



Web Service

Make everything happen,
but rarely in the spotlight

Struktur Kontrol

Struktur kontrol (*control structure*) adalah konsep dalam pemrograman yang memungkinkan kita untuk menentukan urutan, alur, dan kondisi eksekusi dari sebuah program. Struktur kontrol ini digunakan untuk mengarahkan bagaimana sebuah program berjalan, memilih di antara berbagai cabang logika, dan mengatur pengulangan. Struktur kontrol merupakan inti dari logika pemrograman.

if/if else

if statement merupakan fundamental statement yang memungkinkan PHP untuk membuat keputusan apakah mengeksekusi program atau tidak.

```
if.php x
belajarpHP > if.php
1 <?php
2 $angka = 10;
3
4 if ($angka > 5) {
5     echo "Angka lebih besar dari 5";
6 }
7
8 ?>
```

```
ifelse1.php x
belajarpHP > ifelse1.php
1 <?php
2 $angka = 3;
3
4 if ($angka > 5) {
5     echo "Angka lebih besar dari 5";
6 } else {
7     echo "Angka kurang dari atau sama dengan 5";
8 }
9
10 ?>
```

```
ifelse2.php x
belajarpHP > ifelse2.php
1 <?php
2 $angka = 5;
3
4 if ($angka > 5) {
5     echo "Angka lebih besar dari 5";
6 } elseif ($angka == 5) {
7     echo "Angka sama dengan 5";
8 } else {
9     echo "Angka kurang dari 5";
10 }
11
12 ?>
```

for

for adalah perintah perulangan dalam berbagai bahasa pemrograman yang digunakan untuk menjalankan blok kode secara berulang dengan jumlah iterasi yang telah ditentukan. Struktur dari for tampak seperti berikut ini.

```
for.php x
belajarphp > for.php
1 <?php
2 for ($i = 1; $i <= 5; $i++) {
3     echo "Angka : " . $i . "<br>";
4 }
5
6 ?-|
```

foreach

foreach adalah sebuah perintah atau konstruksi yang digunakan untuk iterasi atau perulangan pada elemen-elemen dalam sebuah koleksi, seperti array, list, set, atau struktur data lainnya. Pada dasarnya, foreach memungkinkan kita untuk menelusuri setiap elemen dalam koleksi tersebut tanpa harus secara manual mengelola indeks seperti dalam perulangan for tradisional.

```
foreach.php x
belajarphp > foreach.php
1 <?php
2 $buah = array("Apel", "Jeruk", "Pisang", "Mangga");
3
4 foreach ($buah as $item) {
5     echo "Buah: " . $item . "<br>";
6 }
7
8 ?>
```

while

while adalah sebuah perintah atau konstruksi dalam bahasa pemrograman yang digunakan untuk melakukan perulangan selama kondisi tertentu terpenuhi. Selama kondisi yang diberikan bernilai benar (true), blok kode di dalam perulangan akan terus dieksekusi. Jika kondisi menjadi salah (false), perulangan akan berhenti.

```
while.php x
belajarphp > while.php
1 <?php
2 $counter = 1;
3
4 while ($counter <= 5) {
5     echo "Angka : " . $counter . "<br>";
6     $counter++;
7 }
8
9 ?>
```



Web Service

break

break adalah perintah yang digunakan untuk menghentikan atau keluar dari loop.

```
break1.php ×
belajarphp > break1.php
1 <?php
2 for ($i = 1; $i <= 10; $i++) {
3     if ($i == 6) {
4         break; // Hentikan loop ketika $i sama dengan 6
5     }
6     echo "Angka : " . $i . "<br>";
7 }
8
9 ?>
```

```
break2.php ×
belajarphp > break2.php
1 <?php
2 $counter = 1;
3
4 while ($counter <= 10) {
5     if ($counter == 4) {
6         break; // Hentikan loop ketika $counter sama dengan 4
7     }
8     echo "Counter: " . $counter . "<br>";
9     $counter++;
10 }
11
12 ?>
```



Web Service

continue

continue adalah perintah yang digunakan untuk melewati iterasi yang sedang berjalan dalam sebuah loop dan melanjutkan ke iterasi berikutnya.

```
continue1.php ×
belajarphp > continue1.php
1 <?php
2 for ($i = 1; $i <= 10; $i++) {
3     if ($i == 5) {
4         continue; // Lewati iterasi ketika $i sama dengan 5
5     }
6     echo "Angka: " . $i . "<br>";
7 }
8
9 ?>
```

```
continue2.php ×
belajarphp > continue2.php
1 <?php
2 $counter = 1;
3
4 while ($counter <= 10) {
5     $counter++;
6
7     if ($counter == 7) {
8         continue; // Lewati iterasi ketika $counter sama dengan 7
9     }
10    echo "Counter: " . $counter . "<br>";
11 }
12
13 ?>
```

Program Kalkulator Dinamis

Berikut ini merupakan contoh program kalkulator sederhana dengan inputan secara dinamis melalui REST API Client dan output berupa JSON.

```
kalkulator.php x
simpleapi > kalkulator.php
1 <?php
2 $bilangan1 = $_POST["bilangan1"];
3 $bilangan2 = $_POST["bilangan2"];
4
5 $hasil = $bilangan1 + $bilangan2;
6
7 // RETURN DATA BENTUK JSON
8 echo json_encode([
9     'status' => 'success',
10    'hasil' => $hasil,
11 ]);
12
13 ?>
```

Lakukan request pada REST API Client.

The screenshot shows a REST API Client interface for a request to `http://localhost/materiwebservice/simpleapi/kalkulator.php`. The request method is `POST`. The request body is in `Form` format, with fields `bilangan1` (value 4) and `bilangan2` (value 2). The response status is `200 OK`, with a size of `30 Bytes` and a time of `9 ms`. The response body is a JSON object: `{ "status": "success", "hasil": 6 }`.