

## **BAB 2**

### **OPERATOR PADA PHP DAN DATABASE**

#### **2.1. Operator pada PHP**

PHP mendukung berbagai macam operator untuk melakukan operasi pada nilai-nilai atau variabel.

Operator merupakan symbol yang biasa di libatkan dalam program untuk melakukan suatu operasi atau manipulasi.

Misalnya untuk:

1. Menjumlahkan dua buah nilai
2. Memberikan nilai ke suatu variable
3. Membandingkan kesamaan dua buah nilai

Berikut adalah beberapa jenis operator yang umum digunakan dalam PHP:

##### **2.1.1. Operator Aritmatika**

Adalah operator yang digunakan dalam melakukan suatu proses operasi matematika, seperti penjumlahan, perkalian dan sebagainya.

Contoh:

$\$a + \$b$

symbol  $+$  merupakan operator untuk melakukan operasi penjumlahan dari kedua operand-nya (yaitu  $\$a$  dan  $\$b$ ). Karena operator penjumlahan melibatkan dua buah operand, maka operator ini tergolong sebagai operator binary.

**Tabel 2.1. Operator Aritmatika**

Operasi	nama	Hasil
$\$a + \$b$	Penjumlahan	Jumlah dari $\$a$ dan $\$b$
$\$a - \$b$	Pengurangan	Sisa dari pengurangan $\$b$ dari $\$a$ .
$\$a * \$b$	Perkalian	Hasil kali dari $\$a$ dan $\$b$ .
$\$a / \$b$	Pembagian	Hasil bagi dari $\$a$ dan $\$b$ .
$\$a \% \$b$	Modulus	sisa bagi dari $\$a$ terhadap $\$b$ .

Operator pembagian (" $/$ ") mengembalikan suatu nilai bulat (hasil dari suatu pembagian bulat) Jika kedua operan adalah integer (atau string yang dikonversi ke integer). Jika salah satu operan adalah nilai floating-point value, pembagian floating-point dilakukan. Operator seperti penjumlahan, pengurangan, perkalian dan pembagian merupakan hal yang umum. Adapun operator sisa pembagian (modulus) berupa  $\%$  adalah sesuatu yang berbeda dengan operator sebelumnya.

Contoh :

$10 \% 3 \Rightarrow 1$	10 dibagi 3 sisanya adalah 1
$8 \% 2 \Rightarrow 0$	8 dibagi 2 sisanya 0
$8 \% 3 \Rightarrow 2$	8 dibagi 3 sisanya 2

### 2.1.2. Operator Perbandingan

Operator perbandingan, sesuai dengan namanya, memperbolehkan anda untuk membandingkan dua nilai.

**Table 2.2. Operator Perbandingan**

Contoh	Nama	Hasil
$\$a == \$b$	Sama dengan	Benar jika $\$a$ sama dengan $\$b$
$\$a != \$b$	Tidak sama	Benar jika $\$a$ tidak sama dengan $\$b$
$\$a < \$b$	Lebih kecil	Benar jika $\$a$ lebih kecil dari $\$b$
$\$a > \$b$	Lebih besar	Benar jika $\$a$ lebih besar dari $\$b$
$\$a <= \$b$	Lebih kecil atau sama	Benar jika $\$a$ lebih kecil atau sama dengan $\$b$
$\$a >= \$b$	Lebih besar atau sama	Benar jika $\$a$ lebih besar atau sama dengan $\$b$

```

$a = 5;
$b = 10;

$equalTo = ($a == $b); // Sama dengan
$notEqualTo = ($a != $b); // Tidak sama dengan
$greaterThan = ($a > $b); // Lebih besar dari
$lessThan = ($a < $b); // Lebih kecil dari
    
```

```
$greaterOrEqual = ($a >= $b); // Lebih besar atau sama dengan
$lessOrEqual = ($a <= $b); // Lebih kecil atau sama dengan
```

## 2.1.2. Operator Logika

Operator logika biasa digunakan untuk menghubungkan dua buah ungkapan kondisi menjadi sebuah ungkapan kondisi

