

LAPORAN PROYEK 4

STRUKTUR KONTROL PERULANGAN DAN ARRAY PHP



OLEH:
M HAFIDZ RAMADHAN NST
(NISN. 0083825885)

REKAYASA PERANGKAT LUNAK
SMK NEGERI 1 KARANG BARU
PEMERINTAH PROVINSI ACEH
2024



MATERI PERTEMUAN 8

Memahami Perulangan Pemrograman



TAHUKAH KAMU...?

- Ada 6 Jenis percabangan dalam pemrograman PHP yang harus kita ketahui:
- Percabangan If
 - Percabangan If/Else
 - Percabangan If/Elseif/Else
 - Percabangan Switch/Case
 - Percabangan dengan Operator Ternary
 - Percabangan Bersarang

Perulangan, Iteration dalam Pemrograman

Apa yang akan kamu lakukan bila disuruh membuat daftar judul artikel dengan PHP?

Apakah akan mencetaknya satu per satu dengan perintah **echo** seperti ini:

```
<?php
```

```
echo "<h2>Belajar Pemrograman PHP untuk Pemula</h2>";
echo "<h2>Cara Menggunakan Perulangan di PHP</h2>";
echo "<h2>Memahami Struktur Kondisi IF di PHP</h2>";
echo "<h2>Memahami Perulangan di PHP</h2>";
echo "<h2>Prosedur dan Fungsi di PHP</h2>";

?>
```

Bisa saja itu dilakukan. Tapi masalahnya, Bagaimana kalau datanya ada 100 atau 1000? Apakah kita mampu mengetik semuanya? Pasti capek. Karena itu, kita harus menggunakan perulangan.

Ada dua jenis perulangan dalam pemrograman: *Counted loop* adalah perulangan yang sudah jelas banyak pengulangannya. Sedangkan *Uncounted loop* tidak pasti berapa kali dia akan mengulang. Manakah diantara keempat perulangan tersebut yang termasuk ke dalam *counted loop* dan *uncounted loop*? Mari kita bahas.

1. Perulangan For

Perulangan *For* adalah perulangan yang termasuk dalam *counted loop*, karena kita bisa menentukan jumlah perulangannya.

Bentuk dasar perulangan for:

```
<?php
```

```
for ($i = 0; $i < 10; $i++){
    // blok kode yang akan diulang di sini!
}

?>
```

Varibel **\$i** dalam perulangan *For* berfungsi sebagai *counter* yang menghitung berapa kali ia akan mengulang.

Hitungan akan dimulai dari nol (**0**), karena kita memberikan nilai **\$i = 0**.

Lalu, perulangan akan diulang selama nilai **\$i** lebih kecil dari **10**. Artinya, perulangan ini akan mengulang sebanyak **10x**.

Maksud dari **\$i++** adalah nilai **\$i** akan ditambah **1** disetiap kali melakukan perulangan.

Contoh:

```
<?php
```

```

for($i = 0; $i <= 10; $i++){
    echo "<h2>Ini perulangan ke-$i</h2>";
}
?>

```

Hasilnya:

```

<?php
for($i = 0; $i <= 10; $i++){
    echo "<h2>Ini perulangan ke-$i</h2>";
}
?>

```

Ini perulangan ke-0
Ini perulangan ke-1
Ini perulangan ke-2
Ini perulangan ke-3
Ini perulangan ke-4
Ini perulangan ke-5
Ini perulangan ke-6
Ini perulangan ke-7
Ini perulangan ke-8
Ini perulangan ke-9
Ini perulangan ke-10

```

<?php
for($i = 0; $i <= 10; $i++){
    echo "<h2>Ini perulangan ke-$i</h2>";
}
?>

```

Ini perulangan ke-0
Ini perulangan ke-1
Ini perulangan ke-2
Ini perulangan ke-3
Ini perulangan ke-4
Ini perulangan ke-5
Ini perulangan ke-6
Ini perulangan ke-7
Ini perulangan ke-8
Ini perulangan ke-9
Ini perulangan ke-10

By M HAFIDZ RAMADHAN NST

2. Perulangan While

Perulangan *while* adalah perulangan yang termasuk dalam *uncontrolled loop*. Karena biasanya digunakan untuk mengulang sesuatu yang belum jelas jumlah pengulangannya.

Namun, perulangan *while* juga bisa digunakan seperti perulangan *for* sebagai *counted loop*.

Bentuk dasarnya:

```
<?php
```

```

while (<kondisi>){
    // blok kode yang akan diulang di sini
}

```

?>

Contoh:

```
<?php
```

```
$ulangi = 0;
```

```

while($ulangi < 10){
    echo "<p>Ini adalah perulangan ke-$ulangi</p>";
    $ulangi++;
}

```

?>

Hasilnya:

The screenshot shows a browser window with the URL `localhost/AHMADIXIRPL2024/ALGORITMA4/02%20Perul...`. The page displays the output of a PHP script. The code in the editor is:

```
<?php  
$ulangi = 0;  
while($ulangi < 10){  
    echo "<p>Ini adalah perulangan ke-$ulangi</p>";  
    $ulangi++;  
}
```

The output on the page is a series of nine lines, each containing the text "Ini adalah perulangan ke-1" through "Ini adalah perulangan ke-9".

The screenshot shows a browser window with the URL `localhost/hafiz%20X%20rp%202/ALGORITMA%204/02%20Perulangan%20while.php`. The page displays the output of a PHP script. The code in the editor is:

```
<?php  
$ulangi = 0;  
while($ulangi < 10){  
    echo "<p>Ini adalah perulangan ke-$ulangi</p>";  
    $ulangi++;  
}  
print_r("<p><strong>iBy M HAFIDZ RAMADHAN NST!</strong></p>");
```

The output on the page is a series of nine lines, each containing the text "Ini adalah perulangan ke-1" through "Ini adalah perulangan ke-9", followed by a line "By M HAFIDZ RAMADHAN NST!".

Perulangan *while* akan terus mengulang selama nilai **\$ulangi** lebih kecil dari **10**.

Lalu di dalam perulangan kita melakukan *increment* nilai **\$ulangi** dengan **\$ulangi++**.

Artinya: Tambah 1 disetiap pengulangan.

Hati-hati, jangan sampai lupa menambahkan *increment*, atau kode yang akan mempengaruhi pengulangan. Karena kalau tidak, pengulangannya tidak akan pernah berhenti dan akan membuat komputer kita *hang*.

3. Perulangan Do/While

Perulangan *Do/While* sama seperti perulangan *while*. Ia juga tergolong dalam *uncounted loop*.

Perbedaan *Do/While* dengan *while* terletak pada cara iya memulai pengulangan.

Perulangan *Do/While* akan selalu melakukan pengulangan sebanyak 1 kali, kemudian melakukan pengecekan kondisi.

Sedangkan perulangan *while* akan mengecek kondisi terlebih dahulu, baru melakukan pengulangan.

Bentuk perulangan *Do/While*:

```
<?php  
do {  
    // blok kode yang akan diulang  
} while (<kondisi>);  
?>
```

Contoh:

```
<?php  
$ulangi = 10;  
  
do {  
    echo "<p>ini adalah perulangan ke-$ulangi</p>";
```

```

    $ulangi--;
} while ($ulangi > 0);
?>

```

Hasilnya:

```

ini adalah perulangan ke-10
ini adalah perulangan ke-9
ini adalah perulangan ke-8
ini adalah perulangan ke-7
ini adalah perulangan ke-6
ini adalah perulangan ke-5
ini adalah perulangan ke-4
ini adalah perulangan ke-3
ini adalah perulangan ke-2
ini adalah perulangan ke-1

```

```

ini adalah perulangan ke-10
ini adalah perulangan ke-9
ini adalah perulangan ke-8
ini adalah perulangan ke-7
ini adalah perulangan ke-6
ini adalah perulangan ke-5
ini adalah perulangan ke-4
ini adalah perulangan ke-3
ini adalah perulangan ke-2
ini adalah perulangan ke-1

```

4. Perulangan Foreach

Perulangan *foreach* sama seperti perulangan *for*. Namun, ia lebih khusus digunakan untuk mencetak array.

Bentuk perulangan *foreach*:

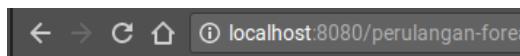
```
<?php
foreach($array as $data) {
    echo $data;
}
```

Contoh:

```
<?php
$books = [
    "Panduan Belajar PHP untuk Pemula",
    "Membangun Aplikasi Web dengan PHP",
    "Tutorial PHP dan MySQL",
    "Membuat Chat Bot dengan PHP"
];

echo "<h5>Judul Buku PHP:</h5>";
echo "<ul>";
foreach($books as $buku) {
    echo "<li>$buku</li>";
}
echo "</ul>";
?>
```

Hasilnya:



Judul Buku PHP:

- Panduan Belajar PHP untuk Pemula
- Membangun Aplikasi Web dengan PHP
- Tutorial PHP dan MySQL
- Membuat Chat Bot dengan PHP

The screenshot shows a code editor with several PHP files in the Explorer panel. The active file is '04 Perulangan Foreach.php'. The code contains a nested loop structure. The browser preview shows the output of the script, which lists the titles of four books under the heading 'Judul Buku PHP:'. Below the list is the text 'By M HAFIDZ RAMADHAN NST!'. The code editor also shows other files like '01 Perulangan for.php', '02 Perulangan while.php', and '03 While.php'.

```
<?php
$books = [
    "Panduan Belajar PHP untuk Pemula",
    "Membangun Aplikasi Web dengan PHP",
    "Tutorial PHP dan MySQL",
    "Membuat Chat Bot dengan PHP"
];
echo "<h2>Judul Buku PHP:</h2>";
echo "<ul>";
foreach($books as $buku){
    echo "<li>$buku</li>";
}
echo "</ul>";
print_r("<p><strong><i>By M HAFIDZ RAMADHAN NST!</i></strong></p>");
?>
```

Perulangan Bersarang

Perulangan bersarang adalah istilah untuk menyebut perulangan di dalam perulangan. Dalam bahasa Inggris, perulangan bersarang disebut *nested loop*.

Contoh perulangan bersarang:

```
<?php
for($i = 0; $i < 5; $i++) {
    for($j = 0; $j < 10; $j++) {
        echo "Ini perulangan ke ($i, $j)<br>";
    }
}
?>
```

Contoh lain:

```
<?php
$i = 0;
while($i < 10) {
    for($j = 0; $j < 10; $j++) {
        echo "Ini perulangan ke ($i, $j)<br>";
    }
    $i++;
}
```

Hasilnya:

The screenshot shows a browser window displaying the output of the nested loop code. The output consists of two sets of nested loops. The outer loop runs from 0 to 5, and the inner loop runs from 0 to 9. Each iteration of the inner loop outputs a message 'Ini perulangan ke (i, j)' where i is the current value of the outer loop and j is the current value of the inner loop. For example, it prints 'Ini perulangan ke (0, 0)', 'Ini perulangan ke (0, 1)', up to 'Ini perulangan ke (0, 9)', then 'Ini perulangan ke (1, 0)', and so on until 'Ini perulangan ke (1, 9)'.

```
Ini perulangan ke (0, 0)
Ini perulangan ke (0, 1)
Ini perulangan ke (0, 2)
Ini perulangan ke (0, 3)
Ini perulangan ke (0, 4)
Ini perulangan ke (0, 5)
Ini perulangan ke (0, 6)
Ini perulangan ke (0, 7)
Ini perulangan ke (0, 8)
Ini perulangan ke (0, 9)
Ini perulangan ke (1, 0)
Ini perulangan ke (1, 1)
Ini perulangan ke (1, 2)
Ini perulangan ke (1, 3)
Ini perulangan ke (1, 4)
Ini perulangan ke (1, 5)
Ini perulangan ke (1, 6)
```

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/05%20Perulangan%20Bersang.php
Ini perulangan ke (0, 0)
Ini perulangan ke (0, 1)
Ini perulangan ke (0, 2)
Ini perulangan ke (0, 3)
Ini perulangan ke (0, 4)
Ini perulangan ke (0, 5)
Ini perulangan ke (0, 6)
Ini perulangan ke (0, 7)
Ini perulangan ke (0, 8)
Ini perulangan ke (0, 9)
Ini perulangan ke (0, 10)
Ini perulangan ke (0, 11)
Ini perulangan ke (1, 0)
Ini perulangan ke (1, 1)
Ini perulangan ke (1, 2)
Ini perulangan ke (1, 3)
Ini perulangan ke (1, 4)
Ini perulangan ke (1, 5)
Ini perulangan ke (1, 6)
Ini perulangan ke (1, 7)
Ini perulangan ke (1, 8)
Ini perulangan ke (1, 9)
Ini perulangan ke (1, 10)
Ini perulangan ke (1, 11)
Ini perulangan ke (2, 0)
Ini perulangan ke (2, 1)
Ini perulangan ke (2, 2)
Ini perulangan ke (2, 3)
Ini perulangan ke (2, 4)
Ini perulangan ke (2, 5)
Ini perulangan ke (2, 6)
Ini perulangan ke (2, 7)
Ini perulangan ke (2, 8)
Ini perulangan ke (2, 9)
Ini perulangan ke (2, 10)
Ini perulangan ke (2, 11)
Ini perulangan ke (3, 0)
Ini perulangan ke (3, 1)
Ini perulangan ke (3, 2)
Ini perulangan ke (3, 3)
Ini perulangan ke (3, 4)
Ini perulangan ke (3, 5)
Ini perulangan ke (3, 6)
Ini perulangan ke (3, 7)
Ini perulangan ke (3, 8)
Ini perulangan ke (3, 9)
Ini perulangan ke (3, 10)
Ini perulangan ke (3, 11)
Ini perulangan ke (4, 0)
Ini perulangan ke (4, 1)
Ini perulangan ke (4, 2)
Ini perulangan ke (4, 3)

```

Line 4, Col 24 - Spaces: 4 - UTF-8 - CRLF - PHP - 6

Pernyataan While

while membuat sebuah loop yang menjalankan blok kode selama kondisi tertentu terpenuhi.

```
while ($kondisi) {
    // Kode yang akan dijalankan
}
```

Contoh penggunaan:

```
$i = 1;
while ($i <= 5) {
    echo "Nilai i: $i";
    $i++;
}
```

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/06%20Pernyataan%20While.php
Nilai i: 1Nilai i: 2Nilai i: 3Nilai i: 4Nilai i: 5Nilai i: 6Nilai i: 7Nilai i: 8Nilai i: 9Nilai i: 10
By M HAFIDZ RAMADHAN NSTI

```

Pernyataan Do-While

do-while serupa dengan while, tetapi memastikan bahwa kode dijalankan setidaknya satu kali.

```
do {
    // Kode yang akan dijalankan
} while ($kondisi);
```

Contoh penggunaan:

```
$i = 1;
do {
    echo "Nilai i: $i";
    $i++;
} while ($i <= 5);
```

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/07%20Pernyataan%20Do-While.php
Nilai i: 1Nilai i: 2Nilai i: 3Nilai i: 4Nilai i: 5Nilai i: 6Nilai i: 7Nilai i: 8Nilai i: 9
By M HAFIDZ RAMADHAN NSTI

```

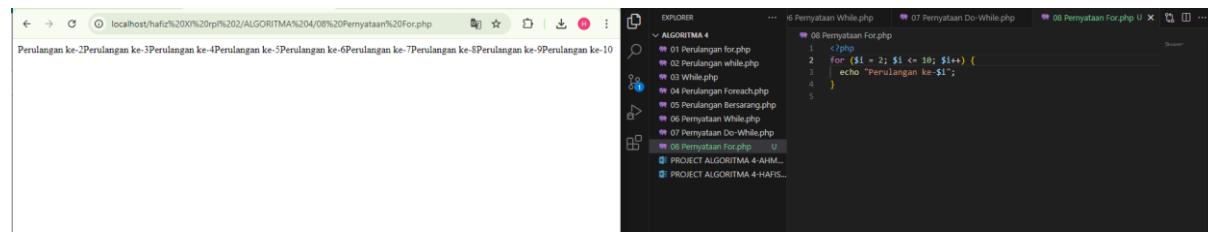
Pernyataan For

for loop digunakan ketika kamu tahu sebelumnya berapa kali kode harus diulang.

```
for (inisialisasi; kondisi; perubahan) {
    // Kode yang akan dijalankan
}
```

Contoh penggunaan:

```
for ($i = 1; $i <= 5; $i++) {
    echo "Perulangan ke-$i";
```



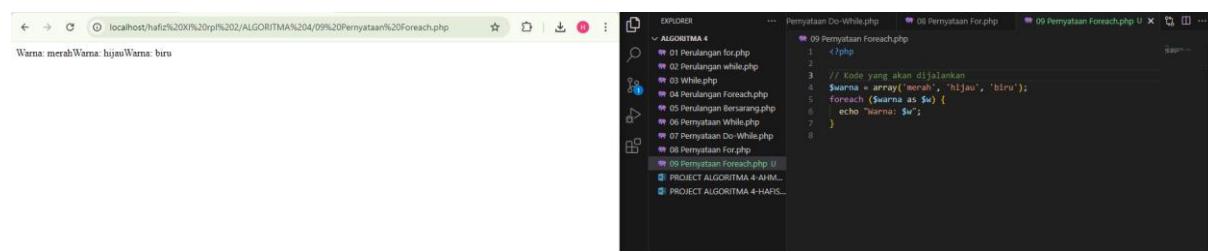
Pernyataan Foreach

Pada PHP, pernyataan foreach sangat berguna untuk mengulangi semua elemen dalam array.

