

**LATIHAN PERTEMUAN 4**

**PEMROGRAMAN WEB**



**NAMA : LILIYAN PRAMUDITA**

**KELAS : SIB 2C**

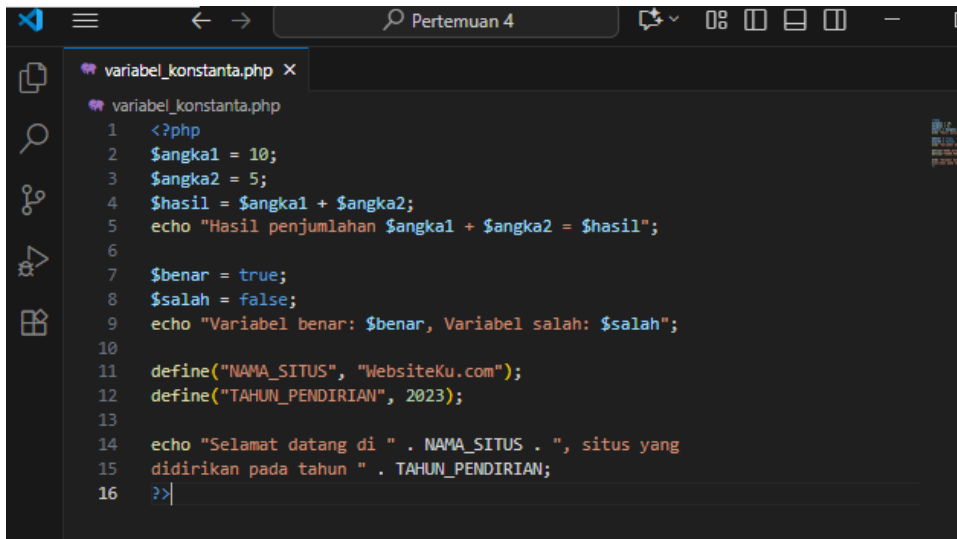
**NIM : 244107060096**

**TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2025**

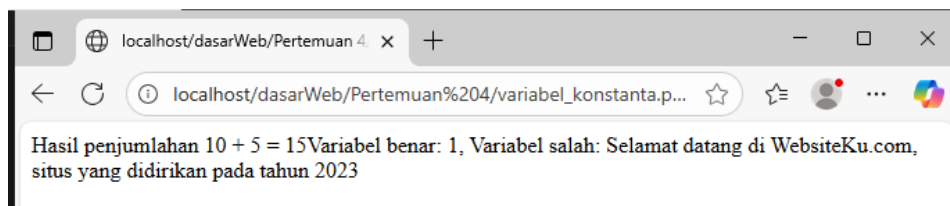
## Praktikum Bagian 1. Variabel dan Konstanta

1. Membuat file variabel\_konstanta.php
2. Menambahkan isi variabel\_konstanta.php



```
1 <?php
2 $angka1 = 10;
3 $angka2 = 5;
4 $hasil = $angka1 + $angka2;
5 echo "Hasil penjumlahan $angka1 + $angka2 = $hasil";
6
7 $benar = true;
8 $salah = false;
9 echo "Variabel benar: $benar, Variabel salah: $salah";
10
11 define("NAMA_SITUS", "WebsiteKu.com");
12 define("TAHUN_PENDIRIAN", 2023);
13
14 echo "Selamat datang di " . NAMA_SITUS . ", situs yang
15 didirikan pada tahun " . TAHUN_PENDIRIAN;
16 ?>
```

3. Hasil dari kode php




### (soal no 1)

Dari kode yang saya amati, variabel adalah tempat untuk menyimpan data yang bisa diubah, dan namanya selalu diawali dengan tanda \$. Sementara itu, konstanta adalah nilai yang sudah ditetapkan dan tidak bisa diubah, yang dibuat menggunakan define() dan ditulis dengan huruf besar. Contohnya, true pada tipe data boolean akan dicetak sebagai angka 1, sedangkan false tidak akan mencetak apa-apa.

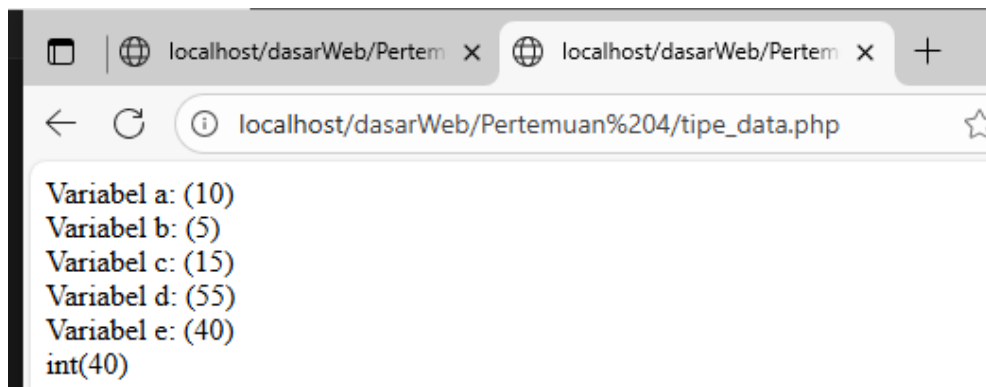
## Praktikum 2 : Penggunaan Tipe Data

1. Membuat file tipe\_data.php
2. Menambahkan isi variable



```
1 <?php
2 $a = 10;
3 $b = 5;
4 $c = $a + 5;
5 $d = $b + (10 * 5);
6 $e = $d - $c;
7
8 echo "Variabel a: ($a) <br>";
9 echo "Variabel b: ($b) <br>";
10 echo "Variabel c: ($c) <br>";
11 echo "Variabel d: ($d) <br>";
12 echo "Variabel e: ($e) <br>";
13
14 var_dump($e);
15 ?>
```