**Table 2.2. Operator Logical**

Contoh	Nama	Hasil
<code>\$a and \$b</code>	And	True kalau kedua \$a dan \$b adalah true.
<code>\$a or \$b</code>	Or	True jika salah satu \$a atau \$b true.
<code>\$a xor \$b</code>	Or	True jika salah satu \$a atau \$b true, tetapi tidak untuk keduanya.
<code>!\$a</code>	Not	True jika \$a tidak true.
<code>\$a &amp;&amp; \$b</code>	And	True kalau kedua \$a dan \$b adalah true.
<code>\$a    \$b</code>	Or	True jika salah satu \$a atau \$b true.

```
$condition1 = true;
$condition2 = false;

$andResult = ($condition1 && $condition2); // Operator logika AND
$orResult = ($condition1 || $condition2); // Operator logika OR
$notResult = !$condition1; // Operator logika NOT - tidak sama dengan
```

## 2.1.4. Operator pemberi nilai atau penugasan

Operator pemberian nilai (penugasan) yang disimbolkan dengan tanda (=) sama dengan, berguna untuk memberikan suatu nilai kesuatu variable. Operator ini dapat digunakan sebagai ungkapan ataupun berdiri sendiri sebagai pernyataan.

```
$a = 2; // memberikan nilai 2 ke $a
$a = ($b = 4) + 5; // $a is equal to 9 now, and $b has been set to 4.
$a = 2 + 2;

$x += $y; // Penugasan Penjumlahan (sama dengan $x = $x + $y)
$x -= $y; // Penugasan Pengurangan
$x *= $y; // Penugasan Perkalian
$x /= $y; // Penugasan Pembagian
$x %= $y; // Penugasan Modulus
```

### 2.1.5. Operator Bitwise

Operator Bitwise memungkinkan anda untuk memanipulasi data dalam bentuk bit tertentu dalam suatu integer on atau off.

**Table 2.4. Operator Bitwise**

Contoh	Nama	Hasil
<code>\$a &amp; \$b</code>	And	Bit yang bernilai 1 pada \$a dan \$b akan di-set 1
<code>\$a   \$b</code>	Or	Bit yang bernilai 1 pada \$a atau \$b akan di di-set 1.
<code>\$a ^ \$b</code>	Xor	Bit yang bernilai 1 pada \$a atau \$b tetapi tidak pada keduanya akan di set 1.
<code>~ \$a</code>	Not	Bit yang bernilai 1 pada \$a akan di set 0 dan sebaliknya
<code>\$a &lt;&lt; \$b</code>	Shift left	Geser kekiri Bit-bit di \$a sebanyak bit \$b (setiap langkah berarti kalikan dua)
<code>\$a &gt;&gt; \$b</code>	Shift right	Geser kekanan Bit-bit di \$a sebanyak bit \$b (setiap langkah berarti bagikan dengan dua)

### 2.1.6. Operator Eksekusi

PHP mendukung sebuah operator eksekusi yang disimpan dalam tanda backticks (`). Dalam tanda backticks tersebut dapat dimasukan perintah yang dilakukan dalam shell, kemudian outputnya akan dikeluarkan di jendela browser.

Contoh:

```
$output = `ls -al`;
echo "<pre>$output</pre>";
```

### 2.1.7. Operator Penaikan dan Penurunan

PHP mendukung perintah C-style yang berhubungan dengan penaikan (increment) dan penurunan (decrement). Kedua operator ini digunakan pada operand bertipe bilangan bulat (integer).

**Tabel 2.5. Increment/decrement Operators**

Contoh	Nama	Hasil
<code>++\$a</code>	Pre-increment	Naikan nilai \$a sebanyak 1, kembalikan di \$a.
<code>\$a++</code>	Post-increment	Kembalikan \$a, kemudian naikan \$a sebanyak 1.
<code>--\$a</code>	Pre-decrement	Turunkan \$a sebanyak 1, kembalikan di \$a.
<code>\$a--</code>	Post-decrement	Kembalikan \$a, kemudian turunkan \$a sebanyak 1.

