

ASSIGNMENT DAN OPERATOR ARITMATIKA

Mata Kuliah: Pemograman Web

Materi Praktikum ke: 7



NIM: 2411102441046

Nama: Fahriandy Adithia

**FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI S1 TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH KALIMANTAN
TIMUR**

BAB I PENDAHULUAN

A. Latar Belakang

Dalam konteks pengembangan perangkat lunak, khususnya pada pemrograman sisi server menggunakan PHP, konsep dasar dalam manipulasi data dan perhitungan matematis sangatlah vital. Dua pilar utama yang mendasari fungsionalitas ini adalah operator assignment dan operator aritmatika. Operator aritmatika ($+$, $-$, $*$, $/$, $\%$) memungkinkan programmer untuk melaksanakan berbagai perhitungan numerik standar, yang merupakan inti dari banyak logika bisnis dan pengolahan data. Sementara itu, operator assignment ($=$, $+=$, $-=$, dll.) berperan penting dalam menginisialisasi dan memperbarui nilai variabel secara efisien. Penguasaan yang solid terhadap mekanisme kerja kedua jenis operator ini menjadi prasyarat dalam menyusun kode PHP yang efektif, dinamis, dan mudah dikelola (termasuk aspek pembacaan dan pemeliharaan). Praktikum ini bertujuan untuk memperdalam pemahaman dan keterampilan implementasi kedua jenis operator tersebut dalam skenario pemrograman dasar.

B. Tujuan

- Mendefinisikan dan **memahami fungsi** dasar dari **operator assignment** dan **operator aritmatika** dalam bahasa pemrograman PHP.
- Mengidentifikasi dan **mengklasifikasikan** berbagai operator aritmatika ($+$, $-$, $*$, $/$, $\%$, $\text{dan } **$) beserta **prioritas eksekusinya**.
- Menganalisis dan **menerapkan** penggunaan **operator assignment** dalam proses inisialisasi dan pembaruan nilai variabel di kode PHP.
- **Mengimplementasikan** operator-operator tersebut untuk **menyelesaikan kasus-kasus** perhitungan matematis dasar hingga menengah dalam skrip PHP.
- Merancang dan **mengembangkan struktur logika** program sederhana yang melibatkan perhitungan berbasis operator *assignment* dan aritmatika menggunakan PHP.

C. Tinjauan Pustaka

2.1 Variabel PHP

Variabel adalah elemen fundamental dalam pemrograman yang berfungsi sebagai wadah untuk menyimpan nilai data. Dalam PHP, variabel harus diawali dengan tanda dolar. Variabel adalah objek utama yang dimanipulasi oleh operator *assignment* dan aritmatika.

2.2 Operator *Assignment* (Penugasan)

Operator *assignment* bertugas menetapkan atau memberikan nilai kepada sebuah variabel. Operator dasarnya adalah tanda sama dengan

Selain itu, PHP menyediakan operator *assignment* gabungan (Operator gabungan ini menyederhanakan proses di mana sebuah variabel diperbarui nilainya berdasarkan operasi aritmatika yang melibatkan nilai lamanya.

2.3 Operator Aritmatika

Operator Aritmatika digunakan untuk melakukan operasi matematika dasar pada nilai-nilai numerik.

Jenis operator ini meliputi:

- **Penjumlahan**
- **Pengurangan**
- **Perkalian**
- **Pembagian**
- **Modulus** Menghasilkan sisa hasil bagi.
- **Eksponensial** Untuk perpangkatan.

Dalam mengeksekusi ekspresi, PHP mematuhi aturan prioritas operator (seperti PEMDAS), di mana perkalian, pembagian, dan modulus akan dievaluasi lebih dulu daripada penjumlahan dan pengurangan. Prioritas dapat diubah menggunakan tanda kurung

BAB II

ALAT DAN BAHAN

A. Alat

1. Komputer atau Laptop
2. Visual Studio Code
3. Browser

B. Bahan

1. Dokumentasi

BAB III PROSEDUR KERJA

A. ByValue

1.

```
1-byvalue.php
1  <?php
2  $a = 20;
3  $b = 15;
4  $c = 5;
5
6  echo "\$a = $a, \$b = $b, \$c = $c";
7  echo "<br />";
8  // hasil proses: $a = 20, $b = 15, $c = 5
9
10 $a = $b = $c + 5;
11 echo "\$a = $a, \$b = $b, \$c = $c";
12 // hasil proses: $a = 10, $b = 10, $c = 5
13 ?>
```

Output :

\$a = 20, \$b = 15, \$c = 5

\$a = 10, \$b = 10, \$c = 5

B. By Array

1.

```
2-byarray1.php
1  <?php
2  //pembuatan array
3  $nama = array(
4      1=>"Andri",
5      2=>"Joko",
6      3=>"Sukma",
7      4=>"Rina",
8      5=>"Sari");
9
10 //cara akses array
11 echo $nama[1]; //Andri
12 echo "<br />";
13 echo $nama[2]; //Joko
14 echo "<br />";
15 echo $nama[3]; //Sukma
16 ?>
```

