

IF ... ELSE ref »

If ... elseif ... else

Satu statemen, tanpa kurung kurawal

```
if ($a > $b)
    echo $a . ' lebih besar dari ' . $b;
else
    echo $a . ' kurang dari ' . $b;
```

atau:

```
if ($a > $b) echo $a ' lebih besar';
$b = 10;
```

Menggunakan kurung kurawal

```
if ($a > $b) {
    echo $a . ' lebih besar dari ' . $b;
} elseif ($a == $b) {
    echo $a . ' sama dengan ' . $b;
} else {
    echo $a . ' lebih kecil dari ' . $b;
}
```

Menggunakan colon, gunakan elseif bukan else if

```
if ($a > $b):
    echo $a . ' lebih besar dari ' . $b;
elseif ($a == $b):
    echo $a . ' sama dengan ' . $b;
else:
    echo $a . ' lebih kecil dari ' . $b;
endif;
```

if...else (ternary operator)

```
$a = 9;
$val = $a > 7 ? 'lulus' : 'gagal'; // lulus
```

Mulai PHP 5.3, dapat menggunakan ?:

```
$val = isset($b) ? $b : 'lainnya'; // lainnya
$val = @$b ?: 'lainnya'; // lainnya
```

Keterangan: karena variabel \$b tidak dideklarasikan maka variabel \$val bernilai 'lainnya'

Mix: PHP dan HTML

```
<?php if ($a > $b) {?>
    <p> $a lebih besar dari $b </p>
<?php } else {?>
    <p> $a lebih kecil dari $b </p>
<?php } ?>
```

Dengan colon:

```
<?php if ($a > $b) :?>
    <p> $a lebih besar dari $b </p>
<?php else :?>
    <p> $a lebih kecil dari $b </p>
<?php endif ?>
```

Tes apakah variabel ada nilainya (kondisi: variabel sudah dideklarasikan)

```
$a = 5; $b = '';
if ($b) {
    echo $a * $b;
} else {
    echo $a * 3;
} // 15
```

Dengan operator !

```
$a = 5; $b = '';
if (! $b) {
    $b = 3;

echo $a * $b; // 15
```

Tes menggunakan operator perbandingan

```
if ($a != $b) { ... } // $a tidak sama $b
if ($a <> $b) { ... } // $a tidak sama $b
if ($a == $b) { ... } // $a sama dengan $b
if ($a === $b) { ... } // $a sama dengan $b
baik nilai maupun tipe datanya
if ($a >= $b) { ... } // $a lebih besar atau sama dengan $b
if ($a <= $b) { ... } // $a lebih kecil atau sama dengan $b
```

SWITCH ... CASE ref »

switch...case

Gunakan switch ... case jika yang diuji hanya satu nilai, contoh menguji nilai variabel \$a. Switch case lebih mudah dipahami dan lebih cepat dieksekusi daripada IF ELSE

Menguji variabel \$a dengan IF ELSE

```
if ($a == 'A') {
    echo 'Baik Sekali';
} elseif ($a == 'B') {
    echo 'Baik';
} elseif ($a == 'C') {
    echo 'Cukup';
} else {
    echo 'Kurang';
}
```

Dengan switch ... case

```
switch ($a) {
    case 'A':
        echo 'Baik Sekali'; break;
    case 'B':
        echo 'Baik'; break;
    case 'C':
        echo 'Cukup'; break;
    default:
        echo 'Kurang';
}
```

IF ELSE dengan OR

```
if ($a == 'A' || $a == 'B') {
    echo 'Lulus';
} elseif ($a == 'C') {
    echo 'Mengulang';
} else {
    echo 'Tidak lulus';
}
```

Dengan switch case

```
switch ($a) {
    case 'A':
    case 'B':
        echo 'Lulus';
        break;
    case 'C':
        echo 'Mengulang';
        break;
    default:
        echo 'Tidak lulus';
}
```

if ... else di dalam switch ... case

```
switch ($a) {
    case 'A':
    case 'B':
        if ($b >= 8) {
            echo 'Lulus';
        } else {
            echo 'Lulus bersyarat';
        }
        break;
    case 'C':
        echo 'Mengulang'; break;
    default:
        echo 'Tidak lulus';
}
```

FOR ref » | WHILE ref » | FOREACH ref »

for () {...}

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
} // 12345678910
```

Break digunakan untuk menghentikan loop

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
    if ($i == 5)
        break;
} // 12345678910
```

Continue digunakan untuk melanjutkan loop

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
    if ($i == 5) continue;
} // 1234678910 -> tanpa 5
```

while () {...}

```
$i = 1;
while ($i <= 10) {
    echo $i;
    $i++;
} // 12345678910
```

Break digunakan untuk menghentikan loop

```
$i = 1;
while ($i <= 10):
    echo $i;
    if ($i == 5)
        break;
    $i++;
endwhile; // 12345
```

do () while{...}

```
do {
    statement; // dieksekusi terlebih dahulu
} while (ekspresi);
```

Pada do while, setidaknya statement akan di eksekusi sekalii, selanjutnya di dilakukan pengecekan pada ekspresi

```
$i = 1;
do {
    echo $i;
} while ($i < 0); // 1
```

Keterangan: pertama \$i dicetak, selanjutnya karena ekspresi tidak terpenuhi, loop dihentikan.

foreach () {...}

foreach digunakan pada array

```
$arr = array ('foo', 'bar');
foreach ($arr as $val) {
    echo $val . ', ';
} // foo, bar,
```

Associative array

```
$arr = array ('kendaraan' => 'mobil',
              'warna' => 'merah',
              'tipe' => 'mpv'
            );
foreach ($arr as $item => $deskripsi) {
    echo $item . ':' . $deskripsi;
}
// kendaraan: mobil warna: merah tipe: mpv
```

Gunakan continue untuk melewati loop:

```
foreach ($arr as $item => $deskripsi) {
    if ($item == 'warna')
        continue;
    echo $item . ':' . $deskripsi;
}
// kendaraan: mobil tipe: mpv
```