#### Contoh 1:

```
$a = 10;  
$b = 10 + ++$a; // $a akan bernilai 11 dan $b akan bernilai 21  
//Perintah diatas akan sama dengan perintah berikut:  
$a = 10 + 1;  
$b = 10 + 11
```

#### Contoh 2:

```
$a = 10;  
$b = 10 + $a++; // $a akan bernilai 11 dan $b akan bernilai 20  
//Perintah diatas akan sama dengan perintah berikut:  
$b = 10 + $a  
$a = 10 + 1;
```

### 2.1.8. Operator Prioritas

Operator prioritas menentukan bagaimana suatu ekspresi dikerjakan oleh PHP. Operator yang mempunyai prioritas tinggi akan didahulukan dalam hal pengerjaannya dibandingkan dengan operator yang memiliki prioritas lebih rendah.

Contoh:

$1 + 5 * 2$ , jawabannya adalah 16 dan bukan 18, sebab tanda operator kali (\*) lebih tinggi derajatnya dibandingkan dengan operator penjumlahan (+).

Dibawah ini adalah tabel operator prioritas dimana operator yang mempunyai derajat lebih rendah akan disimpan diawal.

**Table 2.6. Operator Prioritas**

Associativity	Operators
left	,
left	Or
left	Xor
left	And
right	print
left	= += -= *= /= .= %= &=  = ^= .= <<= >>=
left	? :
left	
left	&&
left	
left	^
left	&
non-associative	== != ===
non-associative	< <= > >=
left	<< >>
left	+ - .
left	* / %
right	! ~ ++ -- (int) (double) (string) (array) (object) @
right	[
non-associative	new

### 2.1.9. Operator String

Operator string adalah operator yang digunakan untuk menggabungkan dua atau beberapa string. Hanya ada satu operator string dalam PHP yaitu operator penggabungan string (".") titik.

Contoh:

```
$a = "Hello ";
$b = $a . "World!"; // now $b = "Hello World!"
```

### 2.1.10. Operator Ternary

Operator ternary adalah cara singkat untuk menulis suatu ekspresi kondisional dalam satu baris. Bentuk umumnya adalah:

```
$variabel = (kondisi) ? nilai_jika_true : nilai_jika_false;
```

Artinya, jika kondisi yang diberikan bernilai true, maka nilai yang terletak sebelum tanda ":" akan digunakan untuk variabel; jika kondisinya bernilai false, maka nilai yang terletak setelah tanda ":" akan digunakan.

Contoh penggunaan operator ternary:

```
$umur = 20;  
// Menggunakan operator ternary untuk menetapkan kategori berdasarkan umur  
$kategorinya = ($umur < 18) ? "Anak-anak" : "Dewasa";  
echo $kategorinya; // Output: "Dewasa"
```

Dalam contoh ini, `$kategorinya` akan diisi dengan nilai "Anak-anak" jika `$umur` kurang dari 18, dan "Dewasa" jika tidak. Ini adalah cara singkat dan bersih untuk menulis pernyataan kondisional yang sederhana. Namun, pada kondisi yang kompleks atau saat menjaga kejelasan, pernyataan kondisional yang lebih panjang atau `if-else` standar mungkin lebih disarankan

### 2.1.11. Operator Array

PHP memiliki beberapa operator yang dapat digunakan untuk melakukan operasi pada array. Berikut adalah beberapa operator array yang umum digunakan:

#### 1. Penggabungan Array (Concatenation)

Operator `+` digunakan untuk menggabungkan dua array.

```
$array1 = [1, 2, 2];  
$array2 = [4, 5, 6];  
$combinedArray = $array1 + $array2; // Penggabungan array
```

#### 2. Penggabungan Array (Union)

Operator `+` juga dapat digunakan untuk menggabungkan array, tetapi duplikasi elemen dari array pertama akan diabaikan.

```
$array1 = [1, 2, 2];  
$array2 = [2, 4, 5];  
$unionArray = $array1 + $array2;  
// Hasil: [1, 2, 2, 4, 5]
```

#### 3. Penggabungan Array (`array_merge()`):

Fungsi `array_merge()` dapat digunakan untuk menggabungkan satu atau lebih array.

```
$array1 = [1, 2, 2];  
$array2 = [4, 5, 6];
```

```
$mergedArray = array_merge($array1, $array2);  
// Hasil: [1, 2, 2, 4, 5, 6]
```

#### 4. Penggabungan Array (array\_merge\_recursive):

Fungsi `array_merge_recursive()` dapat digunakan untuk menggabungkan array secara rekursif.

```
$array1 = ["a" => ["red"], "b" => "green"];  
$array2 = ["a" => "blue", "b" => ["yellow"]];  
$mergedArray = array_merge_recursive($array1, $array2);  
// Hasil: ["a" => ["red", "blue"], "b" => ["green", "yellow"]]
```



## 2.2. Operator Pada Database *(Diajarkan pada materi database)*

Dalam Database anda dapat menggunakan operator sebagaimana pada pemrograman php.

