

# REFERENSI CEPAT PHP

Sintaks, array, OOP, database, dan file I/O esensial

```
Dasar  
Hello World  
<?php  
echo "Hello, World!\n";  
// Kode PHP harus berada di dalam tag <?php ... ?>
```

```
Jalankan PHP  
php script.php # jalankan file  
php -r 'echo "hi\n";' # jalankan kode inline  
php -S localhost:8000 # server dev bawaan
```

```
Komentar  
// komentar satu baris  
# juga satu baris  
/* *multi-baris  
komentar */
```

```
Variabel & Tipe  
Variabel  
$name = "PHP"; // string  
$version = 8.3; // float  
$count = 42; // int  
$active = true; // bool  
$tens = null; // null
```

```
Pemeriksaan Tipe  
gettype($x) Kembalikan tipe sebagai string  
is_string($x) Periksa apakah string  
is_int($x) Periksa apakah integer  
is_array($x) Periksa apakah array  
is_null($x) Periksa apakah null  
isset($x) Periksa apakah sudah dilet dan bukan null  
empty($x) Periksa apakah kosong (falsy)
```

```
Casting Tipe  
$n = (int) "42"; // 42  
$s = (string) "3.14"; // "3.14"  
$b = (bool) ""; // false  
$a = (array) $obj; // object ke array
```

```
Konstanta  
define("MAX_SIZE", 100);  
const API_VERSION = "v2";  
echo MAX_SIZE; // 100
```

```
String  
Dasar String  
$name = "World";  
echo "Hello, $name!"; // interpolasi variabel  
echo "Hello, $name!"; // literal (tanpa interpolasi)  
echo "Value: {${key}}"; // ekspresi kompleks
```

```
Fungsi String  
strlen($s) Panjang string dalam byte  
mb_strlen($s) Panjang string dalam karakter (aman multibyte)  
strtolower($s) Konversi ke huruf kecil  
strtoupper($s) Konversi ke huruf besar  
trim($s) Hapus whitespace dari kedua ujung  
str_replace(a, b, $s) Ganti 'a' dengan 'b' dalam '$s'  
substr($s, 0, 5) Substring dari posisi 0, panjang 5  
strpos($s, 'find') Temukan posisi substring (false jika tidak ada)  
explode(' ', $s) Pecah string menjadi array  
implode(' ', $a) Gabungkan array menjadi string
```

```
Heredoc & Nowdoc  
$html = <<<HTML  
<p>Hello, $name</p>  
HTML;  
$raw = <<<'TEXT'  
No $interpolation here  
TEXT;
```

```
Array  
Terindeks & Asosiatif  
$nums = [1, 2, 3]; // terindeks  
$user = ["name" => "Alice", "age" => 30]; // asosiatif  
$nums[] = 4; // tambahkan ke akhir  
echo $user["name"]; // akses
```

```
Fungsi Array  
count($a) Jumlah elemen  
array_push($a, $v) Tambahkan ke akhir  
array_pop($a) Hapus dan kembalikan elemen terakhir  
array_merge($a, $b) Gabungkan dua array  
in_array($v, $a) Periksa apakah nilai ada  
array_key_exists($k, $a) Periksa apakah key ada  
array_map($fn, $a) Terapkan fungsi ke setiap elemen  
array_filter($a, $fn) Filter elemen berdasarkan callback  
sort($a) Urutkan di tempat (reindeks)  
array_keys($a) Kembalikan semua key  
Iterasi  
foreach ($users as $user) { echo $user; }  
foreach ($map as $key => $value) {  
    echo "$key: $value\n";  
}
```

```
Fungsi  
Fungsi Dasar  
function add(int $a, int $b): int {  
    return $a + $b;  
}  
echo add(3, 5);
```

```
Argumen Default & Named
```

```
function greet(string $name, string $greeting = "Hello"): string {  
    return "$greeting, $name!";  
}  
greet("Alice");  
greet($greeting: "Hi", name: "Bob"); // named args (PHP 8+)
```

```
Array Function  
$double = fn(int $x): int => $x * 2;  
$nums = array_map(fn($n) => $n * 10, [1, 2, 3]);
```

```
Closure  
$factor = 3;  
$multiply = function(int $x) use ($factor): int {  
    return $x * $factor;  
};  
echo $multiply(5); // 15
```

```
Class & Objek  
Definis Class  
class User {  
    public function __construct(  
        private string $name,  
        private int $age = 0,  
    ) {}  