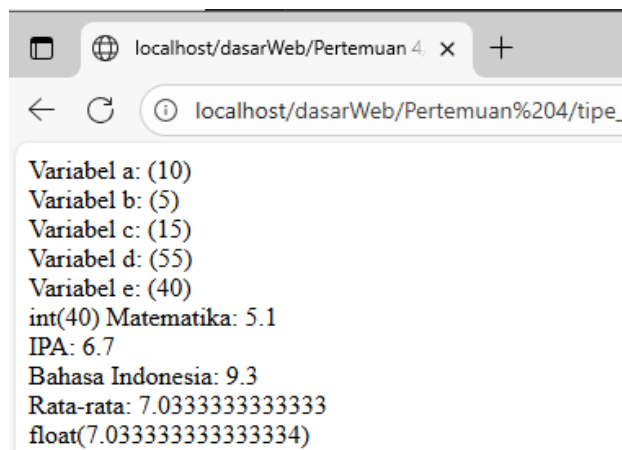
3. Hasil dari kode



4. Menambahkan isi kode

```
15 $nilaiMatematika = 5.1;
16 $nilaiIPA = 6.7;
17 $nilaiBahasaIndonesia = 9.3;
18
19 $rataRata = ($nilaiMatematika + $nilaiIPA + $nilaiBahasaIndonesia) / 3;
20
21 echo "Matematika: {$nilaiMatematika} <br>";
22 echo "IPA: {$nilaiIPA} <br>";
23 echo "Bahasa Indonesia: {$nilaiBahasaIndonesia} <br>";
24 echo "Rata-rata: $rataRata <br>";
25 var_dump($rataRata);
26 ?>
```

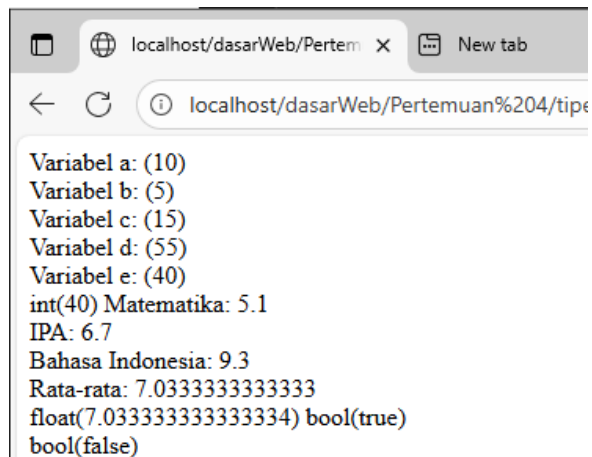
5. Hasil dari run



6. Menambahkan isi kode

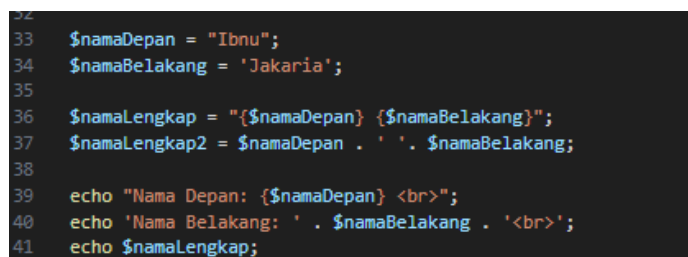
```
27 $apakahSiswaLulus = true;
28 $apakahSiswaSudahUjian = false;
29 var_dump($apakahSiswaLulus);
30 echo "<br>";
31 var_dump($apakahSiswaSudahUjian);
32 ?>
```

## 7. Hasil run



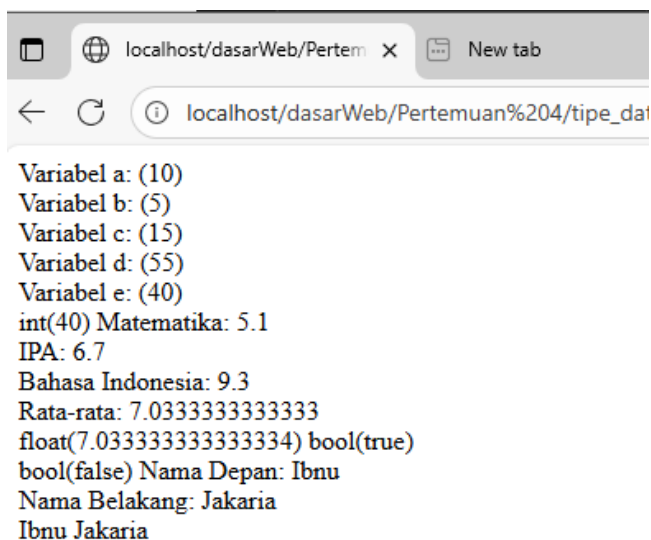
localhost/dasarWeb/Pertem x New tab  
localhost/dasarWeb/Pertemuan%204/tipe  
Variabel a: (10)  
Variabel b: (5)  
Variabel c: (15)  
Variabel d: (55)  
Variabel e: (40)  
int(40) Matematika: 5.1  
IPA: 6.7  
Bahasa Indonesia: 9.3  
Rata-rata: 7.0333333333333  
float(7.03333333333334) bool(true)  
bool(false)

## 8. Menambahkan isi kode



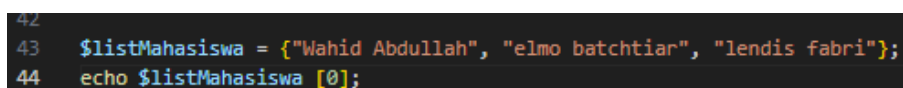
```
33 $namaDepan = "Ibnu";  
34 $namaBelakang = 'Jakaria';  
35  
36 $namaLengkap = "{$namaDepan} {$namaBelakang}";  
37 $namaLengkap2 = $namaDepan . ' ' . $namaBelakang;  
38  
39 echo "Nama Depan: {$namaDepan} <br>";  
40 echo 'Nama Belakang: ' . $namaBelakang . '<br>';  
41 echo $namaLengkap;
```