Dalam MySQL, operator digunakan untuk melakukan operasi pada nilai-nilai dalam pernyataan SQL. Berikut adalah beberapa operator yang umum digunakan dalam MySQL:

### 2.2.1. Operator Aritmatika:

Digunakan untuk melakukan operasi matematika.

`SELECT nama, harga, stok, harga * stok as total FROM barang;`

nama	harga	stok	total
PRINTER	1250000	5	6250000
HANDPHONE	2000000	10	20000000
PRINTER COLOR	2500000	3	7500000
KOMPUTER	2500000	5	12500000
SSD	250000	2	500000
MONITOR	1500000	5	7500000
HARDDISK	450000	3	1350000
MEMORY 16 GB	250000	5	1250000
LAPTOP ACER	6000000	10	60000000
USB FLASHDISK	150000	4	600000

### 2.2.2. Operator Perbandingan:

Digunakan untuk membandingkan nilai.

`SELECT * FROM barang WHERE stok > 2;`

KODE	NAMA	HARGA	STOK	SUPPLIER
B001	PRINTER	1250000	5	SP001
B002	HANDPHONE	2000000	10	SP002
B004	KOMPUTER	2500000	5	SP002
B006	MONITOR	1500000	5	SP004
B008	MEMORY 16 GB	250000	5	SP003
B009	LAPTOP ACER	6000000	10	SP003
B010	USB FLASHDISK	150000	4	SP003

### 2.2.2. Operator Logika:

Digunakan untuk membuat kondisi logika.

```
SELECT * FROM barang WHERE harga > 1000000 and harga <5000000;
```

KODE	NAMA	HARGA	STOK	SUPPLIER
B001	PRINTER	1250000	5	SP001
B002	HANDPHONE	2000000	10	SP002
B003	PRINTER COLOR	2500000	3	SP003
B004	KOMPUTER	2500000	5	SP002
B006	MONITOR	1500000	5	SP004

### 2.2.4. Operator Gabungan String (Concatenation):

Menggabungkan nilai-nilai string.

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM employees;
```

### 2.2.5. Operator Penugasan:

Digunakan untuk menetapkan nilai ke variabel.

```
SET @my_variable = 5;
```

### 2.2.6. Operator IN:

Digunakan untuk memeriksa apakah nilai berada dalam satu set nilai.

```
SELECT * FROM barang WHERE harga IN (1000000, 1500000, 2000000);
```

KODE	NAMA	HARGA	STOK	SUPPLIER
B002	HANDPHONE	2000000	10	SP002
B006	MONITOR	1500000	5	SP004

### 2.2.7. Operator BETWEEN:

Digunakan untuk memeriksa apakah nilai berada dalam rentang tertentu.

`SELECT * FROM barang WHERE harga BETWEEN 1000000 AND 2000000;`

KODE	NAMA	HARGA	STOK	SUPPLIER
B001	PRINTER	1250000	5	SP001
B002	HANDPHONE	2000000	10	SP002
B006	MONITOR	1500000	5	SP004

### 2.2.8. Operator LIKE:

Digunakan untuk mencocokkan pola string.

Misal:tampilkan seluruh isi tabel barang yang nama depannya diawali dengan huruf M

`SELECT * FROM barang WHERE nama LIKE 'M%';`

KODE	NAMA	HARGA	STOK	SUPPLIER
B006	MONITOR	1500000	5	SP004
B008	MEMORY 16 GB	250000	5	SP003

### 2.2.9. Operator IS NULL / IS NOT NULL:

Digunakan untuk memeriksa apakah nilai kolom NULL atau tidak.

`SELECT * FROM my_table WHERE column1 IS NULL;`

### 2.2.10. Operator UNION:

Menggabungkan hasil dua atau lebih pernyataan SELECT.

