESTful API yaitu :

2XX : adalah response code yang menampilkan bahwa request berhasil.



0812-3431-1135



www.arkatama.id



info.arkatama@gmail.com



Perum. Joyoagung Greenland No. B4-B5,  
Tlogomas, Lowokwaru, Malang, Jawa Timur

4XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi client.

5XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi server.

**d. Format Response**

Setiap request yang dilakukan client akan menerima data response dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data response tersebut barulah client bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.

**5.1. Hal yang Perlu Diperhatikan Saat Membuat API**

Untuk membuat REST yang baik ada beberapa hal yang perlu diperhatikan:

**a. Endpoint gunakanlah Kata Benda Jamak (Plural)**

/pets (mengembalikan list pet)  
/pets/1 (mengembalikan pet dengan Id )  
dan jangan menggunakan kata kerja  
/getAllPets  
/getAllPetsByCategory/1

**b. Gunakan HTTP Method yang benar**

Method-method standar yang biasa digunakan yaitu :

GET: Untuk mendapatkan data

POST: Untuk menambahkan data baru

PUT: Untuk memanipulasi Data

DELETE: Untuk menghapus Data

GET hanya untuk mengambil data, jangan gunakan untuk memanipulasi data

**c. Gunakan sub-resource untuk sebuah relasi**

Kadang kita mempunyai kebutuhan sebagai berikut, akan menampilkan semua data foods dari 1 pet, nah untuk kebutuhan sebagai berikut baiknya standarnya sebagai berikut

/pets/1/foods (mengeluarkan data foods dari data pets ID 1)

/pets/1/foods/2 (mengeluarkan data foods dengan ID 2)

**d. Menggunakan Parameter**

Untuk kebutuhan filtering bisa menggunakan parameter dengan standar berikut

/pets?category='cat' (mengeluarkan data pets dengan kategori 'cat')

**e. Menggunakan Paging**

Khususnya dalam menggunakan method GET, untuk jumlah data yang besar kita bisa mengeluarkan data berdasarkan limit data (Paging)

/pets?limit=20 (mengeluarkan 20 data pets)

**f. Pengurutan Data**

Disamping paging, kadang kita juga perlu untuk mengurutkan data berdasarkan request dari client

/pets?sort=updated\_at (Mengeluarkan data berdasarkan dari yang update terbaru )

**g. HTTP Code**

Berikut adalah List kode yang biasanya digunakan dalam REST

- 200: Untuk menunjukkan operasi yang di-request berhasil di proses
- 201 CREATED: Menunjukan operasi yang dilakukan berhasil membuat resource baru

- 400 BAD REQUEST: Biasanya kode error yang menunjukkan dari sisi client misalkan client salah menginput data
- 403 FORBIDDEN: Response dari client misalnya client mengakses endpoint yang menyertakan Authorization tapi tidak mengirim token
- 401 NOT FOUND: Reponse yang menunjukkan sumber yang direquest tidak tersedia
- 500 INTERNAL SERVER ERROR: terjadi kesalahan / error dari sisi server

#### h. **Gunakan Versioning**

Penggunaan versi API sangatlah penting, apalagi di perusahaan yang banyak menggunakan versi API.

/v1/products

/v2/products

#### i. **API Response**

Tidak ada standart khusus mengenai API response namun umumnya sebagai berikut

- Success Response

```
1 {  
2   "status": "success",  
3   "data": [],  
4   "message": "Sukses mengambil data hewan peliharaan"  
5 }  
6
```

- **Error Response**

```
1 {  
2   "status": "error",  
3   "data": null,  
4   "message": "Gagal mengambil data hewan peliharaan",  
5   "errorCode": "500"  
6 }  
7
```

## **Fungsi API**

API dapat menyederhanakan dan mempercepat pengembangan aplikasi. Bagi developer, API memberikan beberapa keuntungan diantaranya :

1. Pengembangan aplikasi menjadi lebih cepat dan efektif dengan proses integrasi antar dua aplikasi
2. Beban kerja server menjadi lebih ringan karena tidak perlu menyimpan data.
3. Developer bisa mengembangkan aplikasi yang kompleks tetapi lebih fungsional.

