



Web Service

Make everything happen,
but rarely in the spotlight



Web Service

Requirements

Bahasa Pemrograman

PHP

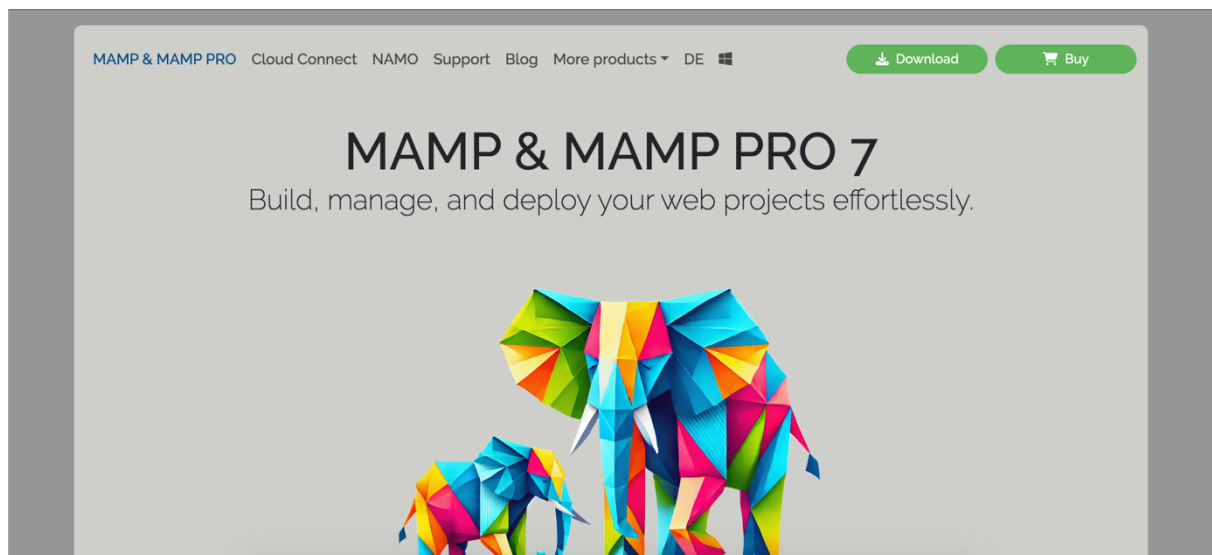
Web Server

MAMP, Laragon, XAMPP, dll

Apabila menggunakan MAMP, silakan download pada website resminya :

<https://www.mamp.info/>

Apabila menggunakan **universal development environment** seperti MAMP, Laragon, atau XAMPP terdapat folder **htdocs** tempat kita membuat dan menjalankan project PHP.



Database

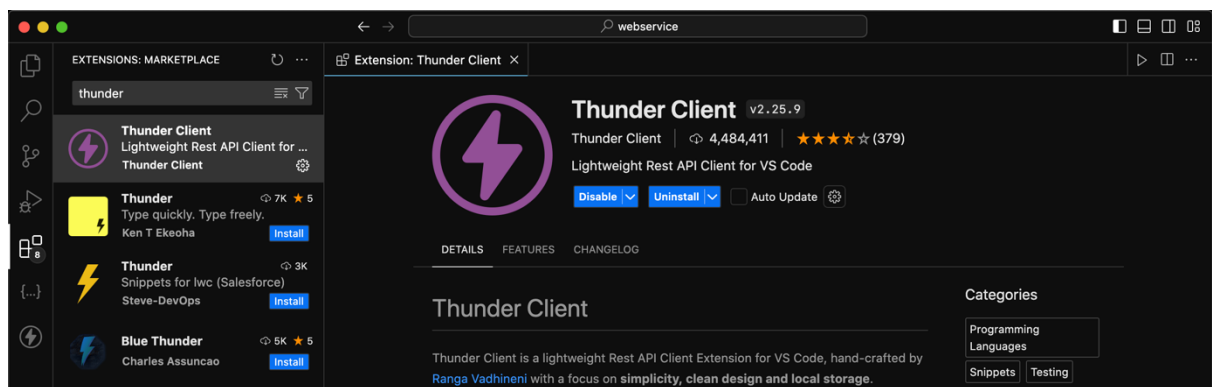
MySQL, PostgreSQL, dll

Jika menggunakan MAMP, Laragon, atau XAMPP sudah include database MySQL.

Rest API Client

Thunder Client, Postman, dll

Cara instal Thunder Client cukup dengan melakukan searching pada ekstensi VS Code kemudian ketik **"Thunder Client"** dan instal.





Web Service

Web Service dengan PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang sangat populer untuk pengembangan web, dan dapat digunakan untuk membuat berbagai jenis web service. Web service memungkinkan aplikasi untuk berkomunikasi satu sama lain melalui jaringan, biasanya menggunakan protokol HTTP.