```
foreach ($array sebagai $nilai) {
    // Kode yang akan dijalankan
}
```

Contoh penggunaan:

```
$warna = array('merah', 'hijau', 'biru');
foreach ($warna as $w) {
    echo "Warna: $w";
```



Struktur kontrol adalah kunci untuk membuat kode yang efisien dan mudah dibaca. Dengan menguasai pernyataan kondisional dan perulangan di PHP, kamu akan bisa mengatur bagaimana dan kapan kode dieksekusi dengan lebih tepat.

PENALARAN

I. Penulisan For Pada PHP

Perulangan for pada PHP dapat ditulis menggunakan kurung kurawa, colon, atau tanpa keduanya:

```
// Kurung Kurawa, paling umum digunakan
for (ekspresi1; ekspresi2 ; ekspresi3) {
    // kode
}

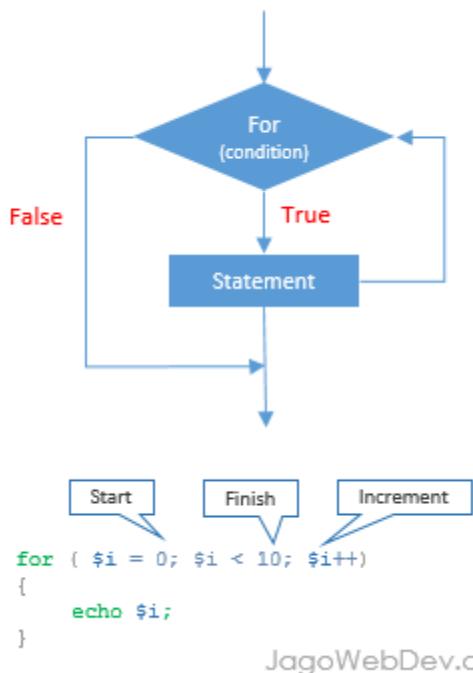
// Colon
for (ekspresi1; ekspresi2 ; ekspresi3) :
    // kode
endfor;

// Tanpa Keduanya
for (ekspresi1; ekspresi2 ; ekspresi3)
    statement;
```

Dalam menjalankan fungsi loop, PHP akan melakukan eksekusi dengan urutan sebagai berikut:

- PHP akan membaca ekspresi1
- Selanjutnya PHP akan mengevaluasi ekspresi2, jika nilanya TRUE, maka statement di dalam kurung kurawa dijalankan, jika bernilai FALSE maka loop dihentikan.

- Setelah itu PHP akan mengevaluasi atau menjalankan ekspresi3 begitu seterusnya hingga loop selesai, jika digambarkan dalam bentuk flowchart:



Ketentuan mengenai ekspresi

Beberapa ketentuan terkait penulisan ekspresi:

- Semua ekspresi dapat bernilai kosong atau dapat bernilai lebih dari satu dengan pemisah tanda koma
- Semua ekspresi pada ekspresi2 akan di evaluasi, namun untuk menentukan nilai (TRUE atau FALSE – yang menentukan loop berhenti atau berlanjut), digunakan ekspresi yang terakhir.
- Jika ekspresi2 yang bernilai kosong maka loop akan dijalankan terus hingga dihentikan oleh break statemen yang ada di dalam kurung kurawa.

Berikut beberapa contoh penulisan for loop dengan berbagai ekspresi yang semuanya akan mencetak angka 1 s.d 10. (Bentuk 1 merupakan bentuk yang **sering (atau selalu)** dipakai (termasuk saya))

```

<?php
/*
Contoh 1, bentuk lengkap, SERING DAN UMUM DIGUNAKAN
*/
for ($i = 1; $i <= 10; $i++) {
    echo $i;
}
echo "<hr>";
/*
Contoh 2, dengan ekspresi2 kosong, kode dihentikan dengan brak statement
*/
for ($i = 1;; $i++) {
    if ($i > 10) {
        break;
    }
    echo $i;
}
echo "<hr>";
/*
Contoh 3, semua ekspresi kosong
*/
$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
}
  
```

```

echo $i;
$i++;
}
echo "<hr>";
/*
Contoh 4, tanpa statement hanya ekspresi saja
*/
for ($i = 1; $i <= 10; print $i, $i++) ;
echo "<hr>";

```

```

12345678910
12345678910
12345678910
12345678910
By M HAFIDZ RAMADHAN NST:

```

```

1 ALGORITMA 4
  01 Perulangan for.php
  02 Perulangan while.php
  03 While.php
  04 Perulangan Foreach.php
  05 Perulangan Bersarang.php
  06 Pernyataan While.php
  07 Pernyataan Do-While.php
  08 Pernyataan For.php
  09 Pernyataan Foreach.php
  10 Pernyataan For Pada PHP.php
PROJECT ALGORITMA 4-AHM...
PROJECT ALGORITMA 4-HAFIS...

```

```

1 <?php
2 for ($i = 1; $i <= 10; $i++) {
3     echo $i;
4 }
5 echo "<hr>";
6 /*
7 Contoh 1, bentuk lengkap, SERING DARI UMUM DIGUNAKAN
8 */
9 for ($i = 1; $i <= 10; $i++) {
10     echo $i;
11 }
12 echo "<hr>";
13 /*
14 Contoh 2, dengan ekspresi kosong, kode dihitung dengan brak stat
15 */
16 for ($i = 1; $i++) {
17     if ($i > 10) {
18         break;
19     }
20     echo $i;
21 }
22 echo "<hr>";
23 /*
24 Contoh 3, semua ekspresi kosong
25 */
26 for (; ; ) {
27     if ($i > 10) {
28         break;
29     }
30     echo $i;
31 }
32 /*
33 Contoh 4, tanpa statement hanya ekspresi saja
34 */
35 for ($i = 1; $i <= 10; print $i, $i++) ;
36 echo "<hr>";
37 print_r("<p><strong><i>By M HAFIDZ RAMADHAN NST</i></strong></p>");
38

```

II. Melompati (skip) For Loop Pada Nilai Tertentu

Pada saat menjalankan loop, terkadang pada kondisi/nilai tertentu, kita ingin melompatinya (skip), untuk keperluan tersebut, kita dapat menggunakan statement continue contoh:

```

<?php
for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        continue;
    }
    echo $i; // 1 2 3 4 6 7 8 9 10
}
?>

```

```

1234678910111213
By M HAFIDZ RAMADHAN NST:

```

```

1 ALGORITMA 4
  01 Perulangan for.php
  02 Perulangan while.php
  03 While.php
  04 Perulangan Foreach.php
  05 Perulangan Bersarang.php
  06 Pernyataan While.php
  07 Pernyataan Do-While.php
  08 Pernyataan For.php
  09 Pernyataan Foreach.php
  10 Pernyataan For Pada PHP.php
  11 II. Melompati (skip) For Loop Pada Nilai Tertentu.php
PROJECT ALGORITMA 4-AHM...
PROJECT ALGORITMA 4-HAFIS...

```

```

1 <?php
2 for ($i = 1; $i <= 13; $i++) {
3     if ($i == 5) {
4         continue;
5     }
6     echo $i; // 1 2 3 4 6 7 8 9 10 12 13
7 }
8
9 print_r("<p><strong><i>By M HAFIDZ RAMADHAN NST</i></strong></p>");
10
11
12
13

```

III. Tips Optimasi Perulangan For Pada PHP

Berikut ini beberapa tips yang dapat kita terapkan agar perulangan for dapat dieksekusi dengan cepat
1 Hindari pemanggilan fungsi dan pendefinisian variabel yang nilainya tetap di dalam loop
Fungsi ini dapat berupa fungsi bawaan PHP seperti count, substr, strlen, dll maupun fungsi yang kita buat sendiri, contoh berikut pengulangan untuk mendapatkan nama bulan:

```

<?php
$bulan = array('1'=>'Januari',
               'Februari',
               'Maret' ,
               'April' ,
               'Mei' ,
               'Juni' ,
               'Juli' ,
               'Agustus' ,

```

```

        'September',
        'Oktober',
        'November',
        'Desember'
    );
$batas_waktu = '2015-11-10';
echo '<table>
    <tr>
        <th>Bulan</th>
        <th>Keterangan</th>
    </tr>';
for ($i = 1; $i <= count($bulan); $i++)
{
    $bln_batas = date("m", strtotime($batas_waktu));
    echo '<tr>
        <td>' . strtoupper($bulan[$i]) . '</td>';

        if ($bln_batas == $i)
            echo '<td>Batas waktu penulisan</td>';
        else
            echo '<td>-</td>';

        echo '</tr>';
}

echo '</table>';
?>
```

output:

| Bulan | Keterangan |
|-----------|-----------------------|
| JANUARI | — |
| FEBRUARI | — |
| MARET | — |
| APRIL | — |
| MEI | — |
| JUNI | — |
| JULI | — |
| AGUSTUS | — |
| SEPTEMBER | — |
| OKTOBER | — |
| NOVEMBER | Batas waktu penulisan |
| DESEMBER | — |

dari contoh tersebut PHP akan: (1) memanggil fungsi `count($bulan)`, (2) fungsi `date("m",strtotime($batas_waktu))` dan (3) mendefinisikan variabel `$bln_batas` berulang ulang, hal tersebut tidak efisien karena akan memakan waktu dan resource.

Berbeda dengan fungsi `strtoupper` yang memang diperlukan di dalam loop, karena nilainya berubah ubah sesuai nama bulan. Untuk itu, fungsi dan variabel yang nilainya tetap sebaiknya didefinisikan di luar loop, kode dapat ditulis ulang menjadi:

```
<?php
$bln_batas = date("m", strtotime($batas_waktu));
$jml_bln = count($bulan);
for ($i = 1; $i <= $jml_bln, $i++)
{
    // code
}
?>
atau
<?php
$bln_batas = date("m", strtotime($batas_waktu));
for ($i = 1, $jml_bln = count($bulan); $i <= $jml_bln, $i++)
{
```

```
//code
}
?>
```

dari contoh diatas, waktu eksekusi lebih cepat karena fungsi count dan date, serta pendefinisian variabel \$bln_batas hanya dijalankan sekali. Dalam kode diatas terdapat \$i++, kode tersebut merupakan kependekan dari \$i = \$i + 1.

Praktek di lapangan bisa menjadi lebih kompleks, misal dengan contoh diatas, kita akan menambahkan informasi deadline suatu tahapan, misal output yang diinginkan adalah:

| Bulan | Deadline | |
|-----------|-------------|------------|
| | Tahapan | Tanggal |
| JANUARI | Perencanaan | 2015-01-31 |
| FEBRUARI | Analisis | 2015-02-28 |
| MARET | Perancangan | 2015-03-31 |
| APRIL | Penerapan | 2015-04-30 |
| MEI | Evaluasi | 2015-05-31 |
| JUNI | Penggunaan | 2015-06-30 |
| JULI | — | — |
| AGUSTUS | — | — |
| SEPTEMBER | — | — |
| OKTOBER | — | — |
| NOVEMBER | — | — |
| DESEMBER | — | — |

Kode yang kita gunakan:

```
<?php
$bulan = array('1'=>'Januari',
               'Februari',
               'Maret' ,
               'April' ,
               'Mei' ,
               'Juni' ,
               'Juli' ,
               'Agustus',
               'September',
               'Oktober',
               'November',
               'Desember'
             );
// Informasi yang diperoleh dari database
$stahapan = array(
array('tahap' => 'Perencanaan',      'tgl' => '2015-01-31'),
array('tahap' => 'Analisis',        'tgl' => '2015-02-28'),
array('tahap' => 'Perancangan',     'tgl' => '2015-03-31'),
array('tahap' => 'Penerapan',       'tgl' => '2015-04-30'),
array('tahap' => 'Evaluasi',        'tgl' => '2015-05-31'),
array('tahap' => 'Penggunaan',      'tgl' => '2015-06-30')
);
echo '<table>
          <tr>
            <th rowspan="2">Bulan</th>
            <th colspan="2">Deadline</th>
          </tr>
          <tr>
            <th>Tahapan</th>
            <th>Tanggal</th>
          </tr>
        ';
$jml_bln =
for ($i = 1; $i <= count($bulan); $i++)
```

```

{
    echo '<tr>
        <td> ' . strtoupper($bulan[$i]) . ' </td>';

        $data_tahapan = false;
        foreach ($tahapan as $tahap)
        {
            $bln_batas = date("m", strtotime($tahap['tgl']));
            if ($bln_batas == $i) {
echo '<td>' . $tahap['tahap'] . '</td>
        <td>' . $tahap['tgl'] . '</td>';
$data_tahapan = true;
            }

        }
        if (!$data_tahapan)
echo '<td>-</td>
        <td>-</td>';

        echo '</tr>';
}
echo '</table>';
?>
```

dari data diatas terdapat pengulangan fungsi date yaitu sebanyak 72 kali (12 x 6), dengan struktur data seperti diatas, agak ribet jika harus memenuhi kondisi ideal seperti contoh sebelumnya.

Kondisi tersebut dapat dipenuhi, namun kode yang ditulis bisa jadi menjadi lebih kompleks dan membutuhkan tenaga yang lebih untuk memahaminya (tergantung kondisi lapangan).

Pada kondisi ini, kondisi ideal dapat dilanggar jika kode yang ditulis menjadi lebih sederhana dan mudah dipahami dan performa aplikasi juga tidak terganggu.

Namun jika tidak ada salahnya kita mencobanya. Dengan sedikit perubahan, kode diatas dapat kita tulis kembali menjadi:

```
<?php
$bulan = array('1'=>'Januari',
               'Februari',
               'Maret',
               'April',
               'Mei',
               'Juni',
               'Juli',
               'Agustus',
               'September',
               'Oktober',
               'November',
               'Desember'
               );

// Informasi yang diperoleh dari database
$tahapan = array(
array('tahap' => 'Perencanaan',      'tgl' => '2015-01-31'),
array('tahap' => 'Analisis',          'tgl' => '2015-02-28'),
array('tahap' => 'Perancangan',       'tgl' => '2015-03-31'),
array('tahap' => 'Penerapan',         'tgl' => '2015-04-30'),
array('tahap' => 'Evaluasi',          'tgl' => '2015-05-31'),
array('tahap' => 'Penggunaan',        'tgl' => '2015-06-30')
);

foreach ($tahapan as $key => $tahap)
{
    $bln_batas = date("n", strtotime($tahap['tgl']));
    $ref_tahapan[$bln_batas] = $key;
}
echo '<table>
        <tr>
            <th rowspan="2">Bulan</th>
```

```

        <th colspan="2">Deadline</th>
    </tr>
    <tr>
        <th>Tahapan</th>
        <th>Tanggal</th>
    </tr>

    ';

$jml_bln = count($bulan);
for ($i = 1; $i <= $jml_bln; $i++)
{
    echo '<tr>
            <td> ' . strtoupper($bulan[$i]) . ' </td>';

            if (key_exists($i, $ref_tahapan))
            {
echo '<td>' . $tahapan[$ref_tahapan[$i]]['tahap'] . '</td>
            <td>' . $tahapan[$ref_tahapan[$i]]['tgl'] . '</td>';
            }
            else
            {
echo '<td>-</td>
            <td>-</td>';
            }

            echo '</tr>';
}
echo '</table>';
?>
```

pada kode diatas line 26 s.d 30 kita membuat variabel baru bernama \$ref_tahapan yang berbentuk array dengan key bulan dan value index dari array tahapan, contoh \$ref_tahapan[1] = 0 yang berarti bulan 1 merujuk ke \$tahapan[0].

Kode diatas tidak terlalu kompleks dan masih wajar untuk digunakan walaupun pengguna kode membuat kita harus menambah tenaga untuk memahami variabel baru \$ref_tahapan

Ketika di tes, kode diatas membutuhkan waktu eksekusi 0.0011389255523682 detik sedangkan sebelumnya 0.014470100402832 detik, tidak terasa perbedaannya oleh karena itu kita dapat menggunakan kedua cara diatas, namun jika kode yang ditulis kompleks, cara kedua bisa dipertimbangkan untuk digunakan.

2Hindari eksekusi perintah SQL di dalam loop

Sebisa mungkin jangan pernah melakukan eksekusi kode SQL di dalam loop, dengan eksekusi yang berulang ulang maka akan memberatkan server database yang pada akhirnya akan menurunkan performa aplikasi anda.

Contoh dibawah ini pengulangan eksekusi MySQL di dalam loop (variabel \$bulan menggunakan contoh sebelumnya):

```
$jml_bln = count($bulan);
for ($i = 1; $i <= $jml_bln, $i++) {
    $bln = substr('0' . $i, -2);
    $sql = 'SELECT jml_byr FROM penjualan WHERE MONTH(tgl_byr) = $bln';
    $stmt = $pdo->prepare($sql);
    $stmt->execute();
    $penjualan[$bln] = $stmt->fetchAll(PDO::FETCH_ASSOC);
}
```

Contoh diatas akan mengeksekusi perintah SQL sebanyak 12 kali yang tentu saja akan memberatkan, terlebih lagi jika datanya sangat besar.

3Berhati – hati dalam penulisan nested loop

Terkadang kita menuliskan banyak loop di dalam loop, untuk kehati-hatian, gunakan variabel yang mencerminkan kondisi yang ada, tidak sekedar \$i, mengingat nilai variabel akan berubah jika kita mendefinisikan dengan nama yang sama (baik sengaja maupun tidak), contoh berikut akan menghasilkan

```

*
**
***
****

<?php
for ($row = 1; $row <= 5; $row++)
{
    for ($col = 1; $col <= $row; $col++)
    {
        echo '*' . '<br/>';
    }
}
?>

```

KASUS

Cara Membuat Looping For di dalam Tabel

Script HTML di bawah ini digunakan untuk menampilkan bentuk table yang akan kita buat nantinya, dari table sederhana inilah kita akan membuat sebuah looping for.

```

<!DOCTYPE html>
<html>
<head>
    <title>Cara Membuat Looping For di dalam Tabel</title>
</head>
<body>
    <h2>Cara Membuat Looping For di dalam Tabel</h2>
    <form>
        <table border="1" cellspacing="0">
            <tr>
<th>NO</th>
<th>BUAH</th>
<th>SAYUR</th>
            </tr>
            <tr>
<td></td>
<td></td>
<td></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

Script CSS untuk membuat tampilan table di atas lebih menarik.

```

<style>
    table{width:300px; text-align:center; margin:auto;}
    table th { background-color: #95a5a6; }
    h2 {text-align:center; font-style:italic; font-weight:bold; }
</style>

```

Nah pada contoh Ini kita akan memanfaatkan sebuah data di dalam table, langsung saja lihat scriptnya di bawah ini.

```

<form>
    <table border="1" cellspacing="0">
        <tr>
<th>NO</th>
<th>BUAH</th>
<th>SAYUR</th>

```

```

</tr>

<?php for ($no = 1, $i=10, $a=100; $i<=100, $a<=1000 ; 
    $i+=10, $a+=100) { ?>

<tr>
<td> <?php echo $no; ?></td>
<td><?php echo $i; ?></td>
<td><?php echo $a; ?></td>
</tr>

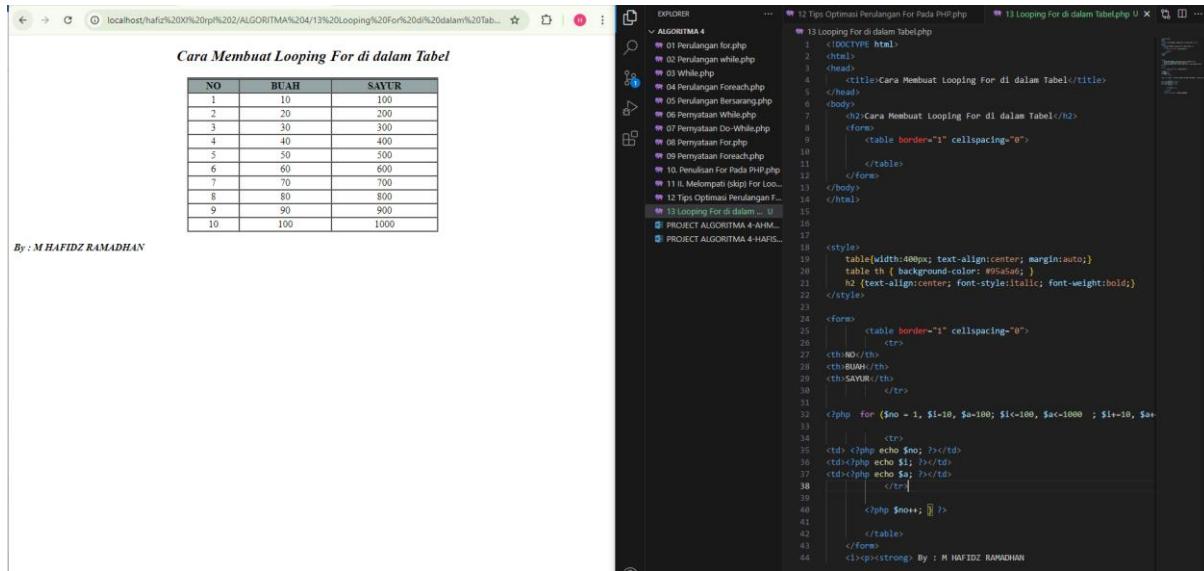
<?php $no++; } ?>

</table>
</form>

```

Cara Membuat Looping For di dalam Tabel

| NO | BUAH | SAYUR |
|----|------|-------|
| 1 | 10 | 100 |
| 2 | 20 | 200 |
| 3 | 30 | 300 |
| 4 | 40 | 400 |
| 5 | 50 | 500 |
| 6 | 60 | 600 |
| 7 | 70 | 700 |
| 8 | 80 | 800 |
| 9 | 90 | 900 |
| 10 | 100 | 1000 |



The screenshot shows a code editor with two panes. The left pane displays a table titled "Cara Membuat Looping For di dalam Tabel" with the following data:

| NO | BUAH | SAYUR |
|----|------|-------|
| 1 | 10 | 100 |
| 2 | 20 | 200 |
| 3 | 30 | 300 |
| 4 | 40 | 400 |
| 5 | 50 | 500 |
| 6 | 60 | 600 |
| 7 | 70 | 700 |
| 8 | 80 | 800 |
| 9 | 90 | 900 |
| 10 | 100 | 1000 |

The right pane shows the corresponding PHP code:

```

<!DOCTYPE html>
<html>
<head>
<title>Cara Membuat Looping For di dalam Tabel</title>
</head>
<body>
<h2>Cara Membuat Looping For di dalam Tabel:</h2>
<form>
<table border="1" cellspacing="0">
<tr>
<td><?php echo $no; ?></td>
<td><?php echo $i; ?></td>
<td><?php echo $a; ?></td>
</tr>
<?php for ($no = 1, $i=10, $a=100; $i<=100, $a<=1000 ; $i+=10, $a+=100) { ?>
<tr>
<td><?php echo $no; ?></td>
<td><?php echo $i; ?></td>
<td><?php echo $a; ?></td>
</tr>
<?php $no++; ?>
</table>
</form>
<!><p><strong> By : M HAFIDZ RAMADHAN</strong></p>

```

Cara Membuat Tabel dengan perulangan

```

<?php

echo "<table border=1>";
for($i=1; $i <=3; $i++) {
    echo "<tr>";
    for($j=1; $j<=5; $j++) {
        echo "<td>";
        echo $i.$j;
        echo "</td>";
    }
    echo "</tr>";
}
echo "</table>";
?>

```

The screenshot shows a web browser window with the URL localhost/hafiz%20X%20rp%202/ALGORITMA%204/14%20Cara%20Membuat%20Tabel%20dengan.... The page displays a 3x5 grid of numbers:

| | | | | |
|----|----|----|----|----|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |

Below the grid, a copyright notice reads "By M HAFIDZ RAMADHAN NST".