## 9. Hasil run



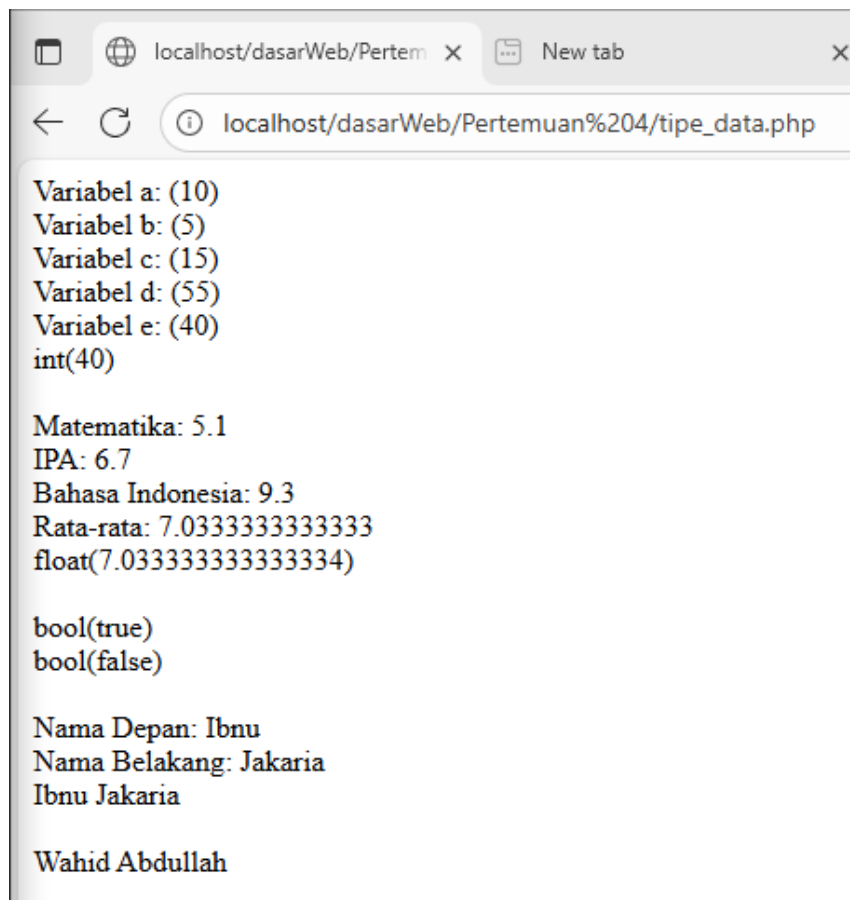
localhost/dasarWeb/Pertem x New tab  
localhost/dasarWeb/Pertemuan%204/tipe\_da  
Variabel a: (10)  
Variabel b: (5)  
Variabel c: (15)  
Variabel d: (55)  
Variabel e: (40)  
int(40) Matematika: 5.1  
IPA: 6.7  
Bahasa Indonesia: 9.3  
Rata-rata: 7.0333333333333  
float(7.03333333333334) bool(true)  
bool(false) Nama Depan: Ibnu  
Nama Belakang: Jakaria  
Ibnu Jakaria

## 10. Menambahkan isi kode dan merapikan



```
42  
43 $listMahasiswa = {"Wahid Abdullah", "elmo batchtiar", "lendis fabri"};  
44 echo $listMahasiswa [0];
```

## 11. Hasil run



```
Variabel a: (10)
Variabel b: (5)
Variabel c: (15)
Variabel d: (55)
Variabel e: (40)
int(40)

Matematika: 5.1
IPA: 6.7
Bahasa Indonesia: 9.3
Rata-rata: 7.0333333333333
float(7.033333333333334)

bool(true)
bool(false)

Nama Depan: Ibnu
Nama Belakang: Jakaria
Ibnu Jakaria

Wahid Abdullah
```

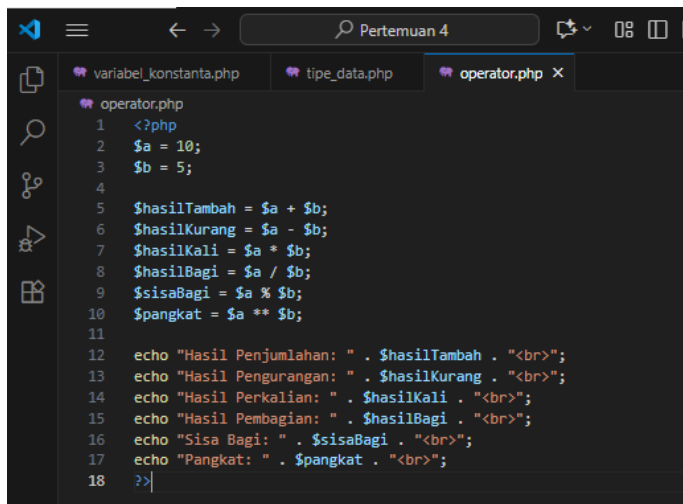
### (soal no 2)

Saya mengamati penggunaan beberapa tipe data dasar pada kode tersebut, yaitu:

- Integer: Tipe data ini digunakan untuk angka bulat. Contohnya \$a = 10 dan \$b = 5.
- Float: Tipe data ini digunakan untuk angka desimal, seperti nilai-nilai mata pelajaran \$nilaiMatematika = 5.1, \$nilaiIPA = 6.7, dan \$nilaiBahasaIndonesia = 9.3.
- Boolean: Tipe data ini hanya memiliki dua nilai, yaitu true (benar) atau false (salah). Contohnya \$apakahSiswaLulus dan \$apakahSiswaSudahUjian.
- String: Tipe data ini berisi teks. Teks bisa diapit dengan tanda kutip dua (") atau tanda kutip satu ('). Saya juga melihat ada cara untuk menggabungkan dua teks atau lebih, yaitu menggunakan tanda titik (.).
- Array: Tipe data ini bisa menyimpan banyak data dalam satu variabel. Saya melihat \$listMahasiswa yang menyimpan daftar nama. Untuk mengaksesnya, saya menggunakan indeks angka yang dimulai dari 0.