```
SELECT nama FROM barang
UNION
SELECT nama FROM supplier;
```

nama

PRINTER

HANDPHONE

PRINTER COLOR

KOMPUTER

SSD

MONITOR

HARDDISK

MEMORY 16 GB

LAPTOP ACER

USB FLASHDISK

PT.ABADI

CV.UNIKOM

CV.GARUDA

PT.NASIONAL

CV.ALLISA PRATAMA

PT.CIWARUGA DOT COM

### 2.2.11. Operator JOIN:

Menggabungkan baris dari dua atau lebih tabel berdasarkan kolom yang sesuai.

```
SELECT * FROM table1
INNER JOIN table2 ON table1.col1 = table2.col1;
```

```
SELECT * FROM barang
INNER JOIN supplier ON barang.supplier = supplier.kode;
```



KODE	NAMA	HARGA	STOK	SUPPLIER	KODE	NAMA	ALAMAT	TELP
B001	PRINTER	1250000	5	SP001	SP001	PT.ABADI	JL.MERDEKA 25 BANDUNG	022-2507689
B002	HANDPHONE	2000000	10	SP002	SP002	CV.UNIKOM	JL.DIPATIUKUR 112 BANDUNG	0816789666
B003	PRINTER COLOR	2500000	3	SP003	SP003	CV.GARUDA	JL.RIAU 67	0878232222
B004	KOMPUTER	2500000	5	SP002	SP002	CV.UNIKOM	JL.DIPATIUKUR 112 BANDUNG	0816789666
B005	SSD	250000	2	SP003	SP003	CV.GARUDA	JL.RIAU 67	0878232222
B006	MONITOR	1500000	5	SP004	SP004	PT.NASIONAL	JL.PAJAJARAN 88 CIMAH	0866622222
B007	HARDDISK	450000	3	SP002	SP002	CV.UNIKOM	JL.DIPATIUKUR 112 BANDUNG	0816789666
B008	MEMORY 16 GB	250000	5	SP003	SP003	CV.GARUDA	JL.RIAU 67	0878232222
B009	LAPTOP ACER	6000000	10	SP003	SP003	CV.GARUDA	JL.RIAU 67	0878232222
B010	USB FLASHDISK	150000	4	SP003	SP003	CV.GARUDA	JL.RIAU 67	0878232222

### 2.2.12. Operator ALIAS:

Digunakan untuk memberikan nama alias pada kolom atau tabel.

SELECT nama AS "NAMA BARANG" FROM barang;

**NAMA BARANG**

PRINTER

HANDPHONE

PRINTER COLOR

KOMPUTER

SSD

MONITOR

HARDDISK

MEMORY 16 GB

LAPTOP ACER

USB FLASHDISK

### 2.2.13. Operator AGGREGATE:

Operator seperti COUNT(), SUM(), AVG(), MIN(), MAX() digunakan untuk melakukan operasi agregasi pada data.

Contoh: ingin menampilkan berapa jumlah baris yang ada dalam tabel barang;

SELECT COUNT(\*) AS total\_rows FROM my\_table;

**total\_rows**

10

Menghitung berapa Total harga yang ada dalam tabel Barang.

```
SELECT SUM(harga) AS total_rows FROM barang;
```

```
total_rows  
16850000
```

## 2.2.14. Operator ORDER BY:

Mengurutkan hasil kueri berdasarkan kolom tertentu.

```
SELECT * FROM my_table ORDER BY column1 DESC;
```

Misal:Tampilkan seluruh isi tabel supplier dengan urutan nama dari Besar ke Kecil atau dari Abjad huruf paling akhir urut ke abjad huruf awal