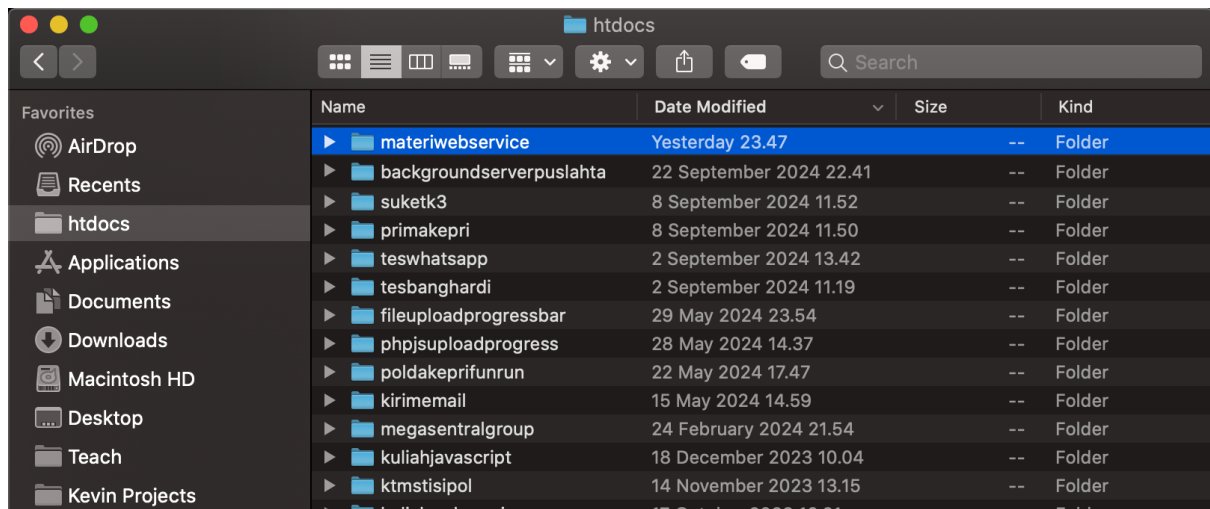
Tipe Web Service

Ada dua tipe utama web service :

- SOAP (Simple Object Access Protocol): Protocol yang lebih formal dan menggunakan XML sebagai format data.
- REST (Representational State Transfer): Lebih sederhana dan sering menggunakan JSON sebagai format data.

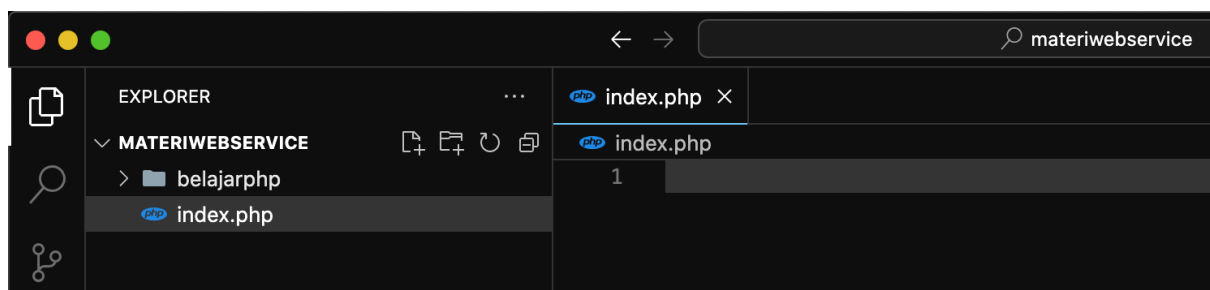
Lokasi Directory Project

Buat sebuah folder dengan nama **materiwebservice** di dalam **htdocs**.



Struktur Folder

Berikut adalah contoh sederhana tentang cara membuat web service RESTful menggunakan PHP. Silakan open folder **materiwebservice** pada VS Code.



Sebelum praktik lebih jauh tentang Web Service sebaiknya dimulai dengan pengetahuan bahasa pemrograman PHP.