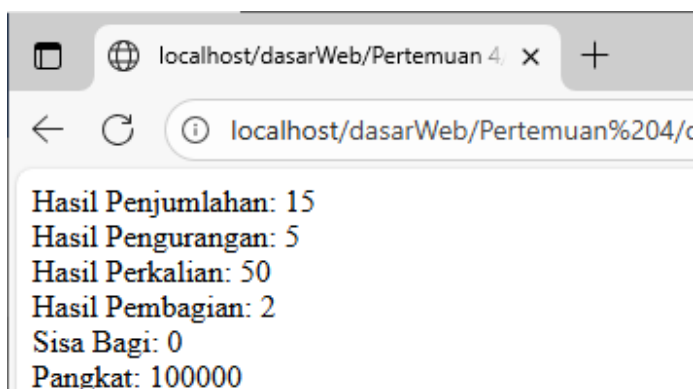
## Praktikum Bagian 3 : Penggunaan Operator PHP

1. Buat satu file baru bernama operator.php
2. melengkapi kode program sehingga bisa menampilkan hasilnya dan rapi



```
1 <?php
2 $a = 10;
3 $b = 5;
4
5 $hasilTambah = $a + $b;
6 $hasilKurang = $a - $b;
7 $hasilKali = $a * $b;
8 $hasilBagi = $a / $b;
9 $sisabagi = $a % $b;
10 $pangkat = $a ** $b;
11
12 echo "Hasil Penjumlahan: " . $hasilTambah . "<br>";
13 echo "Hasil Pengurangan: " . $hasilKurang . "<br>";
14 echo "Hasil Perkalian: " . $hasilKali . "<br>";
15 echo "Hasil Pembagian: " . $hasilBagi . "<br>";
16 echo "Sisa Bagi: " . $sisabagi . "<br>";
17 echo "Pangkat: " . $pangkat . "<br>";
18 >|
```

3. Hasil run



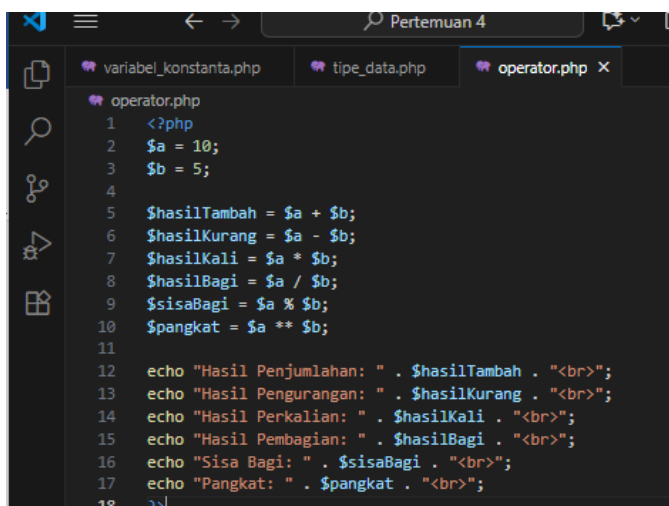
localhost/dasarWeb/Pertemuan 4

localhost/dasarWeb/Pertemuan%204/c

Hasil Penjumlahan: 15  
Hasil Pengurangan: 5  
Hasil Perkalian: 50  
Hasil Pembagian: 2  
Sisa Bagi: 0  
Pangkat: 100000

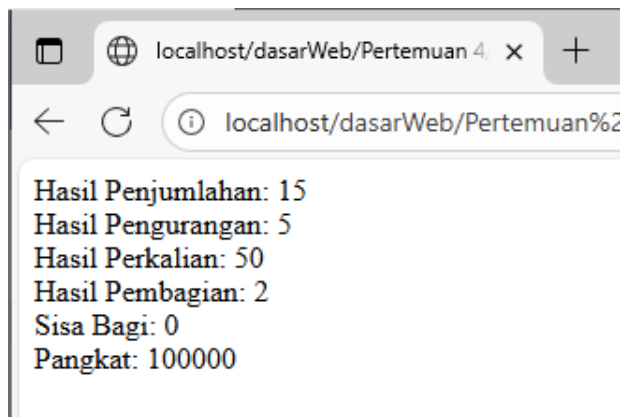
### (soal no 3.1)

Berikut adalah kode dan hasil dari penggunaan operator aritmatika. Kode ini akan mencetak hasil dari operasi penjumlahan, pengurangan, perkalian, pembagian, sisa bagi (modulo), dan pangkat.



```
1 <?php
2 $a = 10;
3 $b = 5;
4
5 $hasilTambah = $a + $b;
6 $hasilKurang = $a - $b;
7 $hasilKali = $a * $b;
8 $hasilBagi = $a / $b;
9 $sisabagi = $a % $b;
10 $pangkat = $a ** $b;
11
12 echo "Hasil Penjumlahan: " . $hasilTambah . "<br>";
13 echo "Hasil Pengurangan: " . $hasilKurang . "<br>";
14 echo "Hasil Perkalian: " . $hasilKali . "<br>";
15 echo "Hasil Pembagian: " . $hasilBagi . "<br>";
16 echo "Sisa Bagi: " . $sisabagi . "<br>";
17 echo "Pangkat: " . $pangkat . "<br>";
18 >|
```