    public function greet(): string { return "Hi, {this->name}"; }  
}
```

```
Inheritance & Interface  
interface Printable {  
    public function toString(): string;  
}  
class Admin extends User implements Printable {  
    public function toString(): string { return "Admin"; }  
}
```

```
Visibility  
public Dapat diakses dari mana saja  
protected Dapat diakses dari class dan subclass  
private Hanya dapat diakses di dalam class  
readonly Hanya bisa diassign sekali (PHP 8.1+)  
static Milik class, bukan instance  
abstract Harus diimplementasikan oleh subclass
```

```
Trait  
trait Timestamped {  
    public function createdAt(): string {  
        return date('Y-m-d H:i:s');  
    }  
}  
class Post { use Timestamped; }
```

```
Penanganan Error  
Try / Catch / Finally  
try {  
    $result = riskyOperation();  
} catch (InvalidArgumentException $e) {  
    echo "Input buruk: " . $e->getMessage();  
} catch (Exception $e) {  
    echo "Error: " . $e->getMessage();  
} finally { cleanup(); }
```

```
Exception Kustom  
class ApiException extends RuntimeException {  
    public function __construct(string $message, private int $statusCode = 500) {  
        parent::__construct($message, $statusCode);  
    }  
}
```

```
Keamanan Null (PHP 8+)  
$len = $user?->address?->zip; // operator nullsafe  
$name = $input ?? "default"; // null coalescing  
$data ??= []; // null coalescing assignment
```

```
File I/O  
Baca & Tulis File  
$content = file_get_contents("data.txt");  
file_put_contents("out.txt", $content);  
$lines = file("data.txt", FILE_IGNORE_NEW_LINES);
```

```
File Handle  
$f = fopen("log.txt", "a");  
fwrite($f, "entry\n");  
fclose($f);
```

```
Fungsi File  
file_exists($path) Periksa apakah file ada  
is_dir($path) Periksa apakah path adalah direktori  
mkdir($path, 0755, true) Buat direktori secara rekursif  
unlink($path) Hapus file  
glob('*.*txt') Temukan file yang cocok dengan pola  
realpath($path) Resolve path absolut penuh
```

```
Database  
Koneksi PDO  
$pdo = new PDO(  
    "mysql:host=localhost;dbname=app",  
    "user", "password",  
    [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]  
);
```

```
Prepared Statement  
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");  
$stmt->execute(["id" => 42]);  
$user = $stmt->fetch(PDO::FETCH_ASSOC);
```

```
Insert & Update  
$stmt = $pdo->prepare("INSERT INTO users (name, email) VALUES  
(?, ?)");  
$stmt->execute(["Alice", "alice@example.com"]);  
$id = $pdo->lastInsertId();
```

```
Mode Fetch PDO  
fetch() Ambil satu baris
```

```
fetchAll() Ambil semua baris  
FETCH_ASSOC Kembalikan sebagai array asosiatif  
FETCH_OBJ Kembalikan sebagai objek anonim  
FETCH_CLASS Kembalikan sebagai instance class tertentu
```

```
Fungsi Umum  
JSON  
$json = json_encode(["name" => "Alice", "age" => 30]);  
$data = json_decode($json, true); // true = array asosiatif  
$data = json_decode($json); // objek
```

```
Tanggal & Waktu  
echo date("Y-m-d H:i:s"); // 2026-03-26 12:00:00  
$ts = strtotime("+1 week");  
$dt = new DateTime("2026-01-01");  
echo $dt->format("D, M j"); // Thu, Jan 1
```

```
Matematika & Random  
abs($n) Nilai absolut  
round($n, 2) Bulatkan ke 2 desimal  
ceil($n) / floor($n) Bulatkan ke atas / bawah  
min($a, $b) / max($a, $b) Minimum / maksimum  
random_int(1, 100) Integer acak kriptografis aman  
number_format($n, 2) Format dengan pemisah ribuan
```

```
Regular Expression  
preg_match('/[a-z]+$/i', $str, $matches);  
preg_match_all('/<\/>+', $str, $all);  
$result = preg_replace('/\s+/', ' ', $str);
```