2.

```
2-byarray2.php
1  <?php
2  // pembuatan array
3  $nama = ["Andri", "Joko", "Sukma", "Rina", "Sari"];
4
5  // pengaksesan array
6  echo $nama[1]; //Joko
7  echo "<br />";
8  echo $nama[2]; //Sukma
9  echo "<br />";
10 echo $nama[3]; //Rina
11 ?>
```

Output :

Andri
Joko
Sukma

Joko
Sukma
Rina

C. By Reference

1.

```

3-byreference1.php
1  <?php
2  $a = 20;
3  $b = $a;
4
5  echo "\$a = $a, \$b = $b";
6  echo "<br />";
7  // hasil proses: $a = 20, $b = 20
8
9  $a = $a + 5;
10 echo "\$a = $a, \$b = $b";
11 echo "<br />";
12 // hasil proses: $a = 25, $b = 20
13
14 $b = $b + 10;
15 echo "\$a = $a, \$b = $b";
16 // hasil proses: $a = 25, $b = 30
17 ?>

```

2.

```

3-byreference2.php
1  <?php
2  $a = 20;
3  $b = 15;
4  $c = 5;
5
6  echo "\$a = $a, \$b = $b, \$c = $c";
7  echo "<br />";
8  // hasil proses: $a = 20, $b = 15, $c = 5
9
10 $a = $b = $c + 5;
11 echo "\$a = $a, \$b = $b, \$c = $c";
12 // hasil proses: $a = 10, $b = 10, $c = 5
13 ?>

```


Output :

$Sa = 20, Sb = 20$

$Sa = 25, Sb = 20$

$Sa = 25, Sb = 30$

$Sa = 20, Sb = 15, Sc = 5$

$Sa = 10, Sb = 10, Sc = 5$

D. Aritmatik

1.

```
4-aritmatik.php
1  <?php
2  $penjumlahan = 2 + 4;
3  $pengurangan = 6 - 2;
4  $perkalian = 5 * 3;
5  $pembagian = 15 / 3;
6  $modulus = 5 % 2;
7
8  echo "Hasil: 2 + 4 = " . $penjumlahan . "<br>";
9  // Hasil: 2 + 4 = 6
10 echo "Hasil: 6 - 2 = " . $pengurangan . "<br>";
11 echo "Hasil: 5 * 3 = " . $perkalian . "<br>";
12 // Hasil: 5 * 3 = 15
13 echo "Hasil: 15 / 3 = " . $pembagian . "<br>";
14 echo "Hasil: 5 % 2 = " . $modulus;
15 // Hasil: 5 % 2 = 1
16 ?>
```

Output :

Hasil: 2 + 4 = 6

Hasil: 6 - 2 = 4

Hasil: 5 * 3 = 15

Hasil: 15 / 3 = 5

Hasil: 5 % 2 = 1

E. Presedensi

1.

```
🐘 5-presedensi.php
1  <?php
2  $a = 3 + 4 * 5 - 6;
3  echo $a;
4  // hasil $a = 17
5  echo "<br />";
6  $a = (3 + 4) * 5 - 6;
7  echo $a;
8  // hasil $a
```

Output :

17
29

F. Increment

1.

```
6-increment.php
1  <?php
2  $x = 4;
3  $x++;
4  echo "Nilai x yang baru : " . $x;
5  echo "<br />";
6  // hasil $x = 5
7  $x = 4;
8  $x--;
9  echo "Nilai x yang baru : " . $x;
10 // hasil $x = 3
11 ?>
```

Output :

Nilai x yang baru : 5

Nilai x yang baru : 3

G. Script

1.

```
script5-1.php
1 <html>
2 <head>
3 <title>Menghitung Komisi Salesman</title>
4 </head>
5 <body>
6 <h1>Menghitung Komisi Salesman</h1>
7 <?php
8 /*
9 Script ini akan menghitung komisi salesman berdasarkan nilai penjualan
10 yang dicapainya yaitu sebesar Rp. 1.500.000,-
11 ketentuan komisinya adalah 5% dari nilai penjualan yang dicapai.
12 */
13 $nilaiJual = 1500000; // nilai penjualan yang didapat salesman
14 $komisi = 0.05 * $nilaiJual; // menghitung komisi yaitu 5% dari nilai penjualan
15
16 echo "<p>Nilai penjualan salesman : Rp. " . $nilaiJual . "</p>"; // menampilkan nilai penjualan salesman
17 echo "<p>Komisi yang didapat salesman adalah Rp. " . $komisi . "</p>";
18 // menampilkan hasil perhitungan komisi
19 ?>
20 </body>
21 </html>
```