```
SELECT * FROM supplier ORDER BY nama DESC;
```

KODE	NAMA	ALAMAT	TELP
SP004	PT.NASIONAL	JL.PAJAJARAN 88 CIMAHI	0866622222
SP006	PT.CIWARUGA DOT COM	JLN.CIWARUGA 62 BANDUNG	022-25073232
SP001	PT.ABADI	JL.MERDEKA 25 BANDUNG	022-2507689
SP002	CV.UNIKOM	JL.DIPATIUUR 112 BANDUNG	0816789666
SP003	CV.GARUDA	JL.RIAU 67	0878232222
SP005	CV.ALLISA PRATAMA	JL.SETIABUDI 102 BANDUNG	08232223322

### 3.2.15. Operator LIMIT dan OFFSET (atau FETCH dan OFFSET):

Digunakan untuk membatasi jumlah baris yang dikembalikan oleh kueri.

`SELECT * FROM my_table LIMIT 10 OFFSET 20;`

Misalnya isi tabel barang seperti berikut:

KODE	NAMA	HARGA	STOK	SUPPLIER
B001	PRINTER	1250000	5	SP001
B002	HANDPHONE	2000000	10	SP002
B003	PRINTER COLOR	2500000	3	SP003
B004	KOMPUTER	2500000	5	SP002
B005	SSD	250000	2	SP003
B006	MONITOR	1500000	5	SP004
B007	HARDDISK	450000	3	SP002
B008	MEMORY 16 GB	250000	5	SP003
B009	LAPTOP ACER	6000000	10	SP003
B010	USB FLASHDISK	150000	4	SP003

Ingin ditampilkan data dari mulai baris ke 4 sebanyak 5 baris. Maka Perintah SQL nya adalah seperti berikut:

`SELECT * FROM barang LIMIT 5 OFFSET 2;`

KODE	NAMA	HARGA	STOK	SUPPLIER
B004	KOMPUTER	2500000	5	SP002
B005	SSD	250000	2	SP003
B006	MONITOR	1500000	5	SP004
B007	HARDDISK	450000	3	SP002
B008	MEMORY 16 GB	250000	5	SP003

Itu adalah beberapa operator umum yang digunakan dalam SQL untuk melakukan berbagai operasi pada database dan mengambil data sesuai dengan kebutuhan kueri Anda.

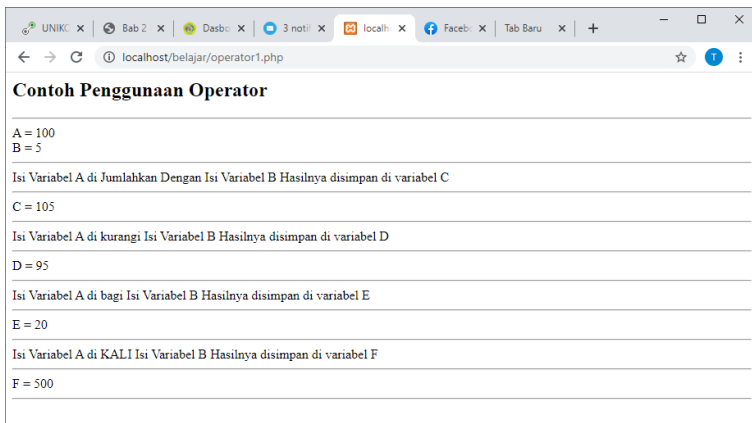
## Latihan 2

### 1. Berikut adalah contoh penggunaan Operator dalam Program PHP Nama File:operator1.php

```
<?php
$A=100;
$B=5;
$C=$A+$B; //Operator + untuk penjumlahan
$D=$A-$B; //Operator - untuk pengurangan
$E=$A/$B; //Operator / untuk pembagian
$F=$A*$B; //Operator * untuk perkalian

echo "<h2>Contoh Penggunaan Operator</h2>";
echo "<hr>";
echo "A = $A<br>";
echo "B = $B<hr>";