## **Jenis-Jenis API**

Berdasarkan hak akses yang diberikan, API terdiri dari empat jenis. Adapun jenis API tersebut meliputi :

1. **Public API**

Public API sering disebut Open API, API jenis public bisa digunakan secara bebas antar platform tanpa autentifikasi. Contoh API yang bersifat public yang bisa digunakan bisa dilihat pada link <https://github.com/farizdotid/DAFTAR-API-LOKAL-INDONESIA>

2. **Private API**

Private API adalah jenis API yang memerlukan autentifikasi untuk bisa diimplementasikan antar aplikasi. Private API

## Arsitektur API

Secara umum terdapat tiga arsitektur yang dapat digunakan developer untuk membangun API. Arsitektur API yang dimaksud adalah bentuk data yang digunakan untuk berkomunikasi. Arsitektur tersebut adalah:

a. RPC

Remote Procedure Calls (RPC) RPC adalah suatu protokol yang menyediakan suatu mekanisme komunikasi antar proses yang memungkinkan suatu program untuk berjalan pada suatu komputer tanpa terasa adanya eksekusi kode pada sistem yang jauh ( remote system ). RPC memiliki dua jenis XML-RPC dan JSON-RPC.

b. SOAP

Arsitektur API lainnya adalah SOAP (Simple Object Access Protocol). Arsitektur ini menggunakan XML (Extensible Markup Language) yang memungkinkan semua data disimpan dalam dokumen.

c. REST

REST adalah singkatan dari REpresentational State Transfer. REST merupakan gaya arsitektur untuk menyediakan standar antara sistem komputer di web, sehingga memudahkan sistem untuk berkomunikasi satu sama lain. Format data yang digunakan REST adalah JSON.

## Instalasi library

Dalam studi kasus ini, kita akan membuat Rest API menggunakan framework codeigniter dan library Rest yang disediakan oleh oleh Chris Kacerguis<sup>1</sup>. CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC

---

<sup>1</sup> <https://github.com/chriskacerguis>



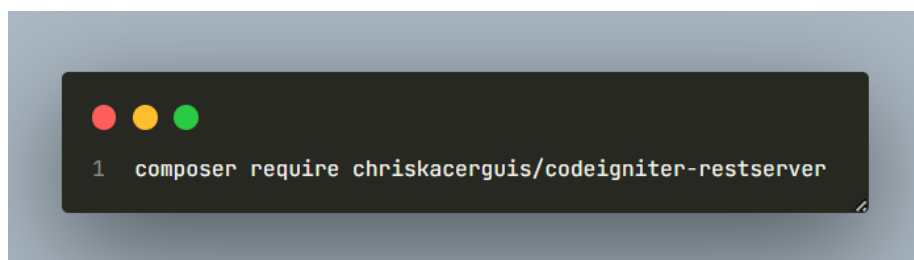
(Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. Berikut ini langkah-langkah untuk melakukan library Rest dengan codeigniter.

**a. Download file codeigniter**

Download file codeigniter pada situs resmi yang telah disediakan<sup>2</sup>, disarankan untuk menggunakan versi 3.x.xx. Kemudian ekstrak file codeigniter yang telah di download tadi untuk menghasilkan sebuah folder CodeIgniter.

**b. Install library REST API codeigniter**

Untuk librari REST API codeigniter kita bisa menggunakan library open source yang disediakan oleh Chris Kacerguis pada link <https://github.com/chriskacerguis>. Selain mendownload langsung dari repositori yang sudah tersedia, kita bisa mendownload library tersebut dengan composer. Berikut ini command yang bisa kita gunakan untuk download composer :



```
1 composer require chriskacerguis/codeigniter-restserver
```

Setelah file library didownload, masuk ke folder vendor/chriskacerguis/codeigniter-restserver/application lalu copy paste beberapa file yang dibutuhkan dalam folder kita. File tersebut yakni :

- **config** (folder) : **\*Idap.php, \*rest.php**
- **helper** (folder) : **\*db\_helper.php**
- **language** (folder) : **\*english (folder)**
- **libraries** (folder) : **\*Format.php, \*REST\_Controller.php**
- Pindahkan file yang di copy tadi ke dalam folder yang kita buat di localhost.

---

<sup>2</sup> <https://codeigniter.com/download>

### 1.3 Request

API Request merupakan perintah untuk mengirimkan permintaan data ke server dengan URL yang disediakan. API request terdiri dari endpoint URL dan HTTP method. HTTP Method mengindikasikan aksi yang kita inginkan kepada API yang tersedia. HTTP method yang sering digunakan dalam request API terdiri dari :

- GET : untuk menerima data dari API
- POST : untuk mengirimkan data baru ke API
- PATCH atau PUT untuk mengupdate data
- DELETE untuk menghapus data

#### 1.3.1 HTTP Verb

Sebagaimana dijelaskan pada sub sebelumnya, HTTP Verb akan mengidentifikasi perintah apa yang akan kita kirim ke API URL yang tersedia. Berikut ini uraian masing-masing HTTP verb dalam API

- Get

Metode GET menyediakan akses baca pada sumber daya yang disediakan oleh REST API. Sebagai contohnya digunakan untuk membaca data dari tabel telepon pada database kontak. Untuk membaca data dari database dapat dilakukan dengan active record yang telah disediakan Codeigniter. Sebelum membaca data dari database, fungsi GET yang akan dibuat terlebih dahulu memeriksa apakah terdapat property id pada address bar sehingga data yang ditampilkan dapat di seleksi berdasarkan id atau ditampilkan semua.