The right side of the screen shows the Visual Studio Code interface with the Explorer sidebar open, displaying a folder named "ALGORITMA 4" containing 16 PHP files. The "14 Cara Membuat Tabel dengan perulangan.php" file is selected and its content is shown in the main editor area:

```
<?php  
echo "<table border='1'>";  
for($i=1;$i<3;$i++){  
    echo "<tr>";  
    for($j=1;$j<5;$j++){  
        echo "<td>";  
        echo $i.$j;  
        echo "</td>";  
    }  
    echo "</tr>";  
}  
echo "</table>";  
print_r("<p><strong>i></strong>By M HAFIDZ RAMADHAN NST!</p>");  
?>
```

Cara Kerja:

Kali ini kita akan membuat table dengan ukuran 3 baris 5 kolom. Di perulangan pertama kita akan membuat kotak baris dengan <tr>. Diperulangan kedua akan kita pecah debanyak 5 kolom dengan <td>. Di dalam kotak akan berisi nilai dengan skrip \$i.\$j sesuai dengan urutan matrik kotak.

PENGAYAAN FOR LOOP

Pelajari cara menggunakan perulangan for di PHP dengan penjelasan dan contoh yang mudah dipahami, cocok untuk pemula dalam pemrograman web.

For loop adalah struktur kontrol yang sangat berguna dalam PHP yang memungkinkan kamu untuk mengeksekusi blok kode berulang kali dengan jumlah tertentu. Instruksi perulangan ini digunakan saat kamu mengetahui sebelumnya berapa kali blok kode harus dijalankan.

Penggunaan For Loop

Perulangan for di PHP memiliki sintaks sebagai berikut:

```
for (inisialisasi; kondisi; increment) {  
    // blok kode yang akan diulang  
}
```

Inisialisasi: Menetapkan nilai awal untuk variabel loop.

Kondisi: Ekspresi yang dievaluasi sebelum setiap iterasi. Jika kondisi bernilai TRUE, blok kode di dalam loop akan dijalankan. Jika FALSE, loop akan berakhir.

Increment: Digunakan untuk meningkatkan (atau kadang-kadang mengurangi) nilai variabel loop.

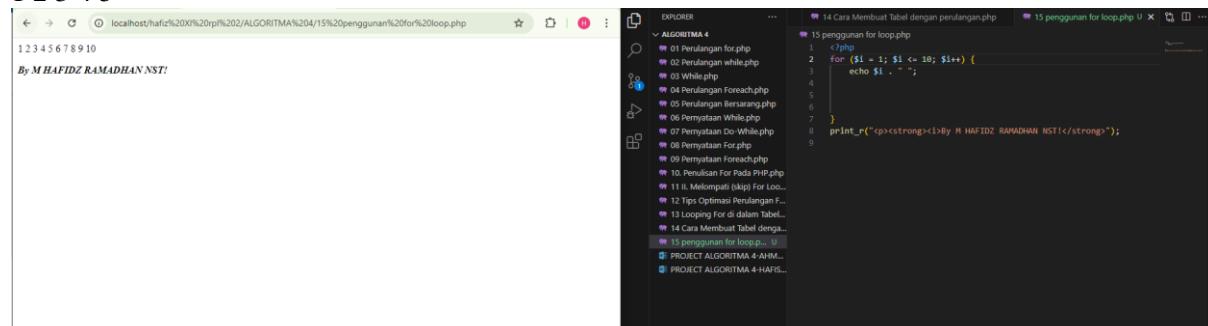
Contoh For Loop

Misalkan kita ingin menampilkan angka dari 1 sampai 5. Kita bisa menggunakan for loop seperti ini:

```
for ($i = 1; $i <= 5; $i++) {  
    echo $i . " ";  
}
```

Output:

1 2 3 4 5



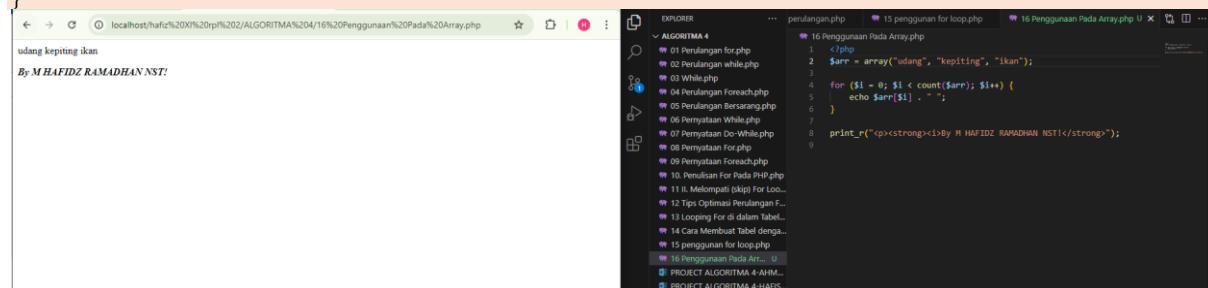
Dalam contoh di atas, \$i = 1 adalah inisialisasi, \$i <= 5 adalah kondisi yang menyatakan loop akan terus berjalan selama \$i kurang dari atau sama dengan 5, dan \$i++ adalah increment yang akan menambahkan nilai \$i setiap kali loop dijalankan.

Penggunaan Pada Array

For loop juga bisa digunakan untuk mengiterasi elemen-elemen dalam sebuah array. Sebagai contoh:

```
$arr = array("Apple", "Banana", "Cherry");
```

```
for ($i = 0; $i < count($arr); $i++) {  
    echo $arr[$i] . " ";  
}
```



Output:

Apple Banana Cherry

Di sini, count(\$arr) digunakan untuk mendapatkan jumlah elemen dalam array sehingga kita bisa mengulang hingga elemen terakhir.

For Loop dengan Step yang Berbeda

Kamu juga bisa mengatur increment untuk langkah yang berbeda-beda. Misalnya, jika kamu ingin menampilkan setiap angka genap antara 1 sampai 10:

```
for ($i = 2; $i <= 10; $i += 2) {
    echo $i . " ";
}
```

Output:

2 4 6 8 10

The screenshot shows a code editor with two tabs open. The left tab contains the output of the PHP code: "2 4 6 8 10". The right tab shows the source code for a file named "17 For Loop dengan Step yang Berbeda.php". The code is as follows:

```
<?php
for ($i = 2; $i <= 12; $i += 2) {
    echo $i . " ";
}
print_r("<p><strong>i</strong></p>");
```

Perulangan for di PHP sangat fleksibel dan dapat digunakan untuk berbagai kasus. Dengan memahami konsep dan sintaks dasarnya, kamu bisa membuat kode yang efisien dan mudah dibaca. Selamat mencoba!

WHILE LOOP

Pelajari cara menggunakan while loop dalam PHP dengan penjelasan dan contoh kode yang mudah dipahami.

While loop adalah struktur kendali yang umum digunakan dalam pemrograman untuk mengeksekusi blok kode berulang kali selama suatu kondisi bernilai true. Dalam PHP, while loop memiliki sintaks yang sederhana dan mudah diimplementasikan. Artikel ini akan memandu kamu untuk memahami dan menggunakan while loop dalam PHP dengan beberapa contoh praktis.

While Loop dalam PHP Sebuah while loop akan terus menjalankan sebuah blok kode selama kondisi yang ditentukan masih benar. Struktur dasar dari while loop dalam PHP ditunjukkan seperti berikut:

```
while (kondisi) {
    // blok kode yang akan diulang
}
```

Kunci dari loop ini adalah kondisi; jika kondisi berhenti menjadi true, maka loop akan berhenti. Apabila kondisi tidak pernah berubah menjadi false, maka kamu akan mendapatkan apa yang dikenal sebagai loop tak terbatas, yang akan terus berjalan selamanya.

Contoh Penggunaan while Loop

Berikut adalah contoh sederhana penggunaan while loop dalam PHP:

```
<?php
$i = 1;

while ($i <= 5) {
    echo "Nomor: $i <br>";
    $i++;
}
?>
```

The screenshot shows a code editor with three tabs open. The left tab contains the output of the PHP code: "Nomor: 1 Nomor: 2 Nomor: 3 Nomor: 4 Nomor: 5". The middle tab shows the source code for a file named "18 WHILE LOOP.php". The code is as follows:

```
<?php
// blok kode yang akan diulang
$i = 1;
while ($i <= 5) {
    echo "Nomor: $i <br>";
    $i++;
}
?>
```

Dalam contoh ini, variabel \$i diinisialisasi dengan nilai 1. Kemudian, while loop memeriksa apakah \$i kurang dari atau sama dengan 5. Jika true, blok kode di dalam loop akan dijalankan, mencetak nomor saat ini serta menambahkan nilai \$i dengan 1. Proses ini berulang sampai kondisi menjadi false, yaitu ketika \$i lebih besar dari 5.

Menghindari Loop Tak Terbatas

Untuk menghindari terjadinya loop tak terbatas, pastikan bahwa kondisi loop pada suatu titik akan bernilai false. Berikut adalah beberapa cara untuk mencegah loop tak terbatas:

Pastikan bahwa ada perubahan nilai dalam variabel yang menjadi kondisi, seperti menggunakan increment (`$i++`)

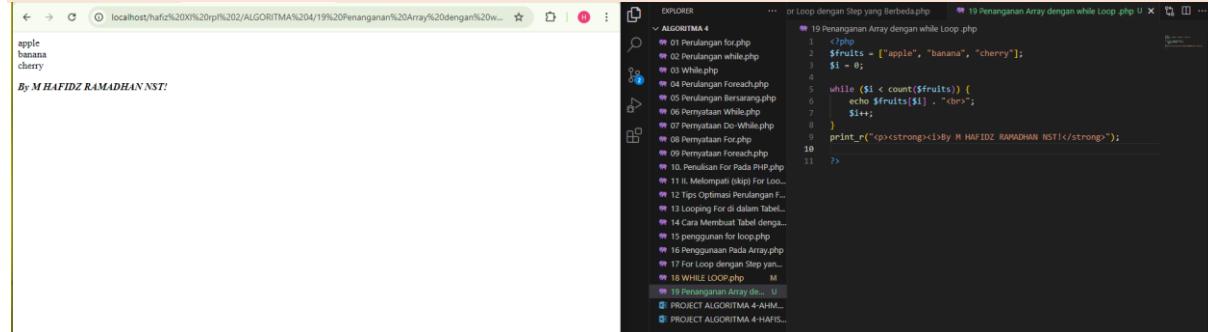
Gunakan kondisi yang pasti akan berubah seiring dengan iterasi loop, misalnya mengecek nilai dari sebuah variabel yang terus berkurang atau bertambah

Pertimbangkan penggunaan break untuk keluar dari loop jika ada kondisi tertentu yang terpenuhi sebelum kondisi utama loop menjadi false

Penanganan Array dengan while Loop

While loop juga bisa digunakan untuk menangani elemen dalam array. Contoh:

```
<?php  
$fruits = ["apple", "banana", "cherry"];  
$i = 0;  
  
while ($i < count($fruits)) {  
    echo $fruits[$i] . "<br>";  
    $i++;  
}  
?>
```



```
localhost/hafiz%20rp%202/ALGORITMA%204/19%20Penanganan%20Array%20dengan%20w...  
apple  
banana  
cherry  
By M HAFIDZ RAMADHAN NSTI:
```

The screenshot shows a browser window displaying the output of a PHP script. The output consists of three lines of fruit names: 'apple', 'banana', and 'cherry', each followed by a new line character. Below the output, there is a copyright notice: 'By M HAFIDZ RAMADHAN NSTI'. To the right of the browser, there is a code editor window titled '19 Penanganan Array dengan while Loop.php'. The code in the editor matches the one provided above, including the file path at the top.

Dalam contoh di atas, while loop digunakan untuk menampilkan tiap elemen dalam array `$fruits`. Loop akan berjalan selama nilai `$i` lebih kecil dari jumlah elemen dalam array.

While loop merupakan alat yang penting dan sering digunakan dalam PHP. Dengan memahami cara kerjanya dan bagaimana mengimplementasikannya dengan benar, kamu dapat menghindari masalah umum seperti loop tak terbatas dan menggunakan loop ini untuk mempermudah proses iterasi dalam programmu.

DO WHILE LOOP

Pelajari bagaimana menggunakan do while loop dalam PHP dengan tutorial langkah demi langkah yang mudah dipahami.

Looping adalah konsep penting dalam pemrograman, dan PHP menawarkan beberapa cara untuk melakukan loop. Salah satu struktur loop yang sering digunakan adalah do while. Loop do while di PHP sangat berguna ketika kamu ingin memastikan bahwa blok kode dijalankan setidaknya sekali, bahkan jika kondisi loop tidak terpenuhi.

Penggunaan Dasar do while

Loop do while menjalankan blok kode sekali, kemudian mengecek kondisi. Jika kondisi bernilai true, maka blok kode akan dijalankan lagi, dan proses ini terus berlanjut sampai kondisi bernilai false.

Sintaks umum dari do while loop adalah sebagai berikut:

```
do {  
    // blok kode untuk dijalankan  
} while (kondisi);
```

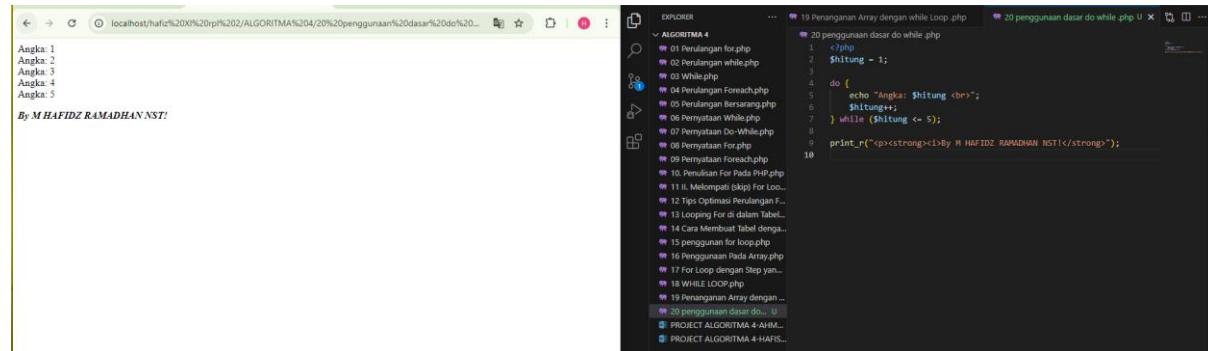
Elemen kunci dari loop ini adalah kondisi yang diperiksa setelah eksekusi blok kode. Ini berarti bahwa blok kode dalam do pasti akan dijalankan setidaknya satu kali, tidak peduli apa kondisi awalnya.

Contoh Sederhana do while Loop

Berikut adalah contoh sederhana penggunaan do while dalam PHP:

```
$hitung = 1;
```

```
do {
    echo "Angka: $hitung <br>";
    $hitung++;
} while ($hitung <= 5);
```



```
<!DOCTYPE html>
<html>
<head>
<title>localhost/nafiz%20X%20rp%202/ALGORITMA%20/20%20penggunaan%20dasar%20do%20...</title>
</head>
<body>
    Angka: 1
    Angka: 2
    Angka: 3
    Angka: 4
    Angka: 5
</body>
</html>
```

Kode di atas akan mencetak angka 1 sampai 5. Walaupun \$hitung dimulai dari 1 dan kondisi adalah \$hitung <= 5, loop tetap dijalankan karena pemeriksaan kondisi terjadi setelah eksekusi pertama.

Perbedaan dengan while Loop

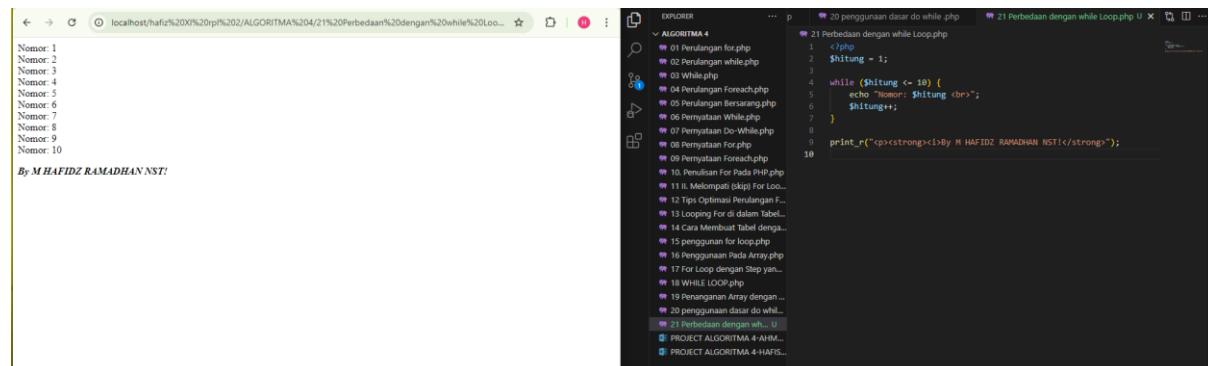
Kamu mungkin bertanya-tanya apa perbedaan antara do while dan while. Loop while mengecek kondisi sebelum menjalankan blok kode, jadi jika kondisi awalnya false, blok kode tidak akan pernah dijalankan. Sementara pada do while, blok kode dijamin untuk dijalankan setidaknya sekali.

Contoh while Loop

Untuk memahami perbedaan tersebut, mari kita bandingkan dengan loop while:

```
$hitung = 1;
```

```
while ($hitung <= 5) {
    echo "Angka: $hitung <br>";
    $hitung++;
}
```



```
<!DOCTYPE html>
<html>
<head>
<title>localhost/nafiz%20X%20rp%202/ALGORITMA%20/21%20Perbedaan%20dengan%20while%20Lo...</title>
</head>
<body>
    Nomor: 1
    Nomor: 2
    Nomor: 3
    Nomor: 4
    Nomor: 5
    Nomor: 6
    Nomor: 7
    Nomor: 8
    Nomor: 9
    Nomor: 10
</body>
</html>
```

Hasil dari kode di atas akan sama dengan contoh do while. Namun, jika kita mengubah variabel \$hitung menjadi lebih besar dari 5 sejak awal, loop while tidak akan mengeksekusi blok kode sama sekali, sementara do while akan mengeksekusinya sekali sebelum berhenti.

Memahami Kondisi

Kondisi dalam loop do while harus diperhatikan secara seksama karena bisa menyebabkan loop yang tidak pernah berakhir jika kondisi selalu bernilai true. Pastikan kondisi yang kamu gunakan pada do while akan menjadi false pada titik tertentu untuk menghentikan loop.

Contoh Kondisi Loop yang Benar

Pastikan kamu mengupdate variabel yang digunakan dalam kondisi, seperti di contoh berikut:

```
$hitung = 1;
```

```

do {
    echo "Angka: $hitung <br>";
    $hitung++;
} while ($hitung <= 5);

```

```

Angka: 1
Angka: 2
Angka: 3
Angka: 4
Angka: 5
By M HAFIDZ RAMADHAN NST!

```

```

EXPLORER
  ✓ ALGORITMA 4
    01 Perulangan for.php
    02 Perulangan while.php
    03 While.php
    04 Perulangan Foreach.php
    05 Perulangan Bersarang.php
    06 Perintahan While.php
    07 Perintahan Do-While.php
    08 Perintahan For.php
    09 Perintahan Foreach.php
    10. Penulisan For Pada PHP.php
    11. Melompati (skip) For Loop...
    12. Tips Optimasi Perulangan F...
    13. Looping For di dalam Tabel...
    14. Cara Membuat Tabel dengan...
    15. penggunaan for loop.php
    16. Penggunaan Pada Array.php
    17. For Loop dengan Step yang...
    18. WHILE LOOP.php
    19. Pengulangan Array dengan ...
    20. penggunaan dasar do whil...
    21. Perbedaan dengan while L...
    22. memahami kondisi loop... U
    PROJECT ALGORITMA 4-AHM...
    PROJECT ALGORITMA 4-HAF...