echo "Variabel A di Jumlahkan Variabel B Hasilnya disimpan di variabel C<hr>";
echo "C = $C<hr>";
echo "Variabel A di Kurangi Variabel B Hasilnya disimpan di variabel D<hr>";
echo "D = $D<hr>";
echo "Variabel A di Bagi Variabel B Hasilnya disimpan di variabel E<hr>";
echo "E = $E<hr>";
echo "Variabel A di KALI Variabel B Hasilnya disimpan di variabel F<hr>";
echo "F = $F<hr>";
?>
```





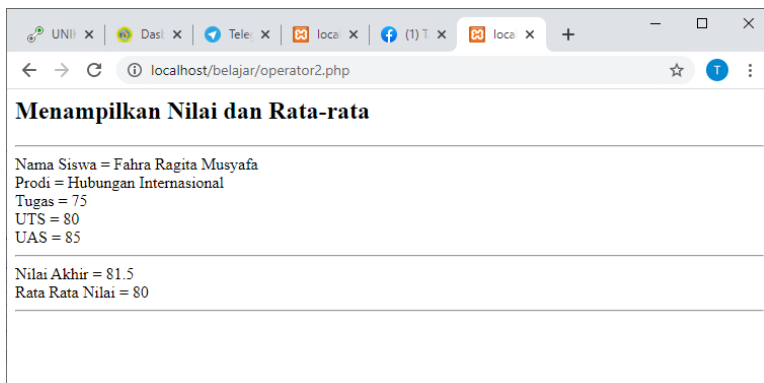


Gambar 2.2. Hasil Program operator1.php

2. Membuat Program Untuk mencari Nilai Akhir dan nilai Rata-rata  
Diketahui nilai Tugas, UTS dan UAS,  
Nilai Akhir = 20% tugas + 20% uts + 50% uas  
Rata = (tugas+uts+uas)/2;

NamaFile:operator2.php

```
<?php
$namasiswa="Fahra Ragita Musyafa";
$prodi="Hubungan Internasional";
$tugas=75;
$uts=80;
$uas=85;
$nilaiakhir=(0.2*$tugas) + (0.2*$uts) + (0.5*$uas);
$rata=($tugas+$uts+$uas)/2;
echo "<h2>Menampilkan Nilai dan Rata-rata</h2>";
echo "<hr>";
echo "Nama Siswa = $namasiswa<br>";
echo "Prodi = $prodi<br>";
echo "Tugas = $tugas<br>";
echo "UTS = $uts<br>";
echo "UAS = $uas<br>";
echo "Nilai Akhir = $nilaiakhir<br>";
echo "Rata Rata Nilai = $rata<br>";
?>
```



Gambar 2.2. Hasil Program Menampilkan Nilai Akhir dan Rata-rata

## DAFTAR PUSTAKA

1. Suryana, Taryana (2024), [Pengantar pemrograman web HTML, CSS, PHP dan MYSQL](#), Deepublish, Yogyakarta
2. Suryana, Taryana; Koesheryatin (2014), (2023), [Aplikasi Internet Menggunakan HTML, CSS & Java Script](#), Elexmedia Komputindo, Jakarta
3. Suryana, Taryana (2010) [Membuat Web Pribadi dan Bisnis dengan HTML](#), Gava Media, Yogyakarta
4. Suryana, Taryana; Sarwono, Jonathan (2007) [E-COMMERCE MENUNGGUNAKAN PHP DAN MYSQL](#), Graha Ilmu, Yogyakarta
5. Wong Jony. 2010. *Internet Marketing for Beginners*. Jakarta: PT Elex Media Komputindo.
6. Kumar, Anil and Manoj Kumar Dash. 2013. *Constructing a Measurement in Service Quality for Indian Banks: Structural Equation Modeling Approach*. *Journal of Internet Banking and Commerce*, April 2013, vol. 18, no.1
7. Siregar, Riki R. 2010. *Strategi Meningkatkan Persaingan Bisnis Perusahaan dengan Penerapan E-commerce*
8. <https://www.php.net/manual/en/book.mysql.php>
9. <https://www.w3schools.com>
10. <https://tekno.kompas.com/read/2022/09/26/10150007/pengertian-e-commerce-beserta-jenis-contoh-dan-manfaatnya>
11. <https://meenta.net/pengertian-e-commerce/>
12. <https://www.rumahweb.com/journal/session-adalah/>
13. <https://rizkypratama.id/blog/2019/11/01/membuat-session-login-pada-php/>
14. <https://www.duniailkom.com/>
15. <https://polinema-programming.github.io/>
16. <https://jagongoding.com/python/dasar/perulangan-besarang/>
17. <https://runestone.academy/>
18. <https://lmsspada.kemdikbud.go.id/>
19. <https://jagongoding.com/>

20. [https://repository.dinus.ac.id/docs/ajar/Pert\\_5\\_Percabangan\\_Bersarang.pptx](https://repository.dinus.ac.id/docs/ajar/Pert_5_Percabangan_Bersarang.pptx)