Hasil run



### (soal no 3.2)

Berikut adalah kode dan hasil dari operator pembandingan. Operator ini menghasilkan nilai true (1) atau false (kosong).

```
20
21 $hasilSama = $a == $b;
22 $hasilTidakSama = $a != $b;
23 $hasilLebihKecil = $a < $b;
24 $hasilLebihBesar = $a > $b;
25 $hasilLebihKecilSama = $a <= $b;
26 $hasilLebihBesarSama = $a >= $b;
27
28 echo "Apakah " . $a . " sama dengan " . $b . "? " . $hasilSama . "<br>";
29 echo "Apakah " . $a . " tidak sama dengan " . $b . "? " . $hasilTidakSama . "<br>";
30 echo "Apakah " . $a . " lebih kecil dari " . $b . "? " . $hasilLebihKecil . "<br>";
31 echo "Apakah " . $a . " lebih besar dari " . $b . "? " . $hasilLebihBesar . "<br>";
32 echo "Apakah " . $a . " lebih kecil sama dengan " . $b . "? " . $hasilLebihKecilSama . "<br>";
33 echo "Apakah " . $a . " lebih besar sama dengan " . $b . "? " . $hasilLebihBesarSama . "<br>";
34
```

Hasil run

```
Apakah 10 sama dengan 5?
Apakah 10 tidak sama dengan 5? 1
Apakah 10 lebih kecil dari 5?
Apakah 10 lebih besar dari 5? 1
Apakah 10 lebih kecil sama dengan 5?
Apakah 10 lebih besar sama dengan 5? 1
```

### (soal no 3.3)

AND (&&): true. Karena \$a dan \$b sama-sama memiliki nilai (bukan 0 atau null), keduanya dianggap true. Jadi, true && true hasilnya true. OR (||): true. Karena \$a atau \$b salah satunya (bahkan keduanya) true, maka hasilnya true. NOT (!): false. Karena \$a dan \$b memiliki nilai (dianggap true), maka negasinya adalah false.

```
37 $hasilAnd = $a && $b;
38 $hasilOr = $a || $b;
39 $hasilNotA = !$a;
40 $hasilNotB = !$b;
41
42 echo "Apakah " . $a . " dan " . $b . " benar? " . $hasilAnd . "<br>";
43 echo "Apakah " . $a . " atau " . $b . " benar? " . $hasilOr . "<br>";
44 echo "Apakah " . $a . " tidak benar? " . $hasilNotA . "<br>";
45 echo "Apakah " . $b . " tidak benar? " . $hasilNotB . "<br>";
46
```

Hasil run

```
Apakah 10 dan 5 benar? 1
Apakah 10 atau 5 benar? 1
Apakah 10 tidak benar?
Apakah 5 tidak benar?
```

### (soal no 3.4)

Saya mengamati bahwa dengan operator penugasan, nilai variabel bisa diubah sekaligus dihitung dalam satu perintah. Misalnya, `$a += $b` artinya nilai `$a` akan ditambah dengan `$b` lalu disimpan kembali ke `$a`. Hal ini berlaku juga untuk pengurangan, perkalian, pembagian, dan sisa bagi. Dengan cara ini, kode menjadi lebih singkat dan efisien karena tidak perlu menulis `$a = $a + $b` secara panjang.

```
49 $awal = $a;
50 $a += $b;
51 echo "Hasil $awal += $b : $a <br>";
52
53 $awal = $a;
54 $a -= $b;
55 echo "Hasil $awal -= $b : $a <br>";
56
57 $awal = $a;
58 $a *= $b;
59 echo "Hasil $awal *= $b : $a <br>";
60
61 $awal = $a;
62 $a /= $b;
63 echo "Hasil $awal /= $b : $a <br>";
64
65 $awal = $a;
66 $a %= $b;
67 echo "Hasil $awal %= $b : $a <br>";
68
```

Hasil run

```
Hasil 10 += 5 : 15
Hasil 15 -= 5 : 10
Hasil 10 *= 5 : 50
Hasil 50 /= 5 : 10
Hasil 10 %= 5 : 0
```



### (soal no 3.5)

Pada bagian ini, saya membandingkan nilai dan tipe data dari dua variabel menggunakan operator identik `===` dan `!==`. Contoh ini membandingkan integer dan string yang memiliki nilai yang sama, sehingga hasilnya menunjukkan bahwa mereka tidak identik.

```
70
71 $hasilIdentik = $a === $b;
72 $hasilTidakIdentik = $a !== $b;
73
74 echo "Hasil Identik: " . $hasilIdentik . "<br>";
75 echo "Hasil Tidak Identik: " . $hasilTidakIdentik . "<br><br>";
76 ?>
```

Hasil run

```
Hasil Identik:
Hasil Tidak Identik: 1
```

### (soal no 3.6)

Pada kode ini saya mendeklarasikan variabel `$totalKursi` bernilai 45 sebagai jumlah kursi di restoran dan `$terisi` bernilai 28 untuk kursi yang sudah ditempati, kemudian saya hitung kursi kosong dengan rumus `$kosong = $totalKursi - $terisi` sehingga hasilnya 17. Setelah itu, untuk mencari persentase kursi kosong saya gunakan rumus  $(\$kosong / \$totalKursi) * 100$  yang disimpan di variabel `$persentaseKosong`. Terakhir, saya menampilkan semua informasi dengan `echo`, yaitu jumlah kursi, kursi terisi, kursi kosong, dan persentase kursi kosong. Dengan cara ini, program langsung menghitung dan menampilkan hasil yang jelas di browser.

```
78
79 $totalKursi = 45;
80 $terisi = 28;
81 $kosong = $totalKursi - $terisi;
82
83 $persentaseKosong = ($kosong / $totalKursi) * 100;
84
85 echo "Total kursi: $totalKursi <br>";
86 echo "Kursi terisi: $terisi <br>";
87 echo "Kursi kosong: $kosong <br>";
88 echo "Persentase kursi kosong: $persentaseKosong %<br>";
89 ?>
```

Hasil run

```
Total kursi: 45
Kursi terisi: 28
Kursi kosong: 17
Persentase kursi kosong: 37.777777777778 %
```

## Praktikum Bagian 4 : Penggunaan Struktur Kontrol Pada PHP

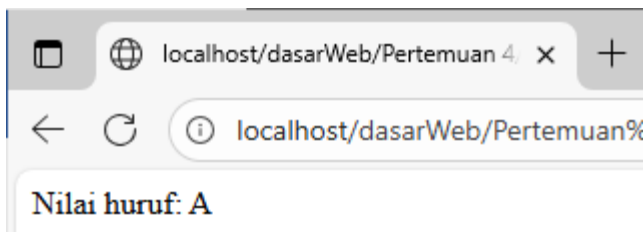
### (soal no 4.1)

Pada bagian ini, saya menggunakan struktur kontrol if, elseif, dan else untuk menentukan nilai huruf berdasarkan skor numerik. Program akan mengecek kondisi dari atas ke bawah dan akan berhenti pada kondisi yang pertama kali bernilai true.