```

```

1 <?php
2 $hitung = 0;
3
4 do {
5     echo "Angka: $hitung <br>";
6     $hitung++;
7 } while ($hitung <= 5);
8
9 print_r("<p><strong><i>By M HAFIDZ RAMADHAN NST!</i></strong>"); 10

```

Dalam kasus ini, \$hitung akan terus bertambah hingga tidak lagi memenuhi kondisi \$hitung \leq 5 dan loop akan berhenti.

Menggunakan loop do while bisa sangat praktis dalam berbagai situasi, terutama ketika kamu membutuhkan setidaknya satu eksekusi dari blok kode, tanpa memandang kondisi awal. Pahami dengan baik cara kerjanya dan implementasikan dengan benar dalam kode PHP kamu.

FOREACH LOOP

Pelajari cara menggunakan foreach loop dalam PHP untuk mengulangi elemen-elemen dalam array dengan tutorial langkah demi langkah ini.

Foreach loop adalah salah satu struktur kontrol yang paling banyak digunakan dalam PHP untuk mengulangi elemen-elemen dalam array. Melalui foreach, kamu bisa mengakses setiap nilai dalam array tanpa perlu mengetahui jumlah elemen yang ada di dalamnya.

Penggunaan Foreach Loop

Foreach loop digunakan untuk menjalankan blok kode untuk setiap elemen dalam array. Ini menjadikannya pilihan yang efisien dan sederhana ketika berurusan dengan array yang berisi kumpulan data.

Sintaks dasar dari foreach loop adalah sebagai berikut:

```

foreach ($array as $value) {
    // Blok kode yang akan dijalankan untuk setiap elemen
}

```

```

apel
durian
ceri
By M HAFIDZ RAMADHAN NST!

```

```

EXPLORER
  ✓ ALGORITMA 4
    01 Perulangan for.php
    02 Perulangan while.php
    03 While.php
    04 Perulangan Foreach.php
    05 Perulangan Bersarang.php
    06 Perintahan While.php
    07 Perintahan Do-While.php
    08 Perintahan For.php
    09 Perintahan Foreach.php
    10. Penulisan For Pada PHP.php
    11. Melompati (skip) For Loop...
    12. Tips Optimasi Perulangan F...
    13. Looping For di dalam Tabel...
    14. Cara Membuat Tabel dengan...
    15. penggunaan for loop.php
    16. Penggunaan Pada Array.php
    17. For Loop dengan Step yang...
    18. WHILE LOOP.php
    19. Pengulangan Array dengan ...
    20. penggunaan dasar do whil...
    21. Perbedaan dengan while L...
    22. memahami kondisi loop... U
    23. Penggunaan Foreach Loop.php
    PROJECT ALGORITMA 4-AHM...
    PROJECT ALGORITMA 4-HAF...

```

```

1 <?php
2 $buah = array("apel", "durian", "ceri");
3
4 foreach ($buah as $val) {
5     echo $val . "<br>";
6 }
7
8 print_r("<p><strong><i>By M HAFIDZ RAMADHAN NST!</i></strong>"); 9

```

Di dalam foreach loop, variabel \$array adalah array yang ingin kamu ulangi, dan \$value adalah nama variabel yang akan memegang nilai dari setiap elemen array saat loop sedang berlangsung.

Foreach Loop dengan Key

Kamu juga bisa mendapatkan kunci (key) dari setiap elemen dalam array selama proses iterasi. Ini sangat berguna ketika kamu perlu menggunakan key tersebut dalam operasi.

Berikut adalah sintaks menggunakan key dalam foreach:

```
foreach ($array as $key => $value) {  
    // Blok kode dengan $key dan $value  
}
```

Contoh Sederhana

Berikut adalah contoh penggunaan foreach loop untuk menampilkan setiap nilai dari array:

```
$buah = array("apel", "pisang", "ceri");
```

```
foreach ($buah as $val) {  
    echo $val . "<br>";  
}
```

Output yang akan dihasilkan dari kode di atas adalah:

```
apel  
pisang  
ceri
```

Contoh dengan Key

Mari kita menggunakan foreach dengan mendapatkan key dari array:

```
$harga = array("apel" => 10000, "pisang" => 5000, "ceri" => 15000);
```

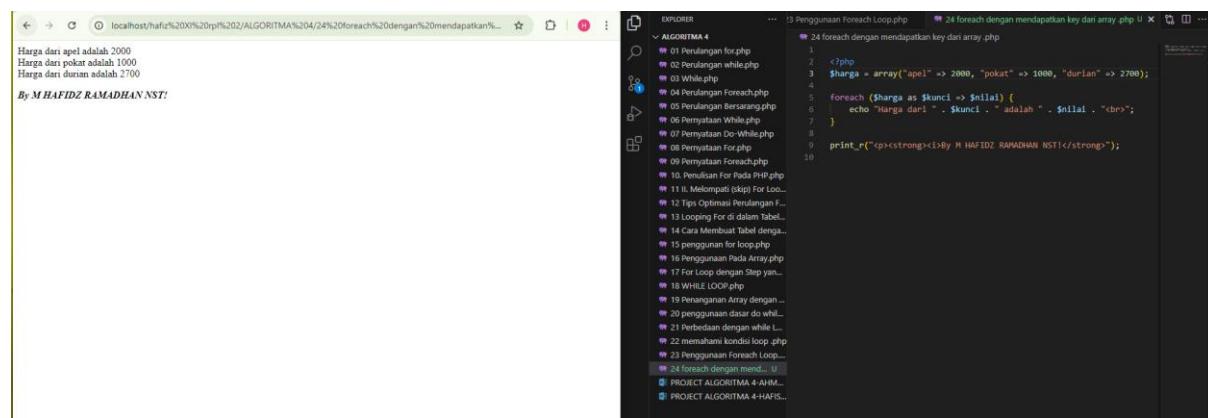
```
foreach ($harga as $kunci => $nilai) {  
    echo "Harga dari " . $kunci . " adalah " . $nilai . "<br>";  
}
```

Contoh di atas akan mengeluarkan output berikut:

Harga dari apel adalah 10000

Harga dari pisang adalah 5000

Harga dari ceri adalah 15000



Penanganan Array Multidimensi

Kadang kita perlu bekerja dengan array yang lebih kompleks, seperti array multidimensi. Dengan mengkombinasikan loop, kamu bisa mengakses elemen-elemen yang terletak lebih dalam.

Contoh penggunaan foreach pada array multidimensi:

```
$matriks = array(  
    array(1, 2, 3),  
    array(4, 5, 6),  
    array(7, 8, 9)  
);
```

```
foreach ($matriks as $baris) {  
    foreach ($baris as $nilai) {  
        echo $nilai . " ";  
    }  
    echo "<br>";  
}
```

Output:

1
2
3
4
5
6
7
8
9



localhost/hafiz%20X%20pi%20ALGORITMA%204%25Penanganan%20Array%20Multidimensi...
1 2 3
4 5
7 8 9
10 11 12

By M HAFIZ RAMADHAN NST:

```
EXPLORER
+ ALGORITMA 4
  ▾ 01 Perulangan for.php
  ▾ 02 Perulangan while.php
  ▾ 03 While.php
  ▾ 04 Perulangan ForEach.php
  ▾ 05 Perulangan Bersarang.php
  ▾ 06 Pengulangan Do-While.php
  ▾ 07 Pengulangan For.php
  ▾ 08 Pengulangan ForEach.php
  ▾ 09 Penanganan Foreach.php
  ▾ 10 Revulisan For Pada PHP.php
  ▾ 11 Melompati (skip) For Lo...
  ▾ 12 Tips Optimasi Perulangan F...
  ▾ 13 Looping For di Dalam Tabel...
  ▾ 14 Cara Membuat Tabel dengan ...
  ▾ 15 penggunaan for loop.php
  ▾ 16 Penggunaan Pada Array.php
  ▾ 17 For Loop dengan Step yan...
  ▾ 18 WHILE LOOP.php
  ▾ 19 Penanganan Array dengan ...
  ▾ 20 penggunaan dasar do whil...
  ▾ 21 Perbedaan dengan while ...
  ▾ 22 memahami kondisi loop.php
  ▾ 23 Penggunaan Foresch Loop...
  ▾ 24 foreach dengan mendapt...
  ▾ 25 Penanganan Array Mu...
  ▾ PROJECT ALGORITMA 4-AHM...
  ▾ PROJECT ALGORITMA 4-HAFIS...
```

1 2 3 4 5 6 7 8 9 10 11 12

17 echo dengan mendapatkan key dari array.php
25 Penanganan Array Multidimensi.php
25 Penanganan Array Multidimensi.php

```
1 $striks = array(
2     array(1, 2, 3),
3     array(4, 5, 6),
4     array(7, 8, 9),
5     array(10, 11, 12)
6 );
7
8
9 foreach ($striks as $baris) {
10     foreach ($baris as $nilai) {
11         echo $nilai . " ";
12     }
13 }
14 echo "<br>";
15
16 print_r("<p><strong><i>By M HAFIZ RAMADHAN NST</i></strong>");
```

Itulah dasar-dasar menggunakan foreach loop di PHP. Dengan memahami konsep ini, kamu akan lebih mudah menangani data dalam array dan mengembangkan aplikasi yang lebih dinamis.

BREAK DAN CONTINUE

Belajar penggunaan perintah break dan continue dalam pengulangan di PHP untuk mengontrol loop secara efektif.

Mengontrol Loop dengan break dan continue di PHP

Dalam pemrograman, terutama saat bekerja dengan loop, terkadang kita perlu menghentikan pengulangan atau melompati iterasi tertentu. PHP menyediakan dua perintah penting, break dan continue, untuk mengatasi hal tersebut.

Penggunaan break

Perintah break digunakan untuk menghentikan eksekusi loop sepenuhnya. Ketika PHP menemui break, ia akan segera keluar dari struktur loop, baik itu for, foreach, while, atau do-while.

Contoh:

```
for ($i = 0; $i < 10; $i++) {  
    if ($i === 5) {  
        break;  
    }  
    echo $i . " ";  
}  
// Output: 0 1 2 3 4
```

Dalam contoh di atas, loop akan berhenti ketika \$i bernilai 5.

Dalam contoh di atas, loop akan berhenti ketika nilai \$i = 10.

The screenshot shows a browser window with the URL `localhost/hafiz%20X%20pf%202/ALGORITMA%204/20penggunaan%20break%20.php`. The page content includes a navigation bar with links from 0 to 9, followed by the text "By M HAFIZ RAMADHAN NSTI:" and a large number of numbered links from 01 to 25. To the right of the browser is a terminal window titled "ray.php" showing the following PHP code:

```
explorer ... ray.php 25 Penggunaan Array Multidimensi.php 26 penggunaan break.php
ALGORITMA 4
01 Perulangan for.php
02 Perulangan while.php
03 While.php
04 Perulangan Foreach.php
05 Perulangan Bersarang.php
06 Pernyataan While.php
07 Pernyataan Do-While.php
08 Pernyataan For.php
09 Pernyataan Foreach.php
10. Penulisan For Pada PHP.php
11. Melompati (skip) For Lo...
12. Tips Optimasi Perulangan F...
13 Looping For di dalam tabel...
14 Cara Membuat Label dengan...
15 penggunaan for loop.php
16 Penggunaan Rada Array.php
17 For Loop dengan Step yah...
18 WHILE LOOP.php
19 Penggunaan break dengan ...
20 penggunaan dasar do whil...
21 Penerapan dengan while L...
22 Penerapan kondisi loop .ph...
23 Penggunaan ForEach Loop...
24 Foreach dengan mendekati...
25 Penggunaan Array Multidi...
26 penggunaan break.php

1 <?php
2     for ($i = 0; $i < 20; $i++) {
3         if ($i == 10) {
4             break;
5         }
6         echo $i . " ";
7     }
8
9     print_r("<p><strong>Isi By M HAFIZ RAMADHAN NSTI</strong></p>");
10
```

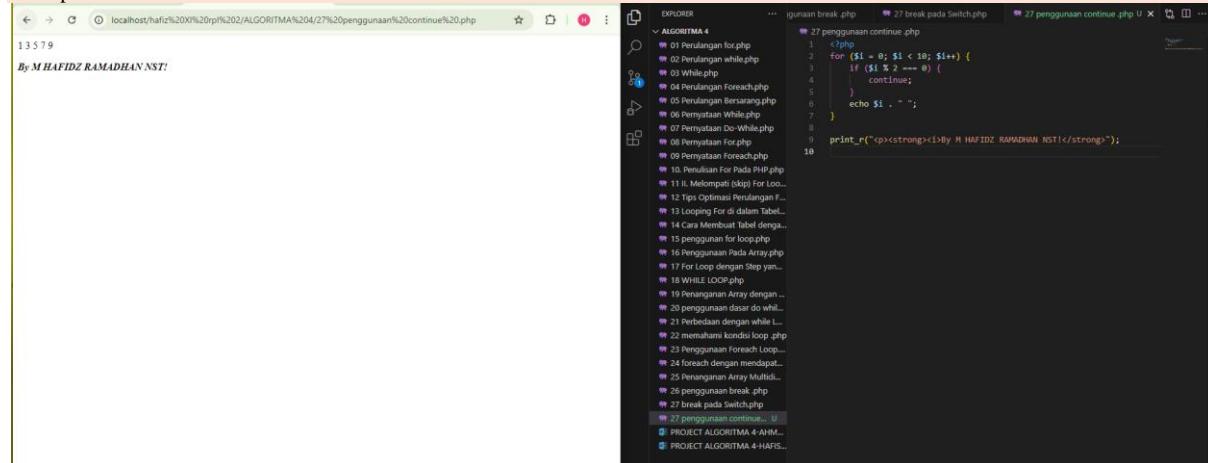
Penggunaan continue

Berbeda dengan break, perintah continue tidak menghentikan loop sepenuhnya. Sebaliknya, continue akan menghentikan iterasi saat ini dan melanjutkan ke iterasi berikutnya dari loop.

Contoh:

```
for ($i = 0; $i < 10; $i++) {  
    if ($i % 2 === 0) {  
        continue;  
    }  
    echo $i . " ";  
}
```

// Output: 1 3 5 7 9



```
<?php  
for ($i = 0; $i < 10; $i++) {  
    if ($i % 2 === 0) {  
        continue;  
    }  
    echo $i . " ";  
}  
print_r("<p><strong><i>By M HAFIDZ RAMADHAN NSTI</i></strong></p>");
```

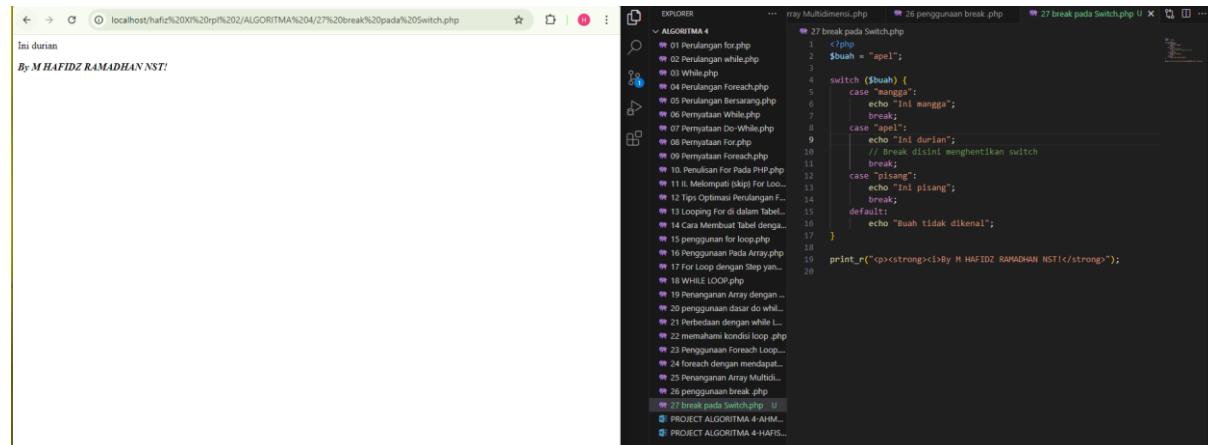
Pada contoh di atas, continue digunakan untuk melewatkkan iterasi loop bila \$i adalah bilangan genap.

break pada Switch

break juga sering digunakan dalam switch untuk menghentikan eksekusi kode setelah sebuah case telah dipenuhi.

Contoh:

```
$buah = "apel";  
  
switch ($buah) {  
    case "mangga":  
        echo "Ini mangga";  
        break;  
    case "apel":  
        echo "Ini apel";  
        // Break disini menghentikan switch  
        break;  
    case "pisang":  
        echo "Ini pisang";  
        break;  
    default:  
        echo "Buah tidak dikenal";  
}
```



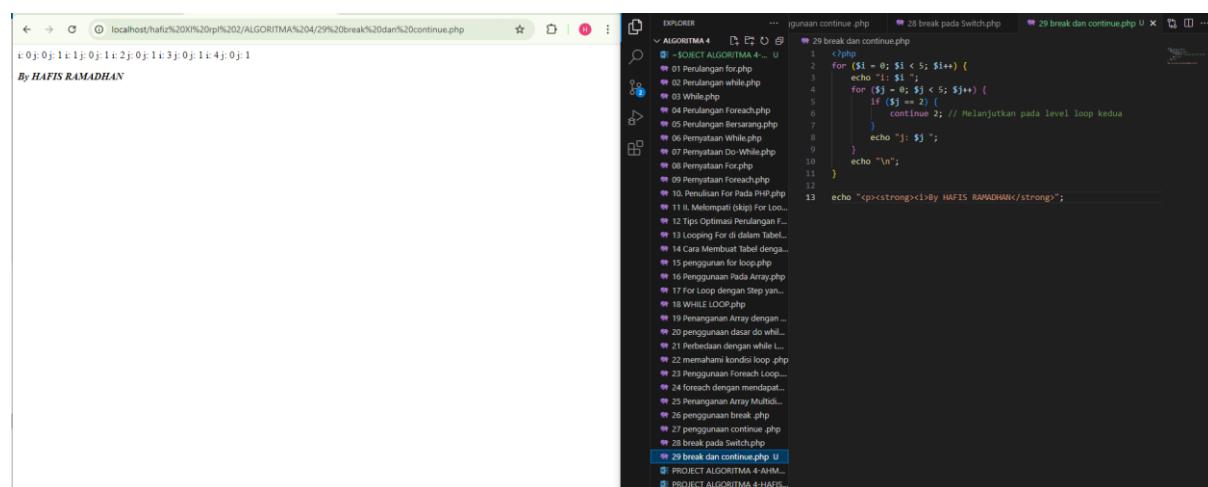
```
<?php  
$buah = "apel";  
  
switch ($buah) {  
    case "mangga":  
        echo "Ini mangga";  
        break;  
    case "apel":  
        echo "Ini durian";  
        // Break disini menghentikan switch  
        break;  
    case "pisang":  
        echo "Ini pisang";  
        break;  
    default:  
        echo "Buah tidak dikenal";  
}  
print_r("<p><strong><i>By M HAFIDZ RAMADHAN NSTI</i></strong></p>");
```

Bersarang Dengan break dan continue

Ketika bekerja dengan loop bersarang, break dan continue juga dapat mengontrol loop di luar loop saat ini. Namun, kamu perlu memberikan level sebagai argumen kepada mereka.

Contoh dengan continue:

```
for ($i = 0; $i < 5; $i++) {  
    echo "i: $i ";  
    for ($j = 0; $j < 5; $j++) {  
        if ($j == 2) {  
            continue 2; // Melanjutkan pada level loop kedua  
        }  
        echo "j: $j ";  
    }  
    echo "\n";  
}
```



Dalam contoh di atas, ketika \$j sama dengan 2, continue akan melewati sisa kode di loop luar dan langsung melanjutkan ke iterasi selanjutnya dari loop tersebut.

Perlu diingat, untuk pemula, kesalahan dalam penggunaan break dan continue dapat membingungkan dan seringkali menyebabkan bug. Oleh karena itu, gunakanlah perintah ini dengan bijak dan selalu tes loop kamu untuk memastikan mereka bekerja sesuai yang diharapkan.



MATERI PERTEMUAN 9

Array Dalam Pemrograman PHP



TAHUKAH KAMU...?

Pada kesempatan ini, kita akan membahas:

- Apa itu Array?
- Cara membuat Array di PHP dan Mengisinya
- Cara menampilkan nilai Array
- Cara Menghapus isi Array
- Cara Menambah isi Array
- Array Assosiatif
- Array Multidimensi

Bayangkan sekarang kita sedang membuat aplikasi web, lalu ingin menampilkan daftar nama-nama produk.

