**PHP: Bekerja Dengan Tanggal Dan Waktu (Prosedural)**

**Tanggal dan Waktu Pada PHP**

Terdapat banyak cara dan pendekatan untuk memanipulasi tanggal dan waktu pada PHP. Yang paling inti adalah pendekatan prosedural dan pendekatan OOP.

Pendekatan prosedural umumnya menggunakan fungsi bawaan php yaitu fungsi date(), sedangkan pendekatan objek biasanya menggunakan kelas bawaan PHP bernama keals DateTime.

Pada tutorial ini kita akan membahas penggunakan pendekatan prosedural untuk memanipulasi tanggal dan waktu pada PHP, dan pada pertemuan selanjutnya insyaallah kita akan membahasnya dengan pendekatan pemrograman berbasis objek.

**Pendekatan OOP vs Prosedural**

Jika anda bingung apa itu *Object Oriented Programming* (OOP) dan apa bedanya dengan pendekatan prosedural, jangan khawatir. Insyaallah kita akan mempelajarinya dalam seri tutorial php berikut-berikutnya.

Intinya, pendekatan prosedural adalah penggunaan [fungsi](https://jagongoding.com/web/php/dasar/fungsi/) tanpa ada pembuatan suatu instan dari kelas tertentu.

Sedangkan penggunakan pendekatan objek atau OOP, anda perlu membuat instan dari suatu kelas tertentu dengan perintah new.

Contoh prosedural:

**<?**php

**echo** date**(**'d-m-Y'**);** *// 17-02-2020*

Contoh pendekatan objek:

**<?**php

$tanggalHariIni **=** **new** DateTime**();**

**echo** $tanggalHariIni**->**format**(**'d-m-Y'**);** *// 17-02-2020*

Untuk lebih detilnya, anda anda bisa melihat pembahasan tentang [memanipulasi tanggal dan waktu pada PHP dengan pendekatan objek](https://jagongoding.com/web/php/menengah/bekerja-dengan-tanggal-dan-waktu-oop/).

**Penggunaan fungsi date()**

Secara sintaks, fungsi date() menerima 2 paremeter. Satu parameter bersifat wajib, dan yang satu lagi bersifat opsional.

**<?**php

$hariIni **=** date**(**format**[,** timestamp**]);**

Penjelasan parameter:

* **format**: adalah format tanggal dan waktu yang akan kita tampilkan. Parameter ini bertipe data string.
* **timestamp**: timestamp bersifat opsional, jika tidak didefinisikan, PHP akan menggunakan waktu sekarang ketika fungsi date dipanggil.

**Contoh menampilkan tahun sekarang:**

**<?**php

**echo** date**(**'Y'**)** **.** '<br>'**;**

**echo** date**(**'y'**)** **.** '<br>'**;** *# 2 digit terakhir dari tahun sekarang*

Output:

2020

20

**Contoh menampilkan hari dan bulan:**

**<?**php

**echo** date**(**'d/m/Y'**)** **.** '<br>'**;**

**echo** date**(**'d-m-Y'**)** **.** '<br>'**;**

**echo** date**(**'d, m Y'**)** **.** '<br>'**;**

*# anda juga bisa menggunakan escape string*

**echo** date**(**'d \wk\wk m \wk\wk Y'**)** **.** '<br>'**;**

Output:

**17/02/2020**

**17-02-2020**

**17,** **02** **2020**

**17** wkwk **02** wkwk **2020**

**Fungsi date() dengan timestamp**

Seperti yang sudah disebutkan di atas, fungsi date() menerima 2 parameter dan parameter yang kedua adalah timestamp.

**Apa itu timestamp?**

Timestamp adalah jarak waktu dari tanggal 1 Januari 1970 pukul 00:00 ke tanggal sekarang dan waktu sekarang. Jarak tersebut didefinisikan dalam satuan detik.

Untuk mengetahui timestamp dari waktu sekarang, anda bisa menggunakan fungsi time().

**<?**php

**echo** time**();**

Output:

**1581946962**

Karena timestamp adalah jarak waktu dengan satuan detik, maka angka di atas akan terus berubah seiring dengan waktu ketika anda memanggil fungsi time().

**Menampilkan timestamp dengan fungsi date()**

Kita bisa menampilkan timestamp ke dalam bentuk string tanggal dan waktu dengan menjadikannya parameter ke-2 pada fungsi date().

Contoh, misalkan kita punya timestamp 1234567890. Lalu kita ingin menampilkannya dalam format d-m-Y, kita bisa melakukannya dengan kode program berikut:

**<?**php

$timestamp **=** **1234567890;**

$stringDate **=** date**(**'d-m-Y H:i'**,** $timestamp**);**

**echo** "Timestamp: {$timestamp} <br>"**;**

**echo** "String date: {$stringDate} <br>"**;**

Output:

Timestamp: 1234567890

String date: 14-02-2009 06:31

**Konversi String date() ke Dalam Bentuk Timestamp**

Kita telah mengetahui cara menampilkan timestamp ke dalam string tanggal dan waktu.