1. Menambahkan isi variable

```
struktur_kontrol.php
1 <?php
2 $nilaiNumerik = 92;
3
4 if ($nilaiNumerik >= 90 && $nilaiNumerik <= 100) {
5     echo "Nilai huruf: A<br>";
6 } elseif ($nilaiNumerik >= 80 && $nilaiNumerik < 90) {
7     echo "Nilai huruf: B<br>";
8 } elseif ($nilaiNumerik >= 70 && $nilaiNumerik < 80) {
9     echo "Nilai huruf: C<br>";
10 } else {
11     echo "Nilai huruf: D<br>";
12 }
13 ?>
```

2. Hasil run



localhost/dasarWeb/Pertemuan 4

Nilai huruf: A

### (soal no 4.2)

Saya menggunakan perulangan while untuk menghitung jumlah hari yang dibutuhkan seorang atlet untuk mencapai target jarak. Perulangan ini akan terus berjalan selama kondisi yang ditentukan masih terpenuhi.

1. Menambahkan isi variable

```
16 $jarakSaatIni = 0;
17 $jarakTarget = 500;
18 $peningkatanHarian = 30;
19 $hari = 0;
20
21 while ($jarakSaatIni < $jarakTarget) {
22     $jarakSaatIni += $peningkatanHarian;
23     $hari++;
24 }
25
26 echo "Atlet tersebut memerlukan $hari hari untuk
27 mencapai jarak 500 kilometer.<br><br>";
28 ?>
```

2. Hasil run

```
Atlet tersebut memerlukan 17 hari untuk mencapai jarak 500 kilometer.
```

**(soal no 4.3)**

Pada bagian ini, saya menggunakan perulangan for untuk menghitung total buah yang akan dipanen. Perulangan for digunakan ketika kita sudah tahu berapa kali perulangan tersebut harus dijalankan.

1. Menambahkan isi variable

```
31 $jumlahLahan = 10;  
32 $tanamanPerLahan = 5;  
33 $buahPerTanaman = 10;  
34 $jumlahBuah = 0;  
35  
36 for ($i = 1; $i <= $jumlahLahan; $i++) {  
37     $jumlahBuah += ($tanamanPerLahan * $buahPerTanaman);  
38 }  
39  
40 echo "Jumlah buah yang akan dipanen adalah: " . $jumlahBuah . "<br><br>";  
41
```

2. Hasil run

```
Jumlah buah yang akan dipanen adalah: 500
```

**(soal no 4.4)**

Saya menggunakan perulangan foreach untuk menghitung total skor dari sebuah array. Perulangan ini sangat efisien untuk mengiterasi setiap elemen dalam array tanpa perlu mengkhawatirkan indeksnya.

1. Menambahkan Isi Variable

```
42 $skorUjian = [85, 92, 78, 96, 88];  
43 $totalSkor = 0;  
44  
45 foreach ($skorUjian as $skor) {  
46     $totalSkor += $skor;  
47 }  
48  
49 echo "Total skor ujian adalah: " . $totalSkor . "<br><br>";  
50 ?>
```

2. Hasil run

```
Total skor ujian adalah: 439
```

### (soal no 4.5)

Pada kode ini, saya mengamati penggunaan perulangan foreach yang dikombinasikan dengan struktur kontrol if untuk memfilter dan mengklasifikasikan data dalam sebuah array. Perulangan foreach bertugas untuk memeriksa setiap nilai siswa satu per satu. Di dalamnya, pernyataan if mengecek apakah nilai kurang dari 60. Jika ya, program akan mencetak status "Tidak Lulus" dan pernyataan continue dieksekusi, yang menyebabkan perulangan langsung melompat ke iterasi berikutnya tanpa menjalankan kode di bawahnya. Jika nilai siswa 60 atau lebih, kondisi if tidak terpenuhi, dan program akan menjalankan baris berikutnya untuk mencetak status "Lulus"

1. Menambahkan isi variable

```
50
51 $nilaiSiswa = [85, 92, 58, 64, 90, 55, 88, 79, 70, 96];
52
53 foreach ($nilaiSiswa as $nilai) {
54     if ($nilai < 60) {
55         echo "Nilai: $nilai (Tidak Lulus) <br>";
56         continue;
57     }
58     echo "Nilai: $nilai (Lulus) <br>";
59 }
```

2. Hasil run

```
Nilai: 85 (Lulus)
Nilai: 92 (Lulus)
Nilai: 58 (Tidak Lulus)
Nilai: 64 (Lulus)
Nilai: 90 (Lulus)
Nilai: 55 (Tidak Lulus)
Nilai: 88 (Lulus)
Nilai: 79 (Lulus)
Nilai: 70 (Lulus)
Nilai: 96 (Lulus)
```