Web Service

Tag

Ada beberapa tag utama pada PHP yang harus diingat yaitu :

1. Penulisan Awal

Bahasa php dimulai dengan tag **<?php** dan diakhiri **?>**

```
tagawal.php ×
belajarphp > tagawal.php
1 <?php
2
3 ?>
```

2. Komentar

Komentar pada program merupakan tulisan pada program yang tidak dieksekusi. Beberapa cara penulisan komentar sebagai berikut.

```
komentar.php ×
belajarphp > komentar.php
1 <?php
2 /*
3 KOMENTAR
4 1 BLOK
5 */
6
7 // KOMENTAR SATU BARIS
8 ?>
9
```



Web Service

Variabel

Dalam PHP, penulisan variabel diawali dengan karakter '\$', kemudian diikuti dengan huruf sebagai karakter pertama. Setelah itu, dapat dilanjutkan dengan kombinasi huruf dan angka. Variabel tidak boleh mengandung spasi maupun tanda baca di dalamnya, kecuali underscore ('_'). Variabel pada PHP bersifat **"case sensitive"**, yang berarti Anda harus memperhatikan penulisan huruf besar dan huruf kecil. Variabel **\$jumlah** berbeda dengan **\$Jumlah**.

```
variabel.php x
belajarpHP > variabel.php
1 <?php
2 $nama = 'Kevin';
3 $kota = 'Tanjungpinang';
4 ?>
```

Contoh penamaan variabel yang benar :

```
variabelbenar.php x
belajarpHP > variabelbenar.php
1 <?php
2 $nama_pemakai = "Kevin";
3 $kota_3 = "Tanjungpinang";
4 $user1 = "kevin";
5 ?>
```

Contoh penamaan variabel yang salah :

```
variabelsalah.php 1 x
belajarpHP > variabelsalah.php
1 <?php
2 $nama pemakai = "";
3 $3kota = "";
4 $us\er1 = "";
5 ?>
```



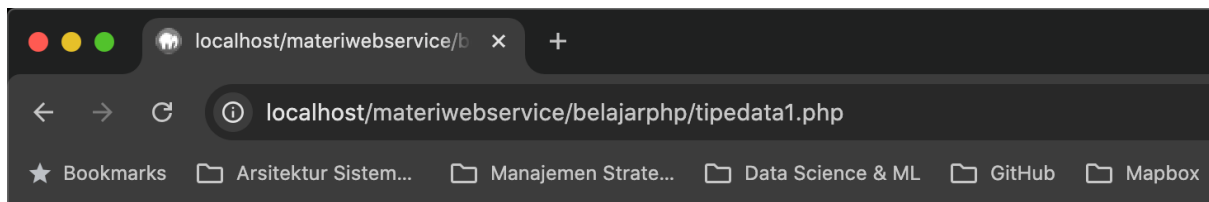
Web Service

Tipe Data dari Variabel

Tipe data variabel tidak perlu dideklarasikan, PHP akan otomatis mengkonversi atau menentukan tipe data variabel berdasarkan nilai yang disimpannya.

```
tipedata1.php x
belajarphp > tipedata1.php
1  <?php
2  $nama = 'Kevin Perdana';
3  $nilai = 90;
4  $phi = 3.14;
5  echo gettype($nama); // CEK TIPE DATA
6  echo "<br>"; // BARIS BARU
7  echo gettype($nilai); // CEK TIPE DATA
8  echo "<br>"; // BARIS BARU
9  echo gettype($phi); // CEK TIPE DATA
10 ?>
```

Cek output dengan cara memanggil url/lokasi file PHP pada browser.



string
integer
double



Operator

Operator Aritmatika

Semua operator ini berfungsi untuk memanipulasi nilai numerik pada variabel, memungkinkan PHP untuk melakukan berbagai jenis perhitungan.

1. Penjumlahan

php

```
$a = 10;  
$b = 5;  
$hasil = $a + $b; // $hasil akan bernilai 15
```

2. Pengurangan

php

```
$a = 10;  
$b = 5;  
$hasil = $a - $b; // $hasil akan bernilai 5
```

3. Perkalian

php

```
$a = 10;  
$b = 5;  
$hasil = $a * $b; // $hasil akan bernilai 50
```



4. Pembagian

php

```
$a = 10;  
$b = 5;  
$hasil = $a / $b; // $hasil akan bernilai 2
```

5. Modulus

php

```
$a = 10;  
$b = 3;  
$hasil = $a % $b; // $hasil akan bernilai 1 (10 : 3 sisa 1)
```

6. Pangkat

php

```
$a = 2;  
$b = 3;  
$hasil = $a ** $b; // $hasil akan bernilai 8 (2 pangkat 3)
```



Operator Penugasan

Operator penugasan dalam PHP digunakan untuk menetapkan nilai ke variabel. Operator ini sering digunakan dalam kombinasi dengan operator aritmatika untuk melakukan perhitungan sekaligus menetapkan hasilnya ke variabel yang sama.