- POST

Metode POST digunakan untuk mengirimkan data baru dari client ke server REST API. Sebagai contohnya digunakan untuk menambahkan kontak baru yang terdiri dari id, nama, dan nomor.

- DELETE

Metode DELETE digunakan untuk menghapus data yang telah ada di server REST API. Sebagai contohnya digunakan untuk menghapus data dengan id 88 pada tabel telepon database kontak

- PATCH/PUT

Metode PUT digunakan untuk memperbarui data yang telah ada di server REST API. Sebagai contohnya digunakan untuk memperbarui data dengan id 88 pada tabel telepon database kontak.

### 1.3.2 Request Header

Request header terdiri dari informasi yang bisa membantu kita dalam mengirimkan request kepada API URL. Dalam API, request header merupakan bagian penting dari request API karena informasi yang terdapat dalam header merupakan meta-data yang berhubungan dengan request API dan response. Informasi penting yang terdapat didalam header meliputi :

- Authorization

Membawa informasi autentikasi yang diperlukan dari client untuk melakukan request ke server seperti token key, jenis autentikasi

- WWW-Authenticate

Merupakan informasi yang dikirimkan server kepada client yang berisi form autentikasi sebelum client bisa menerima response dari request yang dikirimkan

- Accept-Charset

Merupakan pengkodean jenis karakter data yang diizinkan server seperti UTF-8

- Content-Type

Berisi informasi mengenai jenis media yang digunakan untuk menyusun response yang dikirimkan server kepada client. Contoh content-type yang diizinkan dalam API adalah text/HTML, text/JSON dan XML.

- Cache-Control

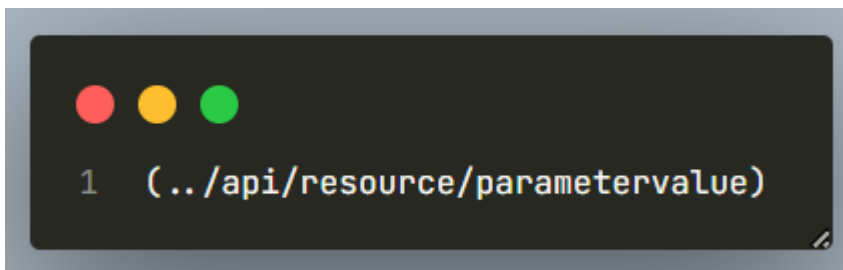
Berisi kebijakan cache yang ditentukan oleh server untuk mengirimkan response kepada client.

### 1.3.3 Request Params

Request params dalam API merupakan cara untuk menambahkan parameter pada URL endpoint dari api yang akan kita akses. Ada 2 cara pemberian parameter pada URL endpoint API.

- Menyematkan sebagai bagian dari URL endpoint

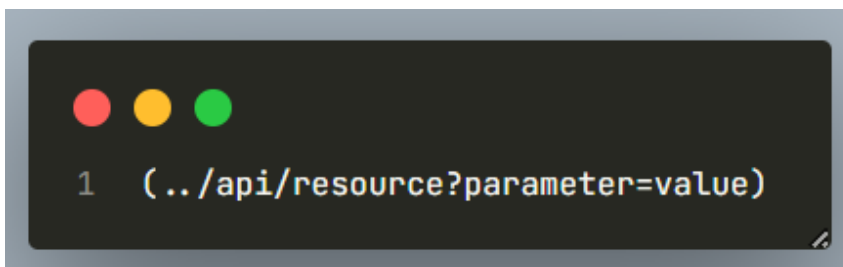
Jika kita menyematkan parameter sebagai bagian dari URL API yang akan diakses, maka parameter yang disematkan bersifat wajib diisi. Apabila parameter tidak diisi maka API tidak akan bisa diakses karena ada bagian yang hilang.



```
1 (../api/resource/parametervalue)
```

- Menyematkan sebagai query argumen

Jika kita menyematkan parameter sebagai query argumen, jika wajib untuk mendefinisikan key yang berisi nilai dari parameter yang dikirim. Key akan diidentifikasi sebagai keyword yang akan diproses di backend API.



```
1 (../api/resource?parameter=value)
```

### 1.3.4 Request Body

Request body merupakan data yang dikirim dari client melalui API. Request body diisi ketika request verb menggunakan method selain GET atau HEAD. Pada umumnya request body digunakan ketika request verb berupa POST, PUT atau PATCH. Adapun format yang digunakan untuk mengirim data pada request body adalah sebagai berikut :

1. form data, merupakan format pengiriman data dari body yang memungkinkan client untuk mengirimkan data teks maupun file.
2. raw data, merupakan format data dari body yang hanya berisi data teks.