Bisa saja kita buat seperti ini:

```
<?php  
$produk1 = "Modem";  
$produk2 = "Hardisk";  
$produk3 = "Flashdisk";  
  
echo "$produk1<br>";  
echo "$produk2<br>";  
echo "$produk3<br>";
```

Apakah boleh seperti ini?

Boleh-boleh saja. Tapi kurang efektif. Kenapa? Bagaimana kalau ada 100 produk, apakah kita akan membut variabel sebanyak 100 dan melakukan **echo** sebanyak 100x?
oleh karena itu kita akan menggunakan array agar lebih efisien

1. Apa itu Array?

Array adalah salah satu struktur data yang berisi sekumpulan data dan memiliki indeks. Indeks digunakan untuk mengakses nilai array (**Array Indexed**).

Indeks array selalu dimulai dari nol (**0**).

Contoh:

| | | |
|---------------|------------------|---------|
| "Hardisk 2TB" | "Flashdisk 32GB" | "Modem" |
| 0 | 1 | 2 |

Jadi, apabila kita ingin menampilkan "Hardisk 2TB", maka kita harus mengambil indeks yang ke-0.

Array menggunakan nomor sebagai identitasnya (Index) dan dimulai dengan nomor 0.

```
<?php  
$nama_variabel=array("isi variabel1","isi variabel2","isi variabel3");  
echo "Variabel " . $nama_variabel[0] .",". $nama_variabel[1] .",".  
$nama_variabel[2] ."!";  
?>
```

Berikut ini merupakan fungsi - fungsi yang berhubungan dengan array pada bahasa pemrograman PHP :

- **arsort()**. Pengurutan berdasarkan value secara descending.

- **asort()**. Pengurutan berdasarkan value secara ascending.
- **krsort()**. Pengurutan berdasarkan index/key secara descending
- **ksort()**. Pengurutan berdasarkan index/key secara ascending.
- **rsort()**. Pengurutan berdasarkan value secara descending dengan mengubah index/key.
- **sort()**. Pengurutan berdasarkan value secara ascending dengan mengubah index/key.
- **shuffle()**. Random pengurutan array.
- **current()**. Mendapatkan element array yang ditunjuk oleh pointer.
- **end()**. Pointer menunjuk pada element array terakhir.
- **key()**. Mendapatkan key yang ditunjuk oleh pointer.
- **next()**. Pointer menunjuk pada element selanjutnya.
- **prev()**. Pointer menunjuk pada element sebelumnya.
- **reset()**. Memindahkan pointer ke array awal (element pertama).
- **count()**. Menghitung jumlah element array.
- **array_search()**. Mencari posisi key berdasarkan value ke dalam array.
- **array_key_exists()**. memeriksa suatu key didalam arary.
- **in_array()**. Memeriksa suatu element kedalam array.

```

$nama_variabel=Array("isi variabel1","isi variabel2","isi variabel3");
echo "Variabel isi variabel1,isi variabel2,isi variabel3";
echo "<p><strong>i</strong></p> By HAFIS RAMADHAN"

```

2. Membuat Array di PHP

Array di PHP dapat kita buat dengan 3 bentuk fungsi, dengan kurung biasa **array()** dan tanda kurung kotak **[]**.

Contoh:

1) Membuat array kosong

```
$buah = array();
$hobi = [];
```

2) Membuat array sekaligus mengisinya

```
$minuman = array("Kopi", "Teh", "Jus Jeruk");
$makanan = ["Nasi Goreng", "Soto", "Bubur"];
```

3) Membuat array dengan mengisi indeks tertentu

```
$anggota[1] = "Dian";
$anggota[2] = "Muhar";
$anggota[0] = "Ahmadi";
```

Cukup mudah bukan.

Oya, array dapat kita isi dengan tipe data apa saja. Bahkan dicampur juga boleh.

Contoh:

```
<?php
```

```
$item = ["Bunga", 123, 39.12, true];
```

```

EXPLORER
ALGORITMA 4
01 Perulangan for.php
02 Perulangan while.php
03 While.php
04 Perulangan Foreach.php
05 Perulangan Bersarang.php
06 Perulangan While.php
07 Pernyataan Do-While.php
08 Pernyataan For.php
09 Pernyataan Foreach.php
10 Perulangan Forada PHP.php
11 Melompati (skip) For Loop...
12 Tips Optimasi Perulangan F...
13 Looping For di dalam Tabel...
14 Cara Membuat tabel dengan...
15 penggunaan forloop.php
16 Penggunaan Pada Array.php
17 For Loop dengan Step yan...
18 WHILE LOOP.php
19 Pengenalan Array dengan ...
20 penggunaan dasar do whil...
21 Perbedaan dengan while L...
22 memahami kondisi loop ph...
23 Penggunaan Foreach Loop...
24 foreach dengan mendapat...
25 Pengenalan Array Multidi...
26 penggunaan break.php
27 penggunaan continue.php
28 break pada Switch.php
29 break dan continue.php
30 Array.php
31 array di php.php
32 menampilkan isi array ... U
PROJECT ALGORITMA 4-AHM...
PROJECT ALGORITMA 4... M

```

```

31 array di php.php
1 <?php
2 //() Membuat array kosong
3 $buah = array();
4 $buah[0] = "Ketimbul";
5 //() Membuat array sekaligus mengisinya
6 $sejukmanis = array("kopi", "teh", "Jus Jeruk");
7 $sejukmanis = ["Nasi Goreng", "Soto", "Bubur"];
8 //() Membuat array dengan mengisi indeks tertentu
9 $anggota[1] = "Dilan";
10 $anggota[2] = "Muhammad";
11 $anggota[3] = "Reyissa";
12
13 echo "<p><strong><br> HAFIS RAMADHAN</strong></p>";
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
30 Array.php
31 array di php.php
32 menampilkan isi array ... U
PROJECT ALGORITMA 4-AHM...
PROJECT ALGORITMA 4... M

```

3. Menampilkan isi Array

Untuk menampilkan isi array, kita bisa mengaksesnya melalui indeks.

Contoh:

```

<?php
// membuat array
$barang = ["Buku Tulis", "Penghapus", "Spidol"];

// menampilkan isi array
echo $barang[0]."<br>";
echo $barang[1]."<br>";
echo $barang[2]."<br>";

```

Hasilnya:

localhost:8080/array/menampil

Buku Tulis
Penghapus
Spidol

```

EXPLORER
ALGORITMA 4
01 Perulangan for.php
02 Perulangan while.php
03 While.php
04 Perulangan Foreach.php
05 Perulangan Bersarang.php
06 Perulangan While.php
07 Pernyataan Do-While.php
08 Pernyataan For.php
09 Pernyataan Foreach.php
10 Perulangan Forada PHP.php
11 Melompati (skip) For Loop...
12 Tips Optimasi Perulangan F...
13 Looping For di dalam Tabel...
14 Cara Membuat tabel dengan...
15 penggunaan forloop.php
16 Penggunaan Pada Array.php
17 For Loop dengan Step yan...
18 WHILE LOOP.php
19 Pengenalan Array dengan ...
20 penggunaan dasar do whil...
21 Perbedaan dengan while L...
22 memahami kondisi loop ph...
23 Penggunaan Foreach Loop...
24 foreach dengan mendapat...
25 Pengenalan Array Multidi...
26 penggunaan break.php
27 penggunaan continue.php
28 break pada Switch.php
29 break dan continue.php
30 Array.php
31 array di php.php
32 menampilkan isi array ... U
PROJECT ALGORITMA 4-AHM...
PROJECT ALGORITMA 4... M

```

```

32 menampilkan isi array ... U
1 <?php
2 // membuat array
3 $barang = ["Buku Tulis", "Penghapus", "Spidol"];
4
5 // menampilkan isi array
6 echo $barang[0]."<br>";
7 echo $barang[1]."<br>";
8 echo $barang[2]."<br>";
9
10 echo "<p><strong><br> HAFIS RAMADHAN</strong></p>";
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
30 Array.php
31 array di php.php
32 menampilkan isi array ... U
PROJECT ALGORITMA 4-AHM...
PROJECT ALGORITMA 4... M

```

Tapi cara ini kurang efektif, karen akita mencetak satu per satu. Nanti kalau datanya ada 1000, berarti harus ngetik perintah **echo** sebanyak 1000.
Lalu bagaimana kah?

Biasanya kita menggunakan perulangan.

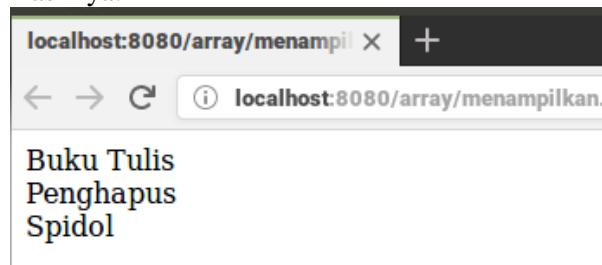
Contoh:

```
<?php
```

```
// membuat array
$barang = ["Buku Tulis", "Penghapus", "Spidol"];
// menampilkan isi array dengan perulangan for
for($i=0; $i < count($barang); $i++) {
    echo $barang[$i]."<br>";
}
```

Kita bisa menggunakan fungsi **count()** untuk menghitung banyaknya isi array. Pada contoh di atas isi array sebanyak 3, maka perulangan akan dilakukan sebanyak 3x.

Hasilnya:



Selain menggunakan perulangan **for**, kita juga bisa menggunakan perulangan **while** dan **foreach**.

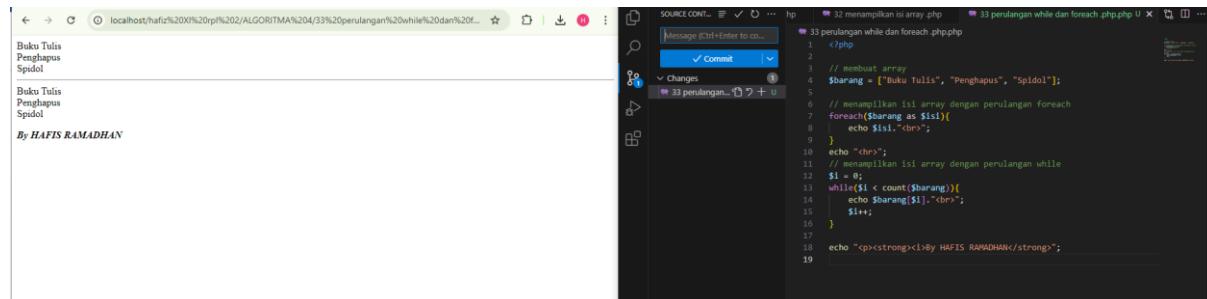
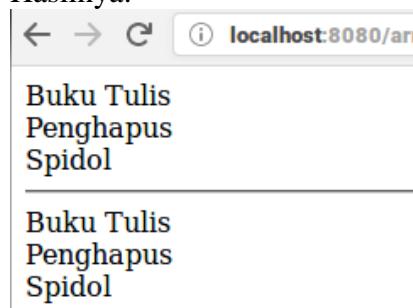
Contoh:

<?php

```
// membuat array
$barang = ["Buku Tulis", "Penghapus", "Spidol"];

// menampilkan isi array dengan perulangan foreach
foreach($barang as $isi){
    echo $isi."<br>";
}
echo "<hr>";
// menampilkan isi array dengan perulangan while
$i = 0;
while($i < count($barang)){
    echo $barang[$i]."<br>";
    $i++;
}
```

Hasilnya:



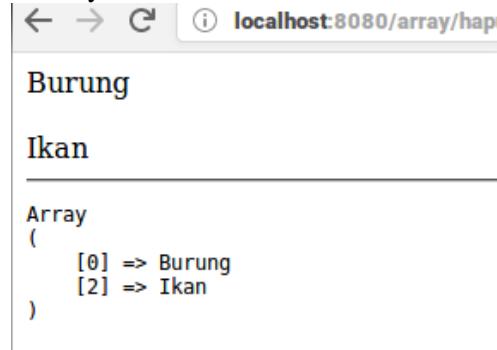
4. Menghapus isi Array

Untuk menghapus isi array, kita bisa menggunakan fungsi **unset()**. Fungsi ini juga dapat digunakan untuk menghapus variabel.

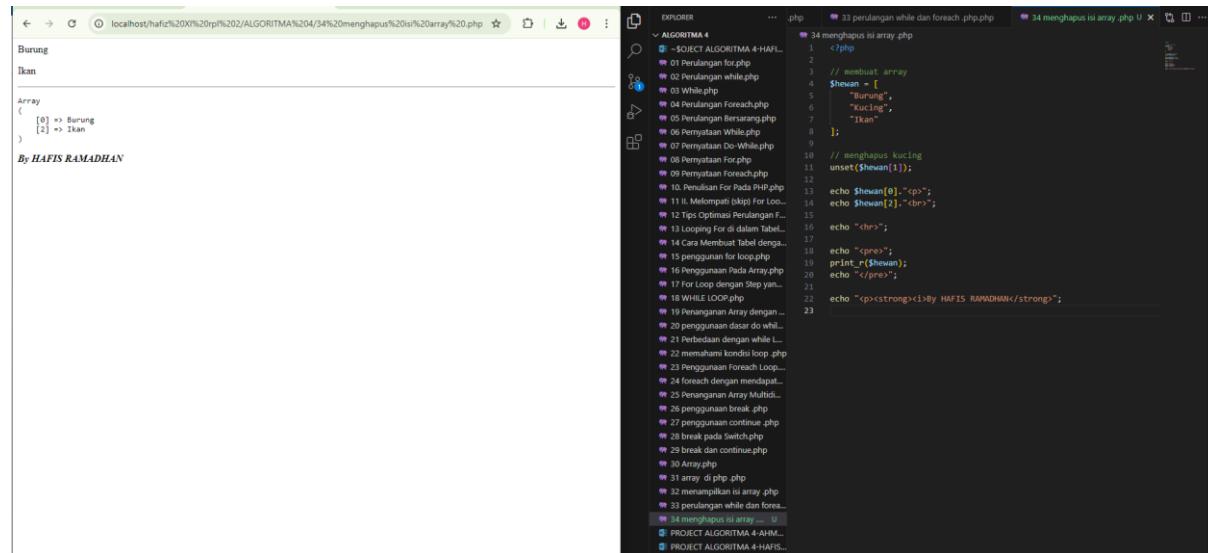
Contoh:

```
<?php  
  
// membuat array  
$hewan = [  
    "Burung",  
    "Kucing",  
    "Ikan"  
];  
  
// menghapus kucing  
unset($hewan[1]);  
  
echo $hewan[0]."<br>";  
echo $hewan[1]."<br>";  
echo $hewan[2]."<br>";  
  
echo "<hr>";  
  
echo "<pre>";  
print_r($hewan);  
echo "</pre>";
```

Hasilnya:



```
Burung  
Ikan  
  
Array  
(  
    [0] => Burung  
    [2] => Ikan  
)
```



```
EXPLORER ... php 34 menghapus isi array.php 34 perulangan while dan foreach.php 34 menghapus isi array.php U ...  
ALGORITMA 4 ...  
  - SOJET ALGORITMA 4-HAFIZ...  
    01 Perulangan for.php  
    02 Perulangan while.php  
    03 While.php  
    04 Perulangan Foreach.php  
    05 Perulangan Bersarang.php  
    06 Pernyataan While.php  
    07 Pernyataan Do-While.php  
    08 Pernyataan For.php  
    09 Pernyataan Foreach.php  
    10. Perulangan For Pada PHP.php  
    11. Melompati (skip) For Loop...  
    12. Tips Optimasi Perulangan F...  
    13. Looping For di dalam Tabel...  
    14. Cara Membuat Tabel dengan...  
    15. penggunaan for loop.php  
    16. Penggunaan Pada Array.php  
    17. For Loop dengan Step yan...  
    18. WHILE LOOP.php  
    19. Penggunaan Array dengan ...  
    20. penggunaan dasar do whil...  
    21. Perbedaan dengan while ...  
    22. memahami kondisi loop.php  
    23. Penggunaan Foreach Loop...  
    24. foreach dengan mendapat...  
    25. Penggunaan Array Multidi...  
    26. penggunaan break.php  
    27. penggunaan continue.php  
    28. break pada Switch.php  
    29. break dan continue.php  
    30. Array.php  
    31. array di php.php  
    32. menampilkan isi array.php  
    33. perulangan while dan fore...  
    34. menghapus isi array ... U  
  - PROJECT ALGORITMA 4-HAFIZ...  
  - PROJECT ALGORITMA 4-HAFIZ...
```

```
1 <?php  
2 // membuat array  
3 $hewan = [  
4     "Burung",  
5     "Kucing",  
6     "Ikan"  
7 ];  
8 // menghapus kucing  
9 unset($hewan[1]);  
10 echo $hewan[0]."<br>";  
11 echo $hewan[2]."<br>";  
12 echo "<hr>";  
13 echo "<pre>";  
14 print_r($hewan);  
15 echo "</pre>";  
16 echo "<hr>";  
17 echo "<pre>";  
18 print_r($hewan);  
19 echo "</pre>";  
20 echo "<hr>";  
21 echo "<pre><strong><li>By HAFIZ RAMADHAN</li></strong></pre>";  
22 echo "<hr>";  
23
```

Pada contoh di atas, Kita menggunakan fungsi **print_r()** untuk menampilkan array secara mentah (*raw*). Biasanya fungsi ini digunakan untuk *debugging*.

5. Menambahkan isi Array

Ada dua cara yang bisa dilakukan untuk menambah isi array:

1. Mengisi langsung ke nomer indeks yang ingin ditambahkan
2. Mengisi langsung ke indeks terakhir

Mari kita coba kedua-duanya.

```

<?php
// membuat array
$hobi = [
    "Membaca",
    "Menulis",
    "Ngeblog"
];

// menambahkan isi pada indeks ke-3
$hobi[3] = "Coding";

// menambahkan isi pada indeks terakhir
$hobi[] = "Olahraga";

// cetak array dengan perulangan
foreach($hobi as $hobiku){
    echo $hobiku."<br>";
}
?>

```

Hasilnya:

Membaca
Menulis
Ngeblog
Coding
Olahraga

```

<?php
// menambahkan isi pada indeks ke-3
$hobi[3] = "Coding";
// menambahkan isi pada indeks terakhir
$hobi[] = "Olahraga";

```

Apabila kita menambahkan pada indeks yang sudah memiliki isi, maka isinya akan ditindih dengan yang baru.

Contoh:

```

<?php
// membuat array
$user = [
    "dian",
    "muhar",
    "ahmadimuslim"
];

// mengisi array pada indek ke-1 ("muhar")
$user[1] = "Ahmadi";

// mencetak isi array
echo "<pre>";

```

```

print_r($user);
echo "</pre>";
?>

```

Hasilnya:

```

Array
(
    [0] => dian
    [1] => Ahmadi
    [2] => ahmadimuslim
)

```

```

Array
{
    [0] => piuss
    [1] => arjun
    [2] => hafis
}
By HAFIS RAMADHAN

EXPLORER                         enambahkan isi array.php
ALGORITMA 4                         36 menambahkan pada indeks yang sudah memiliki isi.php U
+--> SELECT ALGORITMA 4-HAFIS...
    01 Perulangan for.php
    02 Perulangan while.php
    03 While.php
    04 Perulangan Forach.php
    05 Perulangan Bersarang.php
    06 Pernyataan Do-While.php
    07 Pernyataan Do-While.php
    08 Pengulangan For.php
    09 Pernyataan Forach.php
    10. Perulangan For Pada PHP.php
    11 & Melengkapi (dikti) For Loop...
    12 Tips Optimasi Penerapan For...
    13 Looping For di Selain Tabel...
    14 Cara Membuat Tabel dengan...
    15 penggunaan for loop.php
    16 Penggunaan Pada Array.php
    17 For Loop dengan Step yaitu...
    18 WHILE LOOP.php
    19 Pengulangan Array dengan...
    20 penggunaan dasar do whil...
    21 Perbedaan dengan while L...
    22 memahami kondisi loop.php
    23 Penggunaan Forach Loop...
    24 forach dengan mendapat...
    25 Pengulangan Array Multidi...
    26 penggunaan break.php
    27 penggunaan continue.php
    28 break pada Switch.php
    29 break dan continue.php
    30 Array.php
    31 array di php.php
    32 menampilkan isi array.php
    33 perulangan while dan fore...
    34 menghapus isi array.php
    35 menambahkan isi array.php
    36 menambahkan isi... U
    PROJECT ALGORITMA 4-AHM...
    PROJECT ALGORITMA 4-HAFIS...