1. Operator Penugasan Dasar

php

```
$a = 10; // Variabel $a akan memiliki nilai 10
```

2. Penugasan dengan Penjumlahan

php

```
$a = 10;  
$a += 5; // Sama dengan $a = $a + 5, jadi $a akan bernilai 15
```

3. Penugasan dengan Pengurangan

php

```
$a = 10;  
$a -= 5; // Sama dengan $a = $a - 5, jadi $a akan bernilai 5
```

4. Penugasan dengan Perkalian

php

```
$a = 10;  
$a *= 2; // Sama dengan $a = $a * 2, jadi $a akan bernilai 20
```



5. Penugasan dengan Pembagian

php

```
$a = 10;  
$a /= 2; // Sama dengan $a = $a / 2, jadi $a akan bernilai 5
```

Operator Perbandingan

Operator perbandingan dalam PHP digunakan untuk membandingkan dua nilai. Hasil dari perbandingan ini selalu berupa nilai boolean: true atau false.

1. Sama Dengan

php

```
$a = 5;  
$b = "5";  
var_dump($a == $b); // Output: bool(true)
```

2. Identik

php

```
$a = 5;  
$b = "5";  
var_dump($a === $b); // Output: bool(false) (karena tipe data berbeda)
```

3. Tidak Sama Dengan

php

```
$a = 5;  
$b = "10";  
var_dump($a != $b); // Output: bool(true)
```



4. Lebih Besar Dari

```
php

$a = 10;
$b = 5;
var_dump($a > $b); // Output: bool(true)
```

5. Lebih Kecil Dari

```
php

$a = 5;
$b = 10;
var_dump($a < $b); // Output: bool(true)
```

6. Lebih Besar Sama Dengan

```
php

$a = 10;
$b = 10;
var_dump($a >= $b); // Output: bool(true)
```

7. Lebih Kecil Sama Dengan

```
php

$a = 5;
$b = 10;
var_dump($a <= $b); // Output: bool(true)
```



Operator Logika

Operator logika dalam PHP digunakan untuk melakukan operasi logika pada nilai boolean. Hasil dari operasi ini adalah nilai boolean true atau false. Operator logika umumnya digunakan dalam struktur kendali seperti if, while, dan for untuk memeriksa beberapa kondisi secara bersamaan.

1. AND (&& atau and)

Mengembalikan true jika kedua operand bernilai true.

php

```
$a = true;
$b = false;
var_dump($a && $b); // Output: bool(false) (salah satu bernilai fals
var_dump($a and $b); // Output: bool(false)
```

2. OR (|| atau or)

Mengembalikan true jika salah satu operand bernilai true. Jika kedua operand bernilai false, maka hasilnya adalah false.

php

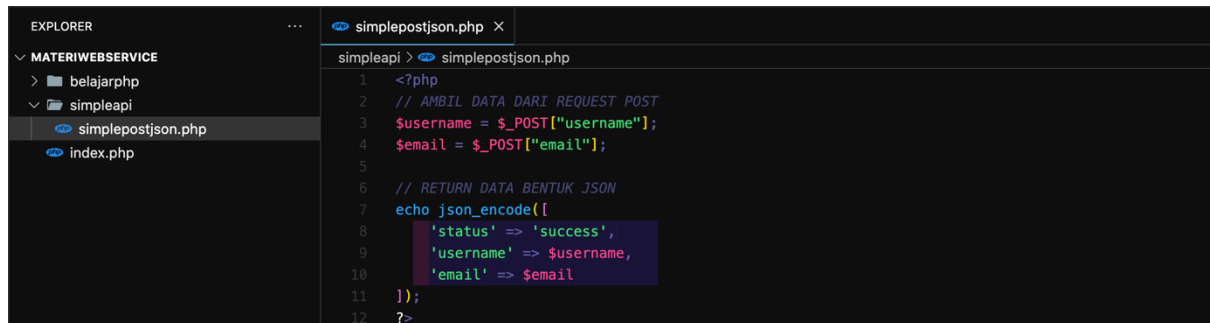
```
$a = true;
$b = false;
var_dump($a || $b); // Output: bool(true) (salah satu bernilai true)
var_dump($a or $b); // Output: bool(true)
```



Web Service

Simple Post to JSON Format

Berikut contoh pengiriman data dari REST API Client dan diterima oleh *endpoint* PHP dengan nama file **simplepostjson.php**



```
1 <?php
2 // AMBIL DATA DARI REQUEST POST
3 $username = $_POST['username'];
4 $email = $_POST['email'];
5
6 // RETURN DATA BENTUK JSON
7 echo json_encode([
8     'status' => 'success',
9     'username' => $username,
10    'email' => $email
11 ]);
12 ?>
```

Buka **Thunder Client**, ubah method menjadi **POST**, ketik url *endpoint* sebagai berikut : **http://localhost/materiwebbservice/simpleapi/simplepostjson.php**
Isi bagian **Body** > **Form** seperti screenshot dibawah. Kemudian klik *Send*.