### (soal no 4.6)

Untuk menyelesaikan soal cerita ini, saya pertama-tama menyimpan semua nilai siswa ke dalam sebuah array. Kemudian, saya menggunakan fungsi sort() untuk mengurutkan nilai-nilai tersebut dari yang terendah ke yang tertinggi, hal ini mempermudah proses identifikasi nilai ekstrem. Setelah array terurut, saya menggunakan fungsi array\_slice() untuk membuang dua nilai terendah (elemen pertama dan kedua) dan dua nilai tertinggi (dua elemen terakhir). Terakhir, sisa nilai yang ada di tengah saya jumlahkan menggunakan fungsi array\_sum() untuk mendapatkan total nilai akhir yang akan digunakan

```
61 $nilaiSiswa = [85, 92, 78, 64, 90, 75, 88, 79, 70, 96];
62
63 sort($nilaiSiswa);
64
65 $nilaiYangDigunakan = array_slice($nilaiSiswa, 2, count($nilaiSiswa) - 4);
66
67 $totalNilai = array_sum($nilaiYangDigunakan);
68
69 echo "Daftar nilai siswa: " . implode(', ', $nilaiSiswa) . "<br>";
70 echo "Setelah mengabaikan dua nilai terendah dan dua nilai tertinggi,
71 nilai yang digunakan adalah: " . implode(', ', $nilaiYangDigunakan) . "<br>";
72 echo "Total nilai yang akan digunakan untuk rata-rata adalah: $totalNilai";
73 ?>
```

Hasil run

```
Daftar nilai siswa: 64, 70, 75, 78, 79, 85, 88, 90, 92, 96
Setelah mengabaikan dua nilai terendah dan dua nilai tertinggi, nilai yang digunakan adalah: 75, 78,
79, 85, 88, 90
Total nilai yang akan digunakan untuk rata-rata adalah: 495
```

#### (soal no 4.7)

Dalam penyelesaian soal ini, saya mendefinisikan variabel untuk harga produk dan ambang batas diskon. Saya menggunakan struktur kontrol if untuk memeriksa apakah harga produk (Rp 120.000) lebih besar dari ambang batas untuk mendapatkan diskon (Rp 100.000). Karena kondisi ini terpenuhi, maka blok kode di dalam if dieksekusi: program menghitung besaran diskon 20% dari harga asli, kemudian menghitung harga akhir dengan cara mengurangi harga asli dengan jumlah diskon tersebut. Hasilnya kemudian ditampilkan sebagai harga yang harus dibayar oleh pelanggan

```
77  $hargaProduk = 120000;
78  $ambangDiskon = 100000;
79  $persenDiskon = 0.20;
80  $hargaYangHarusDibayar = $hargaProduk;
81
82  if ($hargaProduk > $ambangDiskon) {
83      $diskon = $hargaProduk * $persenDiskon;
84      $hargaYangHarusDibayar = $hargaProduk - $diskon;
85      echo "Harga produk: Rp " . number_format($hargaProduk) . "<br>";
86      echo "Anda mendapatkan diskon sebesar 20% (Rp " . number_format($diskon) .
87  } else {
88      echo "Harga produk: Rp " . number_format($hargaProduk) . "<br>";
89      echo "Anda tidak mendapatkan diskon.<br>";
90  }
91
92  echo "Harga yang harus dibayar adalah: Rp " . number_format($hargaYangHarusDiba
93
94
95  ?>
```

Hasil run

```
Harga produk: Rp 120,000
Anda mendapatkan diskon sebesar 20% (Rp 24,000).
Harga yang harus dibayar adalah: Rp 96,000
```

### (soal no 4.8)

Sesuai dengan permintaan soal, saya menerapkan operator ternary untuk menyelesaikan logika kondisional secara ringkas. Pertama, saya menetapkan skor pemain dalam sebuah variabel. Kemudian, saya menampilkan total skor tersebut. Untuk menentukan apakah pemain mendapat hadiah, saya menggunakan ekspresi ternary ( $\$skorPemain > 500$ ) ? "YA" : "TIDAK". Ekspresi ini secara singkat memeriksa jika skor lebih dari 500; jika kondisi benar (true), ia akan mengembalikan nilai "YA", dan jika salah (false), ia akan mengembalikan "TIDAK". Hasil dari operator ternary ini saya simpan dalam variabel  $\$dapatHadiah$  dan menampilkannya pada baris kedua output.

```
96
97 $skorPemain = 650;
98
99 $dapatHadiah = ($skorPemain > 500) ? "YA" : "TIDAK";
100
101 echo "Total skor pemain adalah: $skorPemain<br>";
102 echo "Apakah pemain mendapatkan hadiah tambahan? $dapatHadiah";
```

Hasil run

```
Total skor pemain adalah: 650
Apakah pemain mendapatkan hadiah tambahan? YA
```

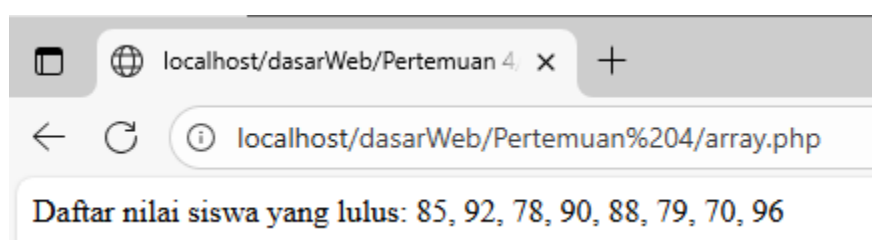
## Praktikum Bagian 5 : Penggunaan Array Pada PHP

### (soal no 5.1)

Pada kode ini, saya mengamati cara kerja perulangan foreach untuk menyaring elemen dari sebuah array numerik bernama  $\$nilaiSiswa$ . Saya membuat sebuah array kosong bernama  $\$nilaiLulus$  untuk menampung nilai-nilai yang memenuhi kriteria kelulusan. Di dalam perulangan foreach, saya menggunakan struktur if untuk memeriksa apakah setiap  $\$nilai$  lebih besar atau sama dengan 70. Jika kondisi tersebut terpenuhi, nilai tersebut akan ditambahkan ke dalam array  $\$nilaiLulus$ . Setelah perulangan selesai, saya menggunakan fungsi `implode()` untuk menggabungkan semua elemen dalam array  $\$nilaiLulus$  menjadi sebuah string yang dipisahkan oleh koma dan spasi, lalu menampilkannya

```
array.php
1 <?php
2 $nilaiSiswa = [85, 92, 78, 64, 90, 55, 88, 79, 70, 96];
3 $nilaiLulus = [];
4
5 foreach ($nilaiSiswa as $nilai) {
6     if ($nilai >= 70) {
7         $nilaiLulus[] = $nilai;
8     }
9 }
10
11 echo "Daftar nilai siswa yang lulus: " . implode(', ', $nilaiLulus);
12 >>
```