Sekarang pertanyaannya, kita memiliki string tanggal atau waktu, lalu kita ingin mendapatkan nilai timestamp-nya, bagaimana caranya?

Kita bisa menggunakan fungsi bawaan php yang bernama strtotime().

Fungsi ini menerima parameter string dalam berbagai bentuk. Perhatikan contoh berikut:

**<?**php

**echo** strtotime**(**"now"**)** **.** '<br>'**;** *# timestime untuk waktu sekarang*

**echo** strtotime**(**"2020-12-31"**)** **.** '<br>'**;** *# timestamp untuk tanggal 31 Desember 2020*

**echo** strtotime**(**"2020-12-31 12:00"**)** **.** '<br>'**;** *# timestamp untuk tanggal 31 Desember 2020 pukul 12:00*

Output:

1581948269

1609347600

1609390800

Ia juga bisa menerima deskripsi tanggal dan waktu dalam bahasa inggris, bahkan ia juga bisa menerima parameter dinamis terhadap waktu sekarang. Perhatikan contoh berikut:

**<?**php

**echo** strtotime**(**"05th maret 2004"**)** **.** '<br>'**;** *# untuk tanggal 5 Maret 2004*

**echo** strtotime**(**"next sunday"**)** **.** '<br>'**;** *# hari minggu yang akan datang*

**echo** strtotime**(**"previous friday"**)** **.** '<br>'**;** *# hari jumat yang lalu*

**echo** strtotime**(**"+2 days"**)** **.** '<br>'**;** *# 2 hari lagi*

Output:

1582390800

1581613200

1582121258

**Format Tanggal dan Waktu**

Kita telah mencoba beberapa kode format tanggal, kode format tanggal didefinisikan dengan satu karakter string. Berikut ini adalah kode format tanggal dan waktu yang sering digunakan pada PHP.

**Kode format hari**

Beberapa kode format hari yang sering digunakan di dalam bahasa pemrograman PHP.

| **Kode** | **Keterangan** | **Nilai Kembalian** |
| --- | --- | --- |
| d | Urutan hari dalam bulan, diikuti angka 0 untuk hari < 10 | 01-31 |
| j | Urutan hari dalam bulan, tidak diikuti angka 0 | 1-31 |
| D | Nama hari, hanya 3 karakter. Nama hari sesuai dengan pilihan lokalisasi | Mon - Sun |
| l | Nama hari secara penuh. Nama disesuaikan dengan pilihan lokalisasi | Sunday - Saturday |

**Kode format bulan**

Beberapa kode format bulan yang sering digunakan di dalam bahasa pemrograman PHP.

| **Kode** | **Keterangan** | **Nilai Kembalian** |
| --- | --- | --- |
| m | Urutan bulan dalam tahun, diikuti angka 0 untuk hari < 10 | 01-12 |
| n | Urutan bulan dalam tahun, tidak diikuti angka 0 | 1-12 |
| M | Nama pendek dari suatu bulan, hanya 3 karakter. Nama disesuaikan dengan pilihan lokalisasi | Jan-Dec |
| F | Nama lengkap dari suatu bulan. Nama disesuaikan dengan pilihan lokalisasi | January - December |

**Kode format tahun**

Beberapa kode format tahun yang sering digunakan di dalam bahasa pemrograman PHP.

| **Kode** | **Keterangan** | **Nilai Kembalian** |
| --- | --- | --- |
| Y | Menampilkan 4 digit tahun | Contoh: 1990 - 2030 |
| y | Menampilkan hanya 2 digit terakhir dari tahun | Contoh: 90 - 30 |

**Kode format waktu**

Beberapa kode format waktu yang sering digunakan di dalam bahasa pemrograman PHP.

| **Kode** | **Keterangan** | **Nilai Kembalian** |
| --- | --- | --- |
| h | Jam dalam format 12-hour | 01-12 |
| H | Jam dalam format 24-hour | 01-24 |
| a | Lowercase Ante meridiem and Post meridiem | am-pm |
| A | Uppercase Ante meridiem and Post meridiem | AM-PM |
| i | Menit dengan diikuti angka 0 | 01-59 |
| s | Detik dengan diikuti angka 0 | 01-59 |

Untuk daftar format tanggal dan waktu yang lebih lengkap, anda bisa merujuk secara langsung ke [dokumentasi resmi php](https://www.php.net/manual/en/function.date.php).

**Pengaturan Zona Waktu atau Time Zone**

Zona waktu PHP bisa berbeda dengan zona waktu pengaturan server kita. Misalkan server kita menggunakan zona waktu "America/Regina", PHP bisa berjalan dengan zona waktu yang lain semisal "Asia/Jakarta".

Dan zona waktu ini adalah zona waktu yang digunakan disemua fungsi yang berkaitan dengan tanggal dan waktu pada PHP pada kode program anda.