```

6. Array Asosiatif

Array asosiatif adalah array yang indeksnya tidak menggunakan nomer atau angka. Indeks array asosiatif berbentuk kata kunci.

Contoh:

```

<?php
// membuat array asosiatif
$artikel = [
    "judul" => "Belajar Pemrograman PHP",
    "penulis" => "ahmadimuslim",
    "view" => 128
];

// mencetak isi array assosiatif
echo "<h2>".$artikel["judul"]."</h2>";
echo "<p>oleh: ".$artikel["penulis"]."</p>";
echo "<p>View: ".$artikel["view"]."</p>";

```

Hasilnya:

localhost:8080/array/asosiatif.x +

localhost:8080/array/asosiatif.php

Belajar Pemrograman PHP

oleh: petanikode

View: 128

```

<?php
$artikel = [
    "judul" => "Belajar Pemrograman PHP dan algoritma",
    "penulis" => "hafiz ramadhan",
    "view" => 123
];
// mencetak isi array assosiatif
echo "<h2>$artikel["judul"].</h2>";
echo "<p>oleh: $artikel["penulis"].</p>";
echo "<p>view: $artikel["view"].</p>";
echo "<p><strong>iBy HAFIS RAMADHAN</strong></p>";

```

Pada array asosiatif, kita menggunakan tanda `=>` untuk mengasosiasikan sebuah kata kunci dengan isi array. Selain menggunakan tanda `=>`, kita juga bisa membuat array asosiatif seperti ini:

```

<?php
$email["subjek"] = "Apa Kabar?";
$email["pengirim"] = "dian@ahmadimuslim.com";
$email["isi"] = "Apa kabar? sudah lama tidak berjumpa";

echo "<pre>";
print_r($email);
echo "</pre>";

```

Hasilnya:

```

Array
(
    [subjek] => Apa Kabar?
    [pengirim] => dian@ahmadimuslim
    [isi] => Apa kabar? sudah lama tidak berjumpa
)

```

```

<?php
$array = [
    "subjek" => "Apa Kabar?",
    "pengirim" => "dian@ahmadimuslim",
    "isi" => "Apa kabar? sudah lama kita tidak berjumpa"
];
By HAFIS RAMADHAN

```

```

<?php
$subjek = "Apa Kabar?";
$pengirim = "hafiz@gmail.com";
$isi = "Apa kabar hafiz? sudah lama kita tidak berjumpa";
echo "<pre>";
print_r(["subjek" => $subjek, "pengirim" => $pengirim, "isi" => $isi]);
echo "</pre>";
echo "<p><strong>iBy HAFIS RAMADHAN</strong></p>";

```

Jika array indexed menggunakan nomor sebagai identitasnya dan dimulai dengan nomor 0 maka array ini adalah dengan menggunakan penamaan, untuk lebih jelasnya lihat contoh

```

<?php
$nama_variabel=array("nama1"=>"70","nama2"=>"67","nama3"=>"89");
echo "hasil dari program " . $nama_variabel['namal'] . " nilai.";
?>

```

```

<?php
$nama_variabel=array("nama1">"70","nama2">"67","nama3">"80");
echo "hasil dari program " . $nama_variabel["nama1"] . " nilai";
echo "<p><strong><i>By HAFIS RAMADHAN</i></strong>" ;
?>

```

Key, Value, Indexed dan Associative Array

Sebelum kita membahas foreach pada PHP, penting untuk memahami tentang key dan value pada array, karena hal tersebut akan lebih mempermudah kita memahami perulangan foreach.

Seperti telah dijelaskan pada artikel Memahami Array Pada PHP, indexed array merupakan array dengan key berupa angka, misal:

\$bulan = array('Januari', 'Februari', 'Maret')

Pada contoh diatas, value dari array adalah Januari, Februari, dan Maret, sedangkan key nya adalah 0, 1, dan 2. Karena key tidak didefinisikan, maka nilai key otomatis mulai dari 0 bentuk diatas sama dengan:

array(0=>'Januari',1=>'Februari',3=>'Maret')

sedangkan associative array merupakan array dengan key berupa nilai tertentu, misal:

array('jenis' => 'Mobil', 'merk' => 'Toyota', 'tipe' => 'Vios')

Pada contoh diatas, value dari array adalah Mobil, Toyota, dan Vios. Sedangkan key-nya adalah jenis, merk, dan tipe.

I. Cara Penulisan Foreach Pada PHP

Setelah kita faham tentang key dan value pada array, kita

Pada PHP foreach dapat ditulis dalam dua cara yaitu:

1Mengabaikan nilai key

Pada cara ini, nilai key akan diabaikan/tidak digunakan:

```

foreach ($array as $value) {
    statement;
}

```

Penjelasan:

- \$array adalah nama variabel array yang akan kita gunakan untuk perulangan.
- \$value adalah nama variabel yang mewakili nilai/data dari array yang ada pada variabel \$array. Kita bebas memberi nama variabel ini, umumnya variabel tersebut diberi nama \$value, \$val atau cukup \$v.

Cara ini bermanfaat jika kita hanya ingin menggunakan data value dari array dan mengabaikan data key nya

Contoh penggunaan:

```

$bulan = array ('Januari', 'Februari', 'Maret');
foreach ($bulan as $nama_bulan) {
    echo $nama_bulan . '<br/>';
}

```

Hasil yang kita peroleh:

```

Januari
Februari
Maret

```

```

1 <?php
2 $bulan = array ('Januari', 'Februari', 'Maret');
3 foreach ($bulan as $nama_bulan) {
4     echo $nama_bulan . '<br/>';
5 }
6
7 echo "<p><strong><i>By HAFIS RAMADHAN</i></strong></p>";

```

2. Menyertakan nilai key

Model kedua adalah dengan menyertakan nilai key, cara ini berlaku baik untuk indexed array maupun associative array. adapun format penulisannya adalah sebagai berikut:

```
foreach ($array as $key => $value) {
    statement;
}
```

Penjelasan:

- \$array adalah nama variabel array yang akan kita gunakan untuk perulangan.
- \$key merupakan nama variabel yang mewakili nilai index yang ada di dalam variabel \$array. Kita bebas memberi nama variabel ini, umumnya variabel tersebut diberi nama \$key atau cukup \$k
- Tanda => digunakan untuk menghubungkan antara \$key dan \$value nya
- Seperti pada model pertama, \$value adalah nama variabel yang mewakili data value yang ada di dalam variabel \$array. Kita juga bebas memberi nama variabel ini, umumnya nama yang digunakan adalah \$value, \$val, atau cukup \$v

Contoh penggunaan pada indexed array:

```
$bulan = array ('Januari', 'Februari', 'Maret');
foreach ($bulan as $index => $nama_bulan) {
    echo ($index + 1) . ' . ' . $nama_bulan . '<br/>';
}
```

Hasil yang kita peroleh:

1. Januari
2. Februari
3. Maret

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/41%20penggunaan%20pada%20indexed%20array.php
1. Januari
2. Februari
3. Maret
By HAFIS RAMADHAN

```

The screenshot shows a browser window with the URL `localhost/hafiz%20X%20rp%202/ALGORITMA%204/41%20penggunaan%20pada%20indexed%20array.php`. The page content displays three months: Januari, Februari, and Maret. Below the content, it says "By HAFIS RAMADHAN". On the right side of the browser, there is an "EXPLORER" sidebar containing a list of PHP files related to loops and arrays, such as "05 Pengulangan Banyak.php", "06 Perintasan While.php", "07 Perintasan Do-While.php", "08 Perintasan For.php", "09 Perintasan ForEach.php", "10. Penulisan For Pada PHP.php", "11. Il. Melompati (skip) For Loop...", "12. Tips Optimasi Perulangan F...", "13 Looping For di dalam Tabel...", "14 Cara Membuat Tabel dengan...", "15 penggunaan for loop.php", "16 Penggunaan Pada Array.php", "17 For Loop dengan Step yang...", "18 WHILE LOOP.php", "19 Pengenalan Array dengan ...", "20 penggunaan dasar do whil...", "21 Perbedaan dengan while L...", "22 memahami kondisi loop .php", "23 Penggunaan Foreach Loop...", "24 foreach dengan mendapat...", "25 Pengenalan Array Multidi...", "26 penggunaan break .php", "27 penggunaan continue .php", "28 break pada Switch.php", "29 break dan continue.php", "30 Array.php", "31 array di php.php", "32 menampilkan isi array.php", "33 perulangan while dan fore...", "34 menghapus isi array.php", "35 menambahkan isi array.php", "36 menambahkan pada indeks...", "37 array assosiatif.php", "38 menggabungkan sebuah k... .", "39 array indexed.php", "40 penulisan foreach pada ph...", "41 penggunaan pada ind... U", "PROJECT ALGORITMA 4-AHM...", "PROJECT ALGORITMA 4-HAFIS..."

Pada contoh diatas terlihat bahwa setiap array pasti memiliki key, sehingga meskipun key tersebut tidak kita tulis, key tersebut tetap ada dan bisa digunakan.

Contoh pada associative array:

```

$kendaraan = array('jenis' => 'Mobil', 'merk' => 'Toyota', 'tipe' =>
'Vios');
foreach ($kendaraan as $key => $val) {
    echo ucfirst($key) . ' : ' . $val . '<br/>';
}

```

Hasil yang kita peroleh:

Jenis: Mobil

Merk: Toyota

Tipe: Vios

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/42%20associative%20array%20.php
Jenis: Mobil
Merk: Lamborgini
Tipe: veneno
By HAFIS RAMADHAN

```

The screenshot shows a browser window with the URL `localhost/hafiz%20X%20rp%202/ALGORITMA%204/42%20associative%20array%20.php`. The page content displays three items: Jenis: Mobil, Merk: Lamborgini, and Tipe: veneno. Below the content, it says "By HAFIS RAMADHAN". On the right side of the browser, there is an "EXPLORER" sidebar containing a list of PHP files related to loops and arrays, such as "06 Perintasan While.php", "07 Perintasan Do-While.php", "08 Perintasan For.php", "09 Perintasan ForEach.php", "10. Penulisan For Pada PHP.php", "11. Il. Melompati (skip) For Loop...", "12. Tips Optimasi Perulangan F...", "13 Looping For di dalam Tabel...", "14 Cara Membuat Tabel dengan...", "15 penggunaan for loop.php", "16 Penggunaan Pada Array.php", "17 For Loop dengan Step yang...", "18 WHILE LOOP.php", "19 Pengenalan Array dengan ...", "20 penggunaan dasar do whil...", "21 Perbedaan dengan while L...", "22 memahami kondisi loop .php", "23 Penggunaan Foreach Loop...", "24 foreach dengan mendapat...", "25 Pengenalan Array Multidi...", "26 penggunaan break .php", "27 penggunaan continue .php", "28 break pada Switch.php", "29 break dan continue.php", "30 Array.php", "31 array di php.php", "32 menampilkan isi array.php", "33 perulangan while dan fore...", "34 menghapus isi array.php", "35 menambahkan isi array.php", "36 menambahkan pada indeks...", "37 array assosiatif.php", "38 menggabungkan sebuah k... .", "39 array indexed.php", "40 penulisan foreach pada ph...", "41 penggunaan pada ind... U", "42 associative array.php", "PROJECT ALGORITMA 4-AHM...", "PROJECT ALGORITMA 4-HAFIS..."

Pada contoh diatas, saya menggunakan fungsi ucfirst untuk membuat huruf pertama dari key menjadi huruf besar.

3. Foreach didalam foreach (nested foreach)

Pada multidimensional array, kita akan sering membuat perulangan foreach didalam foreach.

Hati hati jika membuat perulangan dengan model seperti ini, karena sifat variabel dapat berubah ubah, maka, nilai variabel dari \$key dan \$value juga rawan berubah.

Oleh karena itu, sebaiknya mungkin memberi nama variabel untuk \$key dan \$value sesuai dengan isi datanya, sehingga memudahkan kita untuk membaca alur dari perulangan.

Contoh:

```

foreach ($array as $key => $val) {
    statement;
    foreach ($val as $key => $val) {

```

```

        statement;
    }
}

```

Pada contoh tersebut nilai variabel \$key pada foreach yang pertama akan berubah karena ditimpa oleh nilai variabel \$key pada foreach ke dua.

II. Contoh Penggunaan Foreach Pada PHP

Pada PHP, foreach dapat digunakan untuk berbagai keperluan baik untuk backend maupun frontend, pada frontend, foreach digunakan salahsatunya untuk membuat element dropdown select.

Contoh kita ingin membuat dropdown nama bulan, dibanding menggunakan cara manual, akan jauh lebih efisien jika menggunakan perulangan foreach:

```

<?php
$bulan = array (1=>'Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni',
'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember');

$opsi_bulan = '<select name="bulan">';
foreach ($bulan as $key => $value) {
    $opsi_bulan .= '<option value="' . $key . '">' . $value . '</option>' .
    "\r\n";
}
$opsi_bulan .= '</select>';

echo $opsi_bulan;

```

The screenshot shows a code editor with several tabs open. The active tab contains the following PHP code:

```

$opsi_bulan = '<select name="bulan">';
foreach ($bulan as $key => $value) {
    $opsi_bulan .= '<option value="' . $key . '">' . $value . '</option>' .
    "\r\n";
}
$opsi_bulan .= '</select>';

echo $opsi_bulan;

```

The code defines an array \$bulan containing months from January to December. It then creates a string \$opsi_bulan which is a dropdown menu where each month is an option. The code editor's sidebar shows a list of other files related to algorithm 4, such as '07 Pernyataan Do-While.php', '08 Pernyataan For.php', and '43 perulangan foreach.php'.

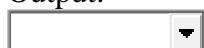
Kode HTML yang kita peroleh:

```

<select name="bulan">
<option value="1">Januari</option>
<option value="2">Februari</option>
<option value="3">Maret</option>
<option value="4">April</option>
<option value="5">Mei</option>
<option value="6">Juni</option>
<option value="7">Juli</option>
<option value="8">Agustus</option>
<option value="9">September</option>
<option value="10">Oktober</option>
<option value="11">November</option>
<option value="12">Desember</option>
</select>

```

Output:



Contoh lain adalah untuk membuat tabel HTML

```

$no = 1;
$tabel =
<table>

```

```

<tr>
    <th>No</th>
    <th>Bulan</th>
    <th>Penjualan</th>
</tr>';
foreach ($sales as $bulan => $nilai) {
    $tabel .= '
<tr>
    <td>' . $no . '</td>
    <td>' . $bulan . '</td>
    <td>' . $nilai . '</td>
</tr>';
    $no++;
}
$tabel .= '</table>';
echo $tabel;

```

Hasil:

```

<table>
    <tr>
        <th>No</th>
        <th>Bulan</th>
        <th>Penjualan</th>
    </tr>
    <tr>
        <td>1</td>
        <td>Januari</td>
        <td>5.500</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Februari</td>
        <td>7.500</td>
    </tr>
    <tr>
        <td>3</td>
        <td>Maret</td>
        <td>11.500</td>
    </tr>
    <tr>
        <td>4</td>
        <td>April</td>
        <td>8.800</td>
    </tr>
    <tr>
        <td>5</td>
        <td>Mei</td>
        <td>7.500</td>
    </tr>
</table>

```

The screenshot shows a browser window with a table titled "No Bulan Penjualan" containing data for January. Below the table, it says "By HAFIS RAMADHAN". To the right is a code editor in Visual Studio Code with an open file named "44 membuat tabel html.php". The code generates an HTML table from an associative array.

```

<?php
// ini adalah array dua dimensi
$matrik = [
    [2,3,4],
    [7,5,0],
    [4,3,8],
];

// cara mengakses isinya
echo $matrik[1][0]; //-> output: 7

```

7. Array Multi Dimensi

Array multi dimensi adalah array yang memiliki dimensi lebih dari satu. Biasanya digunakan untuk membuat matrik, graph, dan stuktur data rumit lainnya.

Contoh:

```

<?php
// ini adalah array dua dimensi
$matrik = [
    [2,3,4],
    [7,5,0],
    [4,3,8],
];

// cara mengakses isinya
echo $matrik[1][0]; //-> output: 7

```

Masi kita coba contoh yang lain:

```

<?php
// membuat array 2 dimensi yang berisi array asosiatif
$artikel = [
    [
        "judul" => "Belajar PHP & MySQL untuk Pemula",
        "penulis" => "ahmadimuslim"
    ],
    [
        "judul" => "Tutorial PHP dari Nol hingga Mahir",
        "penulis" => "ahmadimuslim"
    ],
    [
        "judul" => "Membuat Aplikasi Web dengan PHP",
        "penulis" => "ahmadimuslim"
    ]
];

// menampilkan array
foreach($artikel as $post) {
    echo "<h2>".$post["judul"]."</h2>";
    echo "<p>".$post["penulis"]."<p>";
    echo "<hr>";
}

```

Hasilnya:

Belajar PHP & MySQL untuk Pemula

Tutorial PHP dari Nol hingga Mahir

Membuat Aplikasi Web dengan PHP

PENGAYAAN

```

<?php
// membuat array 2 dimensi yang berisi array assosiatif
$artikel = [
    [
        "judul" => "Belajar PHP & MySQL untuk Pemula",
        "penulis" => "m hafidz ramadhan"
    ],
    [
        "judul" => "Tutorial PHP dari Nol hingga Mahir",
        "penulis" => "m hafidz ramadhan"
    ]
];

// menampilkan array
foreach($artikel as $post){
    echo "<h2>" . $post["judul"] . "</h2>";
    echo "<p>" . $post["penulis"] . "</p>";
    echo "<br>";
}
echo "<p><strong><i>By HAFIS RAMADHAN</i></strong></p>";

```

Contoh Program PHP Menggunakan Array

Setelah memahami tentang array diatas, sekarang saya akan memberikan contoh program dan penjelasan dari program tersebut menggunakan bahasa pemrograman PHP. Disini tentu saja hanya menunjukkan program yang mudah dipahami saja. Berikut ini adalah contoh program PHP menggunakan array.

Array 1 Dimensi PHP

Array satu dimensi adalah array yang hanya memiliki satu indeks saja. Tidak berbeda dengan bahasa pemrograman lain, Pada bahasa pemrograman PHP, index dalam array tersebut diinisialisasikan menggunakan tanda kurung besar ([]). Di bahasa pemrograman lainnya ketika kita mendapatkan sebuah variabel dengan tanda kurung tersebut, lalu terdapat angka didalamnya, itulah yang disebut index.

```

<?php
$test=array("Index pertama","Index Kedua","Index ketiga");
echo "Hasil array : 0." . $test[0] . ", 1." . $test[1] . " dan 2." .
$test[2] . ".";
?>

```

Array 2 Dimensi Pada PHP

Array atau larik dua dimensi ini memiliki dua buah index. Berbeda dengan array satu dimensi yang hanya memiliki satu index. Dalam konteks ini, array dua dimensi memiliki dua index dimana index pertama melambangkan baris, sedangkan index kedua melambangkan kolom. Sama seperti bahasa pemrograman lain. Pada pembuatan program, array dua dimensi adalah array yang paling sering digunakan karena dapat menyelesaikan masalah lebih banyak dibanding array lainnya.

```

<?php
$nilai=array(
    array(90,65,83),
    array(90,78,97),
    array(78,90,78)
);
echo"output array <br>";
echo $nilai[0]." ".$nilai[0][1]." ".$nilai[0][2]."<br>";
echo $nilai[1]." ".$nilai[1][1]." ".$nilai[1][2]."<br>";
echo $nilai[2]." ".$nilai[2][1]." ".$nilai[2][2]."<br>";
?>

```

Array 3 Dimensi PHP

Untuk array tiga dimensi ini kita bisa sebut sebagai array dalam array. Karena array tiga dimensi ini ibarat tabel, satu array bisa menyimpan banyak tabel. Index pertama pada array tersebut merupakan jumlah array maksimal yang bisa disimpan, index kedua merupakan kolom di tiap array. Sedang index ketiga merupakan baris pada tiap array. Artinya satu buah array tiga dimensi, dapat menyimpan banyak array dua dimensi.

```
<?php
```

```

$nilai=array(
    array(
        array(90,65,83),
        array(90,78,97),
        array(78,90,78)
    ),
    array(array(90,65,83),
        array(90,78,97),
        array(78,90,78)
    )
);
echo"output array <br>";
echo $nilai[0] [0]." ".$nilai[0] [0] [1]." ".$nilai[0] [0] [2]."<br>";
echo $nilai[0] [1] [0]." ".$nilai[0] [1] [1]." ".$nilai[0] [1] [2]."<br>";
echo $nilai[0] [2] [0]." ".$nilai[0] [2] [1]." ".$nilai[0] [2] [2]."<br>";
?>

```

```

<?php
$nilai=array(
    array(
        array(90,65,83),
        array(90,78,97),
        array(78,90,78)
    ),
    array(array(90,65,83),
        array(90,78,97),
        array(78,90,78)
    )
);
echo"output array <br>";
echo $nilai[0] [0]." ".$nilai[0] [0] [1]." ".$nilai[0] [0] [2]."<br>";
echo $nilai[0] [1] [0]." ".$nilai[0] [1] [1]." ".$nilai[0] [1] [2]."<br>";
echo $nilai[0] [2] [0]." ".$nilai[0] [2] [1]." ".$nilai[0] [2] [2]."<br>";
?>

```

PENALARAN

Menggabungkan Array dan Menampilkan dalam tabel HTML

```

<?php
$merk = array("Oppo", "Samsung", "Vivo", "Xiaomi", "Nokia", "Realme",
"Sonny");
$harga = array("19000000", "12000000", "16000000", "12000000",
"16000000", "11000000", "19000000");

$totalArray = count($harga);

echo "<table border='1'>";
echo "<tr>";
echo "<th>Merk handphone</th>";
echo "<th>Harga handphone</th>";
echo "</tr>";

for ($i = 0; $i < $totalArray; $i++) {
    echo "<tr>";
    echo "<td>$merk[$i]</td>";
    echo "<td>$harga[$i]</td>";
    echo "</tr>";
}
echo "</table>";
echo "<br>";

```

Hasilnya

| Merk handphone | Harga handphone |
|----------------|-----------------|
| Oppo | 19000000 |
| Samsung | 12000000 |
| Vivo | 16000000 |
| Xiaomi | 12000000 |
| Nokia | 16000000 |
| Realme | 11000000 |
| Sonny | 19000000 |

The screenshot shows a browser window displaying a table of phone models and their prices, and an IDE (Visual Studio Code) showing the PHP code used to generate that table.