Hasil run



```
localhost/dasarWeb/Pertemuan 4 x +
localhost/dasarWeb/Pertemuan%204/array.php
Daftar nilai siswa yang lulus: 85, 92, 78, 90, 88, 79, 70, 96
```

### (soal no 5.2)

Untuk soal ini, saya mengamati penggunaan array dua dimensi untuk menyimpan data karyawan yang terdiri dari nama dan lama pengalaman kerja. Saya menginisialisasi sebuah array kosong `$karyawanPengalamanLimaTahun` untuk menyimpan nama-nama karyawan yang memenuhi syarat. Dengan menggunakan perulangan `foreach`, saya memeriksa setiap sub-array (`$karyawan`) di dalam `$daftarKaryawan`. Di dalam loop, saya menggunakan kondisi `if` untuk mengecek apakah nilai pada indeks ke-1 (pengalaman kerja) lebih dari 5. Jika benar, nama karyawan yang berada pada indeks ke-0 ditambahkan ke dalam array hasil. Terakhir, saya menampilkan daftar nama karyawan yang lolos seleksi dengan menggabungkan elemen-elemen array menggunakan `implode()`.

```
15 $daftarKaryawan = [
16     ['Alice', 7],
17     ['Bob', 3],
18     ['Charlie', 9],
19     ['David', 5],
20     ['Eva', 6],
21 ];
22
23 $karyawanPengalamanLimaTahun = [];
24
25 foreach ($daftarKaryawan as $karyawan) {
26     if ($karyawan[1] > 5) {
27         $karyawanPengalamanLimaTahun[] = $karyawan[0];
28     }
29 }
30
31 echo "Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: " .
32 implode(' ', $karyawanPengalamanLimaTahun);
```

Hasil run

```
Daftar karyawan dengan pengalaman kerja lebih dari 5 tahun: Alice, Charlie, Eva
```

### (soal no 5.3)

Pada bagian ini, saya mempelajari cara mengakses data dalam array asosiatif yang dikombinasikan dengan array numerik, membentuk sebuah array multidimensi. Array utama `$daftarNilai` menggunakan nama mata kuliah sebagai *key*. Saya menetapkan variabel `$mataKuliah` untuk memilih data 'Fisika' dari array. Kemudian, saya menggunakan `foreach` untuk melakukan iterasi pada `$daftarNilai[$mataKuliah]`, yang berisi daftar nilai mahasiswa untuk mata kuliah Fisika. Di setiap iterasi, saya mengakses nama mahasiswa melalui `$nilai[0]` dan nilai mereka melalui `$nilai[1]` untuk ditampilkan

```
36 $daftarNilai = [
37     'Matematika' => [
38         ['Alice', 85],
39         ['Bob', 92],
40         ['Charlie', 78],
41     ],
42     'Fisika' => [
43         ['Alice', 90],
44         ['Bob', 88],
45         ['Charlie', 75],
46     ],
47     'Kimia' => [
48         ['Alice', 92],
49         ['Bob', 80],
50         ['Charlie', 85],
51     ],
52 ];
53
54 $mataKuliah = 'Fisika';
55
56 echo "Daftar nilai mahasiswa dalam mata kuliah $mataKuliah: <br>";
57 foreach ($daftarNilai[$mataKuliah] as $nilai) {
58     echo "Nama: {$nilai[0]}, Nilai: {$nilai[1]} <br>";
59 }
```

hasil run

```
Daftar nilai mahasiswa dalam mata kuliah Fisika:  
Nama: Alice, Nilai: 90  
Nama: Bob, Nilai: 88  
Nama: Charlie, Nilai: 75
```

#### (soal no 5.4)

Untuk menyelesaikan soal cerita ini, pertama saya membuat sebuah array dua dimensi untuk menyimpan data nama dan nilai setiap siswa. Langkah selanjutnya, saya perlu menghitung nilai rata-rata kelas. Untuk itu, saya menjumlahkan semua nilai siswa lalu membaginya dengan jumlah total siswa. Setelah mendapatkan nilai rata-rata, saya melakukan perulangan lagi pada data siswa. Dalam perulangan ini, saya menggunakan if untuk membandingkan nilai setiap siswa dengan nilai rata-rata. Jika nilai siswa lebih tinggi dari rata-rata, saya menampilkan nama dan nilai siswa tersebut sebagai output

```
62  
63 $dataSiswa = [  
64     ['Alice', 85],  
65     ['Bob', 92],  
66     ['Charlie', 78],  
67     ['David', 64],  
68     ['Eva', 90]  
69 ];  
70  
71 $totalNilai = 0;  
72 foreach ($dataSiswa as $siswa) {  
73     $totalNilai += $siswa[1];  
74 }  
75  
76 $rataRata = $totalNilai / count($dataSiswa);  
77  
78 echo "Nilai rata-rata kelas adalah: " . number_format($rataRata, 2) . "<br>";  
79 echo "Daftar siswa dengan nilai di atas rata-rata: <br>";  
80  
81 foreach ($dataSiswa as $siswa) {  
82     if ($siswa[1] > $rataRata) {  
83         echo "Nama: " . $siswa[0] . ", Nilai: " . $siswa[1] . "<br>";  
84     }  
85 }
```

Hasil run

```
Nilai rata-rata kelas adalah: 81.80  
Daftar siswa dengan nilai di atas rata-rata:  
Nama: Alice, Nilai: 85  
Nama: Bob, Nilai: 92  
Nama: Eva, Nilai: 90
```