Untuk mengganti pengaturan zona waktu, anda bisa mengubahnya pada file php.ini.

Atau anda juga bisa menggunakan fungsi date\_default\_timezone\_set().

Fungsi tersebut menerima satu parameter yaitu nama zona waktu yang kita pilih. Perhatikan contoh berikut:

**<?**php

date\_default\_timezone\_set**(**"America/Regina"**)** **.** '<br>'**;**

**echo** date**(**"d-m-Y H:i"**)** **.** '<br>'**;**

*// ganti ke indonesia*

date\_default\_timezone\_set**(**"Asia/Jakarta"**)** **.** '<br>'**;**

**echo** date**(**"d-m-Y H:i"**);**

Dalam kode program di atas, kita memanggil fungsi date() dua kali untuk menampilkan tanggal dan waktu sekarang. Akan tetapi, kedua fungsi date() di atas berjalan dengan zona waktu yang berbeda karena kita telah menggantinya dengan fungsi date\_default\_timezone\_set().

Output dari kode program di atas adalah:

18-02-2020 03:50

18-02-2020 16:50

**Daftar zona waktu yang didukung di PHP**

Untuk melihat daftar zona waktu yang didukung pada PHP, anda bisa pergi ke dokumen resmi [php.net](https://php.net/) pada halaman [Zona Waktu](https://www.php.net/manual/en/timezones.php).

Atau anda bisa mendapatkan list array dari zona waktu tersebut dari fungsi static DateTimeZone::listIdentifiers() seperti contoh berikut:

**<?**php

$daftarTimezone **=** DateTimeZone**::**listIdentifiers**();**

**foreach** **(**$daftarTimezone **as** $timezone**)** **{**

**echo** "{$timezone} <br>"**;**

**}**

Output:

Graphical user interface, text, application, email

Description automatically generated

**Memanipulasi Tanggal dan Waktu**

Memanipulasi tanggal dan waktu adalah pekerjaan yang sangat penting dalam proses pembuatan aplikasi atau web. Misal kita punya toko online, lalu kita ingin mengirim peringatan tagihan yang harus dibayar setelah 3 hari dari proses pembelian, maka di sini kita butuh untuk memanipulasi tanggal dan waktu sehingga kita bisa tahu 3 hari setelah tanggal pembelian itu tanggal berapa dan jam berapa?

Untuk melakukan pekerjaan ini, cara yang paling pas adalah menggunakan pendekatan objek dengan kelas bawaan PHP bernama DateTime, atau mungkin bisa dengan menggunakan pustaka pihak ketiga seperti [Carbon](https://carbon.nesbot.com/).

Untuk lebih detilnya, insyaallah tentang manipulasi tanggal dan waktu ini akan kita bahas pada pertemuan selanjutnya.

**Membandingkan Tanggal dan Waktu**

Kita bisa membandingkan tanggal dan waktu dengan pendekatan prosedural menggunakan 2 cara:

* membandingkan string hasil kembalian dari fungsi date(). Cara ini tidak terlalu direkomendasikan.
* mengubah tanggal menjadi timestamp lalu mulai membandingkan keduanya. Cara ini adalah cara yang direkomendasikan.

Perhatikan contoh berikut:

**<?**php

$tanggal1 **=** "2020-01-01"**;**

$tanggal2 **=** "2020-01-31"**;**

**if** **(**$tanggal1 **<** $tanggal2**)** **{**

**echo** "{$tanggal1} < {$tanggal2}"**;**

**}** **elseif** **(**$tanggal1 **===** $tanggal2**)** **{**

**echo** "{$tanggal1} = {$tanggal2}"**;**

**}** **else** **{**

**echo** "{$tanggal1} > {$tanggal2}"**;**

**}**

Output:

2020-01-01 < 2020-01-31

Contoh perbandingan tanggal dengan menggunakan timestamp:

**<?**php

$tanggal1 **=** "2020-02-01"**;**

$tanggal2 **=** "2020-01-31"**;**

**if** **(**strtotime**(**$tanggal1**)** **<** strtotime**(**$tanggal2**))** **{**

**echo** "{$tanggal1} < {$tanggal2}"**;**

**}** **elseif** **(**$tanggal1 **===** $tanggal2**)** **{**

**echo** "{$tanggal1} = {$tanggal2}"**;**

**}** **else** **{**

**echo** "{$tanggal1} > {$tanggal2}"**;**

**}**

Output:

2020-02-01 > 2020-01-31

**Pembahasan Selanjutnya**

Pada tutorial ini kita telah mempelajari bagaimana cara kita bekerja dengan tanggal dan waktu pada PHP dengan pendekatan prosedural. Kita telah mencoba beberapa fungsi seperti fungsi date(), time(), strtotime() dan juga fungsi date\_default\_timezone\_set() untuk mengatur zona waktu default.

Pada pertemuan selanjutnya, insyaallah kita akan melanjutkan pembahasan yang sama, hanya saja menggunakan pendekatan yang berbeda! Yaitu pendekatan objek.