Table Data:

| Merk handphone | Harga handphone |
|----------------|-----------------|
| Oppo | 19000000 |
| Samsung | 12000000 |
| Vivo | 16000000 |
| Xiaomi | 12000000 |
| i pone | 16000000 |
| Realme | 11000000 |
| Sonny | 19000000 |

PHP Code:

```

49 Menggunakan Array dan Menampilkan dalam tabel HTML.php
1 <?php
2 $merk = array("Oppo", "Samsung", "Vivo", "Xiaomi", "i pone", "Realme");
3 $harga = array("19000000", "12000000", "16000000", "12000000", "16000000", "11000000");
4
5 $totalArray = count($harga);
6
7 echo "<table border='1'>";
8 echo "<tr>";
9 echo "<th>Merk handphone</th>";
10 echo "<th>Harga handphone</th>";
11 echo "</tr>";
12
13 for ($i = 0; $i < $totalArray; $i++) {
14     echo "<tr>";
15     echo "<td>$merk[$i]</td>";
16     echo "<td>$harga[$i]</td>";
17     echo "</tr>";
18 }
19 echo "</table>";
20 echo "<br>";
21
22 echo "<p><strong><i>By HAFIS RAMADHAN</i></strong></p>";
23

```

Contoh Kasus Array Asosiatif

Dimisalkan kita akan membuat array customer yang menyimpan data nama, alamat, no. tlp, pekerjaan, dan gaji. Kemudian kita akan tampilkan datanya ke dalam bentuk tabel. Array yang kita buat adalah array asosiatif dengan jumlah customers sebanyak 3. Jadi bagaimana kita membuatnya ?

Pertama kita buat array asosiatifnya seperti berikut :

```

1 <?php
2 $customers = [ awal array kurung buka
3 [
4     "Nama" => "Andika",
5     "Alamat" => "Diponegoro",
6     "No.Telp" => "081999666777",
7     "Pekerjaan" => "PNS",
8     "Gaji" => "5.000.000"
9 ],
10 [
11     "Nama" => "Fahri",
12     "Alamat" => "Penamparan Agung",
13     "No.Telp" => "085222333444",
14     "Pekerjaan" => "Asisten Manager",
15     "Gaji" => "7.000.000"
16 ],
17 [
18     "Nama" => "Miranda",
19     "Alamat" => "Gunung Salak",
20     "No.Telp" => "085999634788",
21     "Pekerjaan" => "Wiraswasta",
22     "Gaji" => "15.000.000"
23 ],
24 ],
25 ],
26 ]; penutup array kurung tutup
27 ?>

```

Jadi terdapat 3 data customer yang masing-masing sudah memiliki elemen nama, alamat, no. tlp, pekerjaan, dan gaji. Array yang kita buat sudah menggunakan array asosiatif dengan cara Array di dalam array. Jadi awal array pertama (kurung buka paling atas) diisikan elemen array customer pertama, kedua dan ketiga (dengan membuat array data customer didalam array \$customers).

Selanjutnya kita buat code untuk akses setiap elemen di array \$customers dan menampilkan pada tabel. Seperti berikut :

```

55 <head>
56     <title>Array Asosiatif</title>
57 </head>
58 <body>
59 <table>
60     <tr>
61         <th>Nama</th>
62         <th>Alamat</th>
63         <th>No. Telp</th>
64         <th>Pekerjaan</th>
65         <th>Gaji</th>
66     </tr>
67     <?php foreach ($customers as $customer) { ?>
68         <tr>
69             <td><?php echo $customer["Nama"]; ?></td>
70             <td><?php echo $customer["Alamat"]; ?></td>
71             <td><?php echo $customer["No.Telp"]; ?></td>
72             <td><?php echo $customer["Pekerjaan"]; ?></td>
73             <td><?php echo $customer["Gaji"]; ?></td>
74         </tr>
75     <?php } ?>
76 </table>
77 </body>

```

Pada code diatas, kita membuat header tabel pada tag `<th>...</th>` sedangkan untuk isi tabel di tag `<td>...</td>`. Kita menggunakan perulangan `foreach` untuk setiap data array \$customers. Untuk sobat yang ingin mempelajari tabel php bisa mengklik link belajar tabel.

Selanjutnya kita lihat hasilnya pada browser masing-masing, berikut ini adalah contoh hasilnya :

akses direktori latihan3.php

| Nama | Alamat | No. Telp | Pekerjaan | Gaji |
|---------|------------------|--------------|-----------------|------------|
| Andika | Diponegoro | 081999666777 | PNS | 5.000.000 |
| Fahri | Penamparan Agung | 085222333444 | Asisten Manager | 7.000.000 |
| Miranda | Gunung Salak | 085999634788 | Wiraswasta | 15.000.000 |

The screenshot shows a browser window with the URL `localhost/basicphp/latihanarray/latihan3.php`. The page displays a table with three rows of data:

| No | Nama | Alamat | No. Telp | pekerjaan | Gaji |
|----|-------|-------------------|--------------|--------------------|------------|
| 1 | arjun | kebum terungan | 082345467879 | PNS | 5.000.000 |
| 2 | riyan | sekral | 082245349098 | Asisten manager | 7.000.000 |
| 3 | hafis | Kp dalam | 082311678076 | Wiraswasta | 15.000.000 |

Below the table, the text "By HAFIS RAMADHAN" is visible.

The code editor on the right shows a PHP script named `50 Kasus Array Asosiatif.php`. The script defines an associative array `$customer` and prints it to the browser using `foreach` and `echo` statements. The array contains three entries corresponding to the table rows.

```

<?php
// Contoh array asosiatif untuk menyimpan data produk
$custumer = [
    [
        "No" => 1,
        "name" => "arjun",
        "Alamat" => "kebum tengah",
        "No. Telp" => "082345467879",
        "Pekerjaan" => "PNS",
        "Gaji" => "5.000.000",
    ],
    [
        "No" => 2,
        "name" => "riyan",
        "Alamat" => "sekral",
        "No. Telp" => "082245349098",
        "Pekerjaan" => "Asisten manager",
        "Gaji" => "7.000.000",
    ],
    [
        "No" => 3,
        "name" => "hafis",
        "Alamat" => "Kp dalam",
        "No. Telp" => "082311678076",
        "Pekerjaan" => "Wiraswasta",
        "Gaji" => "15.000.000",
    ],
];
// Menampilkan data produk dalam bentuk tabel HTML
echo "<table border='1'>";
echo "<tr><th>No</th><th>Name</th><th>Alamat</th><th>No. Telp</th><th>Pekerjaan</th><th>Gaji</th></tr>";
foreach ($customer as $customer) {
    echo "<tr>";
    echo "<td>" . $customer["No"] . "</td>";
    echo "<td>" . $customer["name"] . "</td>";
    echo "<td>" . $customer["Alamat"] . "</td>";
    echo "<td>" . $customer["No. Telp"] . "</td>";
}

```

PENGAYAAN ARRAY PHP

Pelajari dasar-dasar array di PHP termasuk cara membuat, mengakses, dan memodifikasi array untuk meningkatkan kemampuan pemrograman kamu.

Mengenal Array PHP

PHP adalah bahasa skrip yang populer digunakan dalam pengembangan web. Salah satu fitur fundamental dari PHP yang sering digunakan adalah array. Array memungkinkan kamu menyimpan sejumlah data dalam satu variabel, yang memudahkan pengelolaan dan akses data.

Apa Itu Array?

Array adalah struktur data yang dapat menyimpan banyak nilai di bawah satu nama variabel, dengan setiap nilai dapat diakses melalui indeks atau kunci. Di PHP, ada tiga jenis array:

Indexed arrays - Array dengan indeks numerik.

Associative arrays - Array dengan kunci sebagai string untuk menyimpan data.

Multidimensional arrays - Array yang mengandung satu atau lebih array.

Membuat Array

Untuk membuat array di PHP, kamu bisa menggunakan fungsi `array()` atau menggunakan tanda kurung siku `[]`. Berikut adalah contoh cara membuat array:

```
// Menggunakan fungsi array()
$buah = array('apel', 'jeruk', 'pisang');

// Menggunakan tanda kurung siku []
$warna = ['merah', 'hijau', 'biru'];
```

```

localhost/hafiz%20xi%20rp%202/ALGORITMA%20/51%20membuat%20array%20.php
mangga
merah
hafis
manga
anggur
2
merah
hijau
biru
nama: hafis
umur: 16
pekerjaan: Pengembang Web
By HAFIS RAMADHAN

```

```

1 <?php
2 // Menggunakan fungsi array()
3 $buah = array('apel', 'jeruk', 'pisang');
4 // Menggunakan tanda kurung siku []
5 $warna = ['merah', 'hijau', 'biru'];
6
7 $pengguna = ['nama' => 'hafis', 'umur' => 16, 'pekerjaan' => 'Pengembang Web'];
8
9 $buah[1] = 'mangga';
10 $buah[2] = 'anggur';
11
12 echo $buah[1] . "<br>"; // akan menampilkan 'jeruk'
13 echo $warna[0]; // akan menampilkan 'merah'
14 echo $pengguna['nama'] . "<br>"; // akan menampilkan 'Ali'
15 echo $buah[1] . "<br>"; // akan menampilkan 'manga'
16 echo $buah[2]; //<br> // akan menambahkan 'anggur' ke akhir array
17 unset($buah[1]); // akan menghapus elemen dengan indeks 1 dari array
18 echo count($buah) . "<br>"; // akan menampilkan jumlah elemen dalam array
19
20 foreach ($warna as $w) {
21     echo $w . "<br>";
22 }
23
24 foreach ($pengguna as $kunci => $nilai) {
25     echo $kunci . ':' . $nilai . "<br>";
26 }
27
28 echo "<p><strong>i</strong>By HAFIS RAMADHAN</p>";

```

Mengakses Data Array

Kamu bisa mengakses data dalam array dengan menunjukkan indeksnya, dimulai dari 0 untuk array berindeks dan dengan nama kunci untuk array asosiatif.

```
echo $buah[1]; // akan menampilkan 'jeruk'
echo $warna[0]; // akan menampilkan 'merah'
```

Untuk array asosiatif, kamu harus menggunakan kunci sebagai indeks:

```
$pengguna = ['nama' => 'Ali', 'umur' => 25, 'pekerjaan' => 'Pengembang Web'];
echo $pengguna['nama']; // akan menampilkan 'Ali'
```

Memodifikasi Array

Kamu bisa mengubah nilai dari elemen array dengan menetapkan nilai baru ke elemen tersebut dengan indeksnya.

```
$buah[1] = 'mangga';
echo $buah[1]; // akan menampilkan 'mangga'
```

Kamu juga bisa menambahkan elemen baru ke array:

```
$buah[] = 'anggur'; // akan menambahkan 'anggur' ke akhir array
```

Menghapus Elemennya

Untuk menghapus elemen, kamu bisa menggunakan fungsi `unset()`.

```
unset($buah[1]); // akan menghapus elemen dengan indeks 1 dari array $buah
```

Menghitung Elemen Array

Fungsi `count()` digunakan untuk mengetahui berapa banyak elemen yang ada dalam array:

```
echo count($buah); // akan menampilkan jumlah elemen dalam array $buah
```

Looping Melalui Array

Kamu bisa menggunakan `foreach` untuk beriterasi melalui setiap elemen dalam array.

```
foreach ($warna as $w) {
    echo $w . '<br>';
}
```

Untuk array asosiatif, kamu juga bisa mengakses kuncinya:

```
foreach ($pengguna as $kunci => $nilai) {
    echo $kunci . ':' . $nilai . '<br>';
}
```

Dengan mengenal array PHP, kamu dapat memanfaatkan fitur ini untuk menyimpan dan mengelola data dengan lebih efisien dalam aplikasi webmu. Mulai praktikkan dengan berbagai contoh di atas untuk meningkatkan pemahaman dan keterampilan pemrograman kamu di PHP.

MULTIDIMENSIONAL ARRAY

Pelajari penggunaan array multidimensi di PHP dengan penjelasan sederhana dan contoh kode untuk memudahkan pemahaman tentang struktur data yang kompleks ini.

Array merupakan struktur data penting yang memungkinkan kamu menyimpan dan mengelola kumpulan nilai dalam satu variabel. Dalam PHP, array bisa lebih kompleks lagi dengan adanya array multidimensi. Array ini memungkinkan kamu untuk menata data dalam bentuk tabel, atau bahkan struktur yang lebih rumit.

Mengenal Array Multidimensi di PHP

Array multidimensi adalah array yang berisi satu atau lebih array di dalamnya. Kamu bisa membayangkannya sebagai tabel dengan baris dan kolom, di mana setiap elemen array bisa lagi berisi array lain.

Contohnya, jika kamu ingin menyimpan informasi tentang film, di mana tiap film terdiri dari judul, tahun, dan direktur, kamu bisa menggunakan array multidimensi seperti berikut:

```
$films = array(  
    array("The Shawshank Redemption", 1994, "Frank Darabont"),  
    array("The Godfather", 1972, "Francis Ford Coppola"),  
    array("The Dark Knight", 2008, "Christopher Nolan")  
);
```

The screenshot shows a code editor with several files listed in the sidebar. The main file, `statif.php`, contains the following code:

```
1 <?php  
2 $films = array(  
3     array("The Shawshank Redemption", 1994, "Frank Darabont"),  
4     array("The Godfather", 1972, "Francis Ford Coppola"),  
5     array("The Dark Knight", 2008, "Christopher Nolan")  
6 );  
7 echo $films[0][0] . "<br>" . $films[0][1] . "<br>" . $films[0][2];  
8 echo "<br>" . $films[1][0] . "<br>" . $films[1][1] . "<br>" . $films[1][2];  
9 echo "<br>" . $films[2][0] . "<br>" . $films[2][1] . "<br>" . $films[2][2];  
10  
11 $totalFilms = count($films);  
12 echo "<br>Jumlah film: " . $totalFilms; // Menampilkan jumlah film  
13  
14 echo "<p><strong>By HAFIS RAMADHAN</strong></p>";  
15  
16 Menggunakan Pada Array.php  
17 For Loop dengan Step yan...  
18 WHILE LOOP.php  
19 Pengetahuan Array dengan ...  
20 penggunaan dasar do whi...  
21 Perbedaan dengan while ...  
22 memahami kondisi loop.php  
23 Penggunaan ForEach Loop...  
24 foreach dengan mendapt...  
25 Penggunaan Break.php  
26 penggunaan break.php  
27 penggunaan continue.php  
28 break pada Switch.php  
29 break dan continue.php  
30 Array.php  
31 array() php.php  
32 menampilkan isi array.php  
33 perulangan while dan fore...  
34 menghapus isi array.php  
35 menambahkan isi array.php  
36 menambahkan pada indeks...  
37 array assosiatif.php  
38 menggabungkan sebuah ...  
39 array indexed.php  
40 penulisan forech pada ph...  
41 penggunaan pada indexed ...  
42 associative array.php  
43 perulangan forech.php  
44 membuat tabel html.php  
45 array multi dimensi.php  
46 Array 1 Dimensi PHP.php  
47 array 2 dimensi pada PHP ...  
48 array 3 dimensi PHP.php  
49 Menggabungkan Array dan ...  
50 Kasus Array Assosiatif.php  
51 Membuat array.php  
52 Mengenal Array Multidimensi di PHP.php
```

Mengakses Nilai dalam Array Multidimensi

Untuk mengakses nilai dalam array multidimensi, kamu perlu menggunakan indeks yang sesuai. Indeks ini umumnya merupakan angka yang menunjukkan posisi dari elemen yang kamu ingin akses.

Misalnya, jika kamu ingin mencetak judul film pertama dari contoh array di atas, kamu akan menuliskan:

```
echo $films[0][0]; // Outputs: The Shawshank Redemption
```

Menambahkan Nilai ke Dalam Array Multidimensi

Untuk menambahkan nilai baru ke dalam array multidimensi, kamu bisa menggunakan fungsi `array_push()` atau hanya menggunakan kurung siku `[]`. Namun, pastikan kamu mengetahui indeks array yang benar untuk menempatkan data yang baru.

Misalnya, untuk menambahkan film baru ke dalam daftar:

```
$newFilm = array("Inception", 2010, "Christopher Nolan");  
$films[] = $newFilm; // Menambahkan di akhir array films
```

Melakukan Loops Pada Array Multidimensi

Ketika bekerja dengan array multidimensi, sangat umum untuk menggunakan perulangan, seperti `foreach` atau `for`, untuk mengelola atau menampilkan data.

Contohnya, untuk menampilkan informasi semua film dalam array:

```
foreach ($films as $filmDetails) {
```

```

echo "Judul: " . $filmDetails[0] . "<br>";
echo "Tahun: " . $filmDetails[1] . "<br>";
echo "Sutradara: " . $filmDetails[2] . "<br><br>";
}

```

Fungsi Array dalam Array Multidimensi

PHP menawarkan berbagai fungsi untuk bekerja dengan array, dan banyak di antaranya juga bisa digunakan untuk array multidimensi. Sebagai contoh, kamu bisa menggunakan count() untuk mendapatkan jumlah elemen atau sort() untuk mengurutkan elemen dalam array.

Sebagai contoh penggunaan count() pada array multidimensi:

```

$totalFilms = count($films);
echo "Jumlah film: " . $totalFilms; // Menampilkan jumlah film

```

Dengan memahami konsep array multidimensi di PHP, kamu bisa menyusun dan mengelola data yang kompleks dengan lebih efisien. Eksperimen dengan array ini dan fungsi-fungsi yang tersedia akan membantu kamu menjadi programmer PHP yang lebih cakap dalam menyelesaikan berbagai masalah pemrograman.

ASSOCIATIVE ARRAY

Pelajari cara menggunakan associative array di PHP untuk mengelola data dengan pasangan kunci-nilai.

PHP menawarkan berbagai tipe data array, dan salah satu yang paling berguna adalah associative array. Associative array memungkinkan kamu untuk menggunakan kunci yang kamu tentukan sendiri untuk setiap nilai yang disimpan. Ini memudahkan dalam mengakses dan mengelola data pada aplikasi PHP.

Apa itu Associative Array?

Associative array adalah tipe data di PHP yang mengindeks nilai menggunakan kunci yang tidak harus berupa bilangan bulat. Kunci dalam associative array biasanya berupa string, yang memungkinkan kode menjadi lebih membaca dan memudahkan pencarian nilai berdasarkan penanda tertentu.

Contoh sederhana associative array:

```

$orang = array(
    "nama" => "Ayu",
    "umur" => 25,
    "kota" => "Jakarta"
);

```

```

hijau
Civic
Kunci: apel, Nilai: merah
Kunci: strawberry, Nilai: merah
Kunci: pisang, Nilai: kuning
Kunci: mangga, Nilai: kuning
4
Buah apel tersedia!

```

```

1 // Mengenal Array Multidimensi di PHP.php
2 <?php
3 $orang = array(
4     "nama" => "Ayu",
5     "umur" => 16,
6     "kota" => "aceh"
7 );
8 // Menggunakan fungsi array()
9 $buah = array(
10     "apel" => "hijau",
11     "strawberry" => "merah",
12     "pisang" => "kuning"
13 );
14 // Menetapkan langsung
15 $mobil["toyota"] = "Corolla";
16 $mobil["Honda"] = "Civic";
17 $mobil["Suzuki"] = "Swift";
18 echo $mobil["apel"]; // Output: hijau
19 echo $mobil["Honda"]; // Output: Civic
20 // Menambahkan data baru
21 $buah["mangga"] = "kuning";
22 // Mengubah data yang ada
23 $buah["apel"] = "merah";
24 // Mengubah data yang ada
25 $buah["apel"] = "merah";
26 foreach ($buah as $kunci => $nilai) {
27     echo "$kunci: $kunci, $Nilai<br>";
28 }
29 echo count($buah); // Menampilkan jumlah elemen di dalam array buah
30 if (array_key_exists("apel", $buah)) {
31     echo "<br>Buah apel tersedia!";
32 }
33 // Semua kunci
34 $kunci = array_keys($buah);
35 // Semua nilai
36 $Nilai = array_values($buah);
37 // Berdasarkan kunci
38 ksort($buah);
39 // Berdasarkan nilai
40 asort($buah);
41
42 associative_array.php
43
44 perlungan_forloop.php
45 membuat_tabel_html.php
46 array_multi_dimensi.php
47 Array_1_Dimensi_PHP.php
48 array_2_dimensi_pada_PHP...
49 array_3_dimensi_PHP.php
50 menghubungkan_Array_Ar...
51 buku.php
52 membuat_array.php
53 Mengenal_Array_Multidime...
54 ASOCIATIVE_ARRAY_U...
55 PROJECT_ALGORITMA_4-HAFIS...
56 PROJECT_ALGORITMA_4-HAFIS...

```

Cara Membuat Associative Array

Untuk membuat associative array, kamu bisa menggunakan fungsi array() dengan pasangan kunci dan nilai, atau dengan cara langsung menetapkan nilai ke kunci dalam array yang sudah ada.

```
// Menggunakan fungsi array()
```

```

$buah = array(
    "apel" => "hijau",
    "strawberry" => "merah",
    "pisang" => "kuning"
);

```

```
// Menetapkan langsung
```

```
$mobil['Toyota'] = "Corolla";
$mobil['Honda'] = "Civic";
$mobil['Suzuki'] = "Swift";
```

Mengakses Data dari Associative Array

Untuk mendapatkan nilai dari associative array, kamu bisa menggunakan kunci yang bersangkutan.

```
echo $buah['apel']; // Output: hijau
echo $mobil['Honda']; // Output: Civic
```

Menambahkan dan Mengubah Data

Associative array fleksibel, kamu bisa menambahkan atau mengubah data dengan mudah.

```
// Menambahkan data baru
$buah["mangga"] = "kuning";

// Mengubah data yang ada
$buah["apel"] = "merah";
```

Looping Melalui Associative Array

Kamu bisa melakukan looping melalui associative array dengan menggunakan foreach. Ini memungkinkan kamu memproses setiap pasangan kunci-nilai.

```
foreach ($buah as $kunci => $nilai) {
    echo "Kunci: $kunci, Nilai: $nilai<br>";
}
```

Beberapa Fungsi Bermanfaat untuk Associative Arrays

PHP menyediakan berbagai fungsi yang dapat membantu kamu dalam mengelola associative arrays.

Menghitung Elemen

```
echo count($buah); // Menampilkan jumlah elemen di dalam array buah
```

Memeriksa Kunci Tertentu

```
if (array_key_exists("apel", $buah)) {
    echo "Buah apel tersedia!";
}
```

Mendapatkan Semua Kunci atau Nilai

```
// Semua kunci
$kunci = array_keys($buah);

// Semua nilai
$nilai = array_values($buah);
```

Mengurutkan Associative Array

Kamu dapat mengurutkan associative array baik berdasarkan kunci maupun nilai.

```
// Berdasarkan kunci
ksort($buah);

// Berdasarkan nilai
asort($buah);
```

Associative array di PHP sangat berguna untuk mengelola data yang membutuhkan identifikasi yang jelas antara kunci dan nilai. Dengan fitur ini, PHP memudahkan pengembang untuk menulis kode yang rapi dan mudah dipelihara.

METODE METODE ARRAY

Pelajari berbagai metode array pada PHP yang paling umum digunakan, termasuk contoh penggunaan dan penjelasan sederhana untuk pemula.

PHP adalah bahasa pemrograman server-side yang populer dan sering digunakan untuk pengembangan web. Salah satu fitur yang kuat dalam PHP adalah manipulasi array. Array adalah struktur data yang memungkinkan kamu menyimpan banyak nilai dalam satu variabel. Berikut ini adalah beberapa metode array yang sering digunakan dalam PHP untuk mempermudah pengelolaan data.

array_push()

Metode array_push() digunakan untuk menambahkan satu atau lebih elemen ke akhir array. Elemen yang ditambahkan akan menjadi elemen terakhir dalam array.

```
$buah = ["apel", "jeruk"];
array_push($buah, "pisang", "mangga");
print_r($buah);
```

Hasilnya akan menjadi:

Array

(

```
[0] => apel
[1] => jeruk
```

- [2] => pisang
[3] => mangga

```

Array
(
    [0] => apel
    [1] => jeruk
    [2] => pisang
    [3] => mangga
)
By HAFIS RAMADHAN

```

```

1 <?php
2 echo "<p>";
3 $buah = ["apel", "jeruk"];
4 array_push($buah, "pisang", "mangga");
5 print_r($buah);
6 echo "</p>";
7
8 echo "<p><strong><i>By HAFIS RAMADHAN</strong></i></p>";

```

array_pop()

array_pop() digunakan untuk menghapus elemen terakhir dari array. Fungsi ini akan mengurangi panjang array dan mengembalikan nilai yang dihapus.

```
$buah = ["apel", "jeruk", "pisang"];
$buahTerakhir = array_pop($buah);
echo $buahTerakhir;
```

Output:

Pisang

```

pisang
By HAFIS RAMADHAN

```

```

1 <?php
2 $buah = ["apel", "jeruk", "pisang"];
3 $buahTerakhir = array_pop($buah);
4 echo $buahTerakhir;
5
6 echo "<p><strong><i>By HAFIS RAMADHAN</strong></i></p>";

```

array_shift()

Fungsi array_shift() akan menghapus elemen pertama dari array dan menggeser semua elemen berikutnya ke posisi sebelumnya, sehingga indeks dari setiap elemen akan berkurang satu.

```
$buah = ["apel", "jeruk", "pisang"];
$buahPertama = array_shift($buah);
echo $buahPertama;
```

Output:

Apel

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/56%20array%20shift%20.php
apel
By HAFIS RAMADHAN

EXPLORER
ALGORITMA 4
  20 penggunaan dasar do while...
  21 Perbedaan dengan while...
  22 memahami kondisi loop.php
  23 Penggunaan Foreach Loop...
  24 foreach dengan mendapat...
  25 Penanganan Array Multidi...
  26 penggunaan break.php
  27 penggunaan continue.php
  28 break pada Switch.php
  29 break dan continue.php
  30 Array.php
  31 array di php.php
  32 menampilkan isi array.php
  33 perulangan while dan fore...
  34 menghapus isi array.php
  35 menambahkan isi array.php
  36 menambahkan pada indeks...
  37 array assosiatif.php
  38 mengassosikan sebuah k...
  39 array indexed.php
  40 penulisan forech pada index...
  41 penggunaan pada indexed ...
  42 associative array.php
  43 perulangan forech.php
  44 membuat tabel html.php
  45 array multi dimensi.php
  46 Array 1 Dimensi PHP.php
  47 array 2 dimensi pada PHP ...
  48 array 3 dimensi PHP.php
  49 Menggabungkan Array dan...
  50 Kasus Array Assosiatif.php
  51 membuat array.php
  52 Mengenal Array Multidime...
  53 ASSOCIATIVE ARRAY.php
  54 array push.php
  55 array pop.php
  56 array shift.php
  57 array unshift.php
  PROJECT ALGORITMA 4-AHM...
  PROJECT ALGORITMA 4-HAFIS...

```

array_unshift()

Kebalikan dari array_shift(), array_unshift() menambahkan satu atau lebih elemen ke awal array dan mengembalikan jumlah elemen baru dalam array.

```

$buah = ["apel", "jeruk"];
array_unshift($buah, "pisang", "mangga");
print_r($buah);

```

Output:

Array

```

(
    [0] => pisang
    [1] => mangga
    [2] => apel
    [3] => jeruk
)

```

```

localhost/hafiz%20X%20rp%202/ALGORITMA%204/57%20array%20unshift%20.php
By HAFIS RAMADHAN

EXPLORER
ALGORITMA 4
  21 Perbedaan dengan while...
  22 memahami kondisi loop.php
  23 Penggunaan Foreach Loop...
  24 foreach dengan mendapat...
  25 Penanganan Array Multidi...
  26 penggunaan break.php
  27 penggunaan continue.php
  28 break pada Switch.php
  29 break dan continue.php
  30 Array.php
  31 array di php.php
  32 menampilkan isi array.php
  33 perulangan while dan fore...
  34 menghapus isi array.php
  35 menambahkan isi array.php
  36 menambahkan pada indeks...
  37 array assosiatif.php
  38 mengassosikan sebuah k...
  39 array indexed.php
  40 penulisan forech pada index...
  41 penggunaan pada indexed ...
  42 associative array.php
  43 perulangan forech.php
  44 membuat tabel html.php
  45 array multi dimensi.php
  46 Array 1 Dimensi PHP.php
  47 array 2 dimensi pada PHP ...
  48 array 3 dimensi PHP.php
  49 Menggabungkan Array dan...
  50 Kasus Array Assosiatif.php
  51 membuat array.php
  52 Mengenal Array Multidime...
  53 ASSOCIATIVE ARRAY.php
  54 array push.php
  55 array pop.php
  56 array shift.php
  57 array unshift.php
  58 array unshift.php U
  PROJECT ALGORITMA 4-AHM...
  PROJECT ALGORITMA 4-HAFIS...

```

array_merge()

Fungsi array_merge() menggabungkan dua atau lebih array menjadi satu array. Jika ada elemen dengan kunci yang sama, elemen dari array terakhir yang disediakan akan mengambil alih.

```

$array1 = ["warna" => "merah", 2, 4];
$array2 = ["a", "b", "warna" => "hijau", "bentuk" => "trapesium", 4];
$result = array_merge($array1, $array2);
print_r($result);

```

Output:

Array

```

(

```

[warna] => hijau

[0] => 2

[1] => 4

[2] => a

[3] => b

[bentuk] => trapesium

[4] => 4

The screenshot shows a code editor with an open file named '58 array merge.php'. The code demonstrates how to merge two arrays in PHP. It starts by defining two arrays: '\$array1' with elements 'warna' (value 'merah') at index 2 and 'bentuk' (value 'trapesium') at index 4, and '\$array2' with elements 'a' at index 0 and 'b' at index 1. It then merges them using the 'array_merge()' function and prints the result with 'print_r()'. The output is displayed in a browser window as follows:

```
Array
(
    [0] => hijau
    [1] => 4
    [2] => a
    [3] => b
    [4] => trapesium
)
By HAFIS RAMADHAN
```

The code editor's sidebar lists various PHP topics, including array manipulation like 'array shift', 'array unshift', and 'array merge'.

array_slice()

array_slice() digunakan untuk mendapatkan sebuah sub-array dari array. Kamu bisa menentukan indeks awal dan panjang sub-array yang diinginkan.

```
$buah = ["apel", "jeruk", "pisang", "mangga"];
$subBuah = array_slice($buah, 1, 2);
print_r($subBuah);
```

Output:

Array

(

[0] => jeruk

[1] => pisang

)

The screenshot shows a code editor with an open file named '59 array slice.php'. The code demonstrates how to use the 'array_slice()' function to get a sub-array from the '\$buah' array starting at index 1 and ending at index 2. The output is displayed in a browser window as follows:

```
Array
(
    [0] => jeruk
    [1] => pisang
)
by hafidz ramadhan
```

The code editor's sidebar lists various PHP topics, including array manipulation like 'array shift', 'array unshift', and 'array merge'.

array_splice()

Fungsi array_splice() menghapus elemen tertentu dari array dan bisa juga menggantikannya dengan nilai lain. Fungsi ini sangat berguna untuk manipulasi array yang kompleks.

```
$buah = ["apel", "jeruk", "pisang", "mangga"];
array_splice($buah, 2, 1, "kiwi");
```

```
print_r($buah);
```

Output:

```
Array
(
    [0] => apel
    [1] => jeruk
    [2] => kiwi
    [3] => mangga
)
```

The screenshot shows a code editor interface with several tabs at the top: 'shift.php', 'array merge.php', 'array slice.php', 'array splice.php', and others. The main area displays the following PHP code:

```
<?php
$buah = ["apel", "jeruk", "kiwi", "mangga"];
echo "<p><strong>i>by hafidz ramadhan</strong></p>";
print_r($buah);
echo "<p><strong>i>by hafidz ramadhan</strong></p>";
```

The code defines an array '\$buah' with four elements: 'apel', 'jeruk', 'kiwi', and 'mangga'. It then prints the array using 'print_r()' and outputs a strong text 'by hafidz ramadhan'.

Itulah beberapa metode array pada PHP yang paling sering digunakan. Dengan memahami metode-metode ini, kamu akan lebih mudah untuk melakukan operasi terhadap array saat mengembangkan aplikasi web dengan PHP.

CRUD ARRAY

Panduan untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) pada array dalam PHP dengan langkah-langkah sederhana dan contoh kode yang mudah dipahami.

CRUD adalah singkatan dari Create, Read, Update, dan Delete. Operasi ini sangat dasar dalam pemrograman karena melibatkan manipulasi data. Dalam PHP, CRUD sering digunakan untuk mengelola array. Array adalah struktur data yang bisa menyimpan banyak nilai sekaligus. Berikut adalah cara melakukan operasi CRUD pada array dalam PHP.

Membuat Array (Create)

Untuk membuat array, kamu bisa menggunakan fungsi array() atau menggunakan square brackets []. Berikut contoh cara membuat array:

```
$buah = array("apel", "mangga", "pisang");
// atau
$buah = ["apel", "mangga", "pisang"];
```

Kamu juga bisa membuat array asosiatif, di mana setiap elemen mempunyai key sebagai identifiernya:

```
$sumur = array("Ali" => 20, "Budi" => 25, "Citra" => 22);
```

Membaca Array (Read)

Untuk membaca atau mengakses isi dari array, kamu bisa menggunakan key atau index dari elemen tersebut. Pada array numerik, index dimulai dari 0.

```
echo $buah[0]; // menampilkan "apel"
echo $sumur["Ali"]; // menampilkan 20
```

Untuk membaca seluruh isi array, loop seperti foreach bisa digunakan:

```
foreach ($buah as $b) {
    echo $b . " "; // menampilkan "apel mangga pisang "
```

{}

Memperbarui Array (Update)

Untuk mengubah nilai pada array, kamu cukup menetapkan nilai baru ke key atau index tertentu.

```
$buah[0] = "jeruk"; // mengubah nilai elemen pertama menjadi "jeruk"
```

```
$sumur["Ali"] = 21; // mengubah umur Ali menjadi 21
```

Untuk menambah elemen baru ke dalam array, kamu bisa menggunakan fungsi array_push() atau langsung membuat key atau index baru.

```
array_push($buah, "durian"); // menambah "durian" ke dalam array $buah
```

```
$sumur["Dina"] = 18; // menambah key "Dina" dengan nilai 18 ke dalam array $sumur
```

Menghapus Array (Delete)

Untuk menghapus elemen dari array, kamu bisa menggunakan fungsi unset().

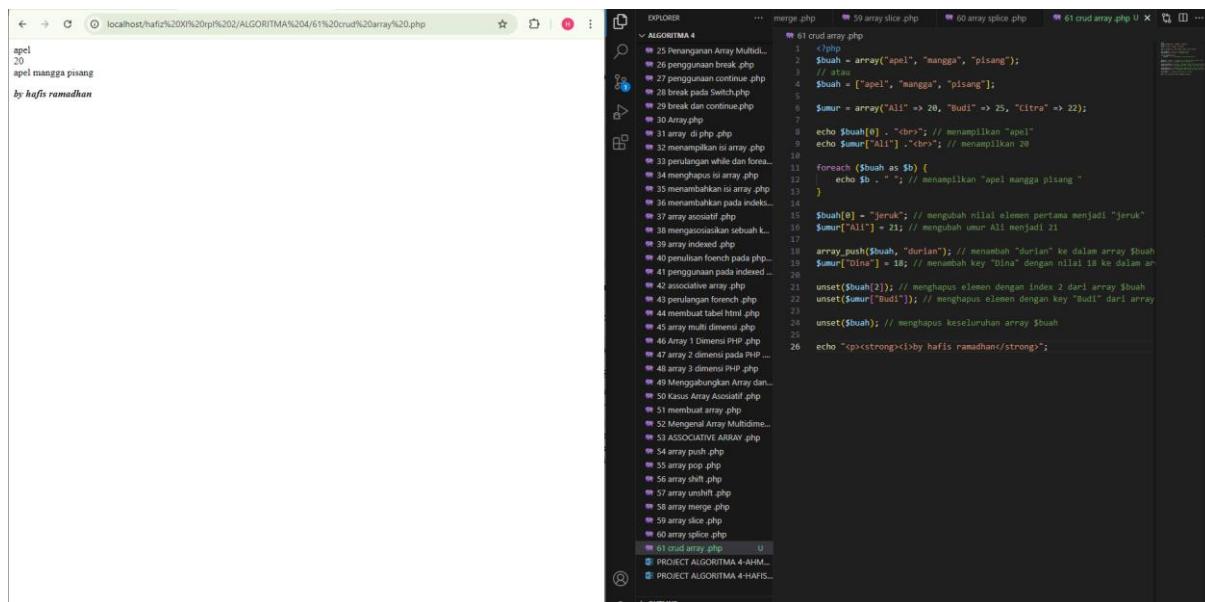
```
unset($buah[2]); // menghapus elemen dengan index 2 dari array $buah
```

```
unset($sumur["Budi"]); // menghapus elemen dengan key "Budi" dari array $sumur
```

Setelah elemen dihapus, index atau key tidak akan digunakan kembali. Jika kamu ingin menghapus semua elemen array, cukup unset() seluruh array.

```
unset($buah); // menghapus keseluruhan array $buah
```

Melakukan operasi CRUD pada array di PHP cukup mudah dan intuitif. Dengan mengikuti langkah-langkah di atas, kamu bisa melakukan manipulasi data pada array dengan lancar. Ingatlah untuk selalu menggunakan struktur dan fungsi yang tepat untuk memenuhi kebutuhan programmu.



```

1 <?php
2 $buah = array("apel", "mangga", "pisang");
3 // atau
4 $buah = ["apel", "mangga", "pisang"];
5
6 $sumur = array("Ali" => 20, "Budi" => 25, "Citra" => 22);
7 echo $buah[0]; // menampilkan "apel"
8 echo $sumur["Ali"]; // menampilkan 20
9
10 foreach ($buah as $b) {
11     echo $b . " "; // menampilkan "apel mangga pisang "
12 }
13
14 $buah[0] = "jeruk"; // mengubah nilai elemen pertama menjadi "jeruk"
15 $sumur["Ali"] = 21; // mengubah umur Ali menjadi 21
16
17 array_push($buah, "durian"); // menambah "durian" ke dalam array $buah
18 $sumur["Dina"] = 18; // menambah key "Dina" dengan nilai 18 ke dalam array $sumur
19
20 unset($buah[2]); // menghapus elemen dengan index 2 dari array $buah
21
22 unset($sumur["Budi"]); // menghapus elemen dengan key "Budi" dari array $sumur
23
24 unset($buah); // menghapus keseluruhan array $buah
25
26 echo "<p><strong>i</strong>by hafis ramadhan</p>";

```