2.

```
script5-2.php
1 <html>
2 <head>
3 <title>Menghitung Gaji Bersih Karyawan</title>
4 </head>
5 <body>
6 <h1>Menghitung Gaji Bersih Karyawan</h1>
7 <?php
8 /*
9 Script ini akan menghitung gaji karyawan yang dirumuskan dengan
10 Gaji Bersih = Gaji Pokok + tunjangan - pajak;
11 Misalkan gaji pokoknya Rp. 1.000.000, tunjangan Rp. 500.000 dan
12 pajaknya 15% dari (gaji kotor = gaji pokok + tunjangan)
13 Berikut ini adalah beberapa cara pembuatan script yang akan menghasilkan
14 output yang sama
15 */
16
17 // CARA KE - 1
18 $gajiPokok = 1000000; // gaji pokok
19 $tunjangan = 500000; // tunjangan
20 $gajiKotor = $gajiPokok + $tunjangan; // hitung gaji kotor
21 $pajak = 0.15 * $gajiKotor; // hitung pajak
22 $gajiBersih = $gajiPokok + $tunjangan - $pajak; // hitung gaji bersih
23 echo "<p>Gaji bersih karyawan adalah Rp. " . $gajiBersih . "</p>"; // menampilkan gaji bersih
24
25 // CARA KE - 2
26 $gajiPokok = 1000000; // gaji pokok
27 $tunjangan = 500000; // tunjangan
28 $gajiKotor = $gajiPokok + $tunjangan; // hitung gaji kotor
29 $gajiBersih = $gajiKotor - (0.15 * $gajiKotor); // hitung gaji bersih
30 echo "<p>Gaji bersih karyawan adalah Rp. " . $gajiBersih . "</p>"; // menampilkan gaji bersih
31
32 // CARA KE - 3
33 $gajiPokok = 1000000; // gaji pokok
34 $tunjangan = 500000; // tunjangan
35 $gajiBersih = $gajiPokok + $tunjangan - 0.15 * ($gajiPokok + $tunjangan); // hitung gaji bersih
36 echo "<p>Gaji bersih karyawan adalah Rp. " . $gajiBersih . "</p>"; // menampilkan gaji bersih
37 ?>
38 </body>
39 </html>
```

3.

```

script5-3.php
1 <html>
2 <head>
3   <title>Konversi Waktu Tempuh Ke Detik</title>
4 </head>
5 <body>
6   <h1>Konversi Waktu Tempuh Ke Detik</h1>
7   <?php
8   /*
9    Script ini akan mengkonversi waktu yang dinyatakan dalam 10:16:42 (10
10   jam, 16 menit dan 42 detik) ke dalam satuan detik.
11   */
12   $jam = 10;
13   $menit = 16;
14   $detik = 42;
15
16   $jamKeDetik = $jam * 3600; // konversi jam ke detik
17   $menitKeDetik = $menit * 60; // konversi menit ke detik
18   $detikKeDetik = $detik; // konversi ke detik
19
20   $totalDetik = $jamKeDetik + $menitKeDetik + $detikKeDetik; // hitung total waktu dalam detik
21
22   echo "<p>Jika waktu " . $jam . ":" . $menit . ":" . $detik . " dinyatakan dalam
23   satuan detik adalah : " . $totalDetik . "</p>";
24   ?>
25 </body>
26 </html>

```

4.

```

script5-4.php
1 <html>
2 <head>
3   <title>konversi jumlah detik ke satuan jam-menit-detik</title>
4 </head>
5 <body>
6   <h1>konversi jumlah detik ke satuan jam-menit-detik</h1>
7   <?php
8   /*
9    Script ini merupakan kebalikan dari script5-3.php
10   Script ini akan mengkonversi waktu yang diketahui dalam satuan detik
11   ke dalam satuan jam-menit-detik.
12   Diketahui waktu dalam detik adalah 15789 detik, akan dikonversi ke
13   bentuk x jam, y menit dan z detik
14   */
15   $totalDetik = 15789; // jumlah total detik mula-mula
16   // memeri waktu dalam jam
17   $sisas = $totalDetik % 3600;
18   $dalamJam = ($totalDetik - $sisas) / 3600;
19
20   // sisa dari perhitungan jam digunakan untuk menghitung menitnya
21   $totalDetik = $sisas;
22   $sisas = $totalDetik % 60;
23   $dalamMenit = ($totalDetik - $sisas) / 60;
24
25   // sisa dalam perhitungan menit digunakan untuk menghitung detiknya
26   $totalDetik = $sisas;
27   $sisas = $totalDetik % 1;
28   $dalamDetik = ($totalDetik - $sisas) / 1;
29
30   echo "<p>Hasil konversinya adalah : " . $dalamJam . " jam : " . $dalamMenit . " menit : " . $dalamDetik . " detik</p>";
31   ?>
32 </body>
33 </html>

```

5.

```

script5-5.php
1  <html>
2  <head>
3      <title>Menghitung selisih dua buah waktu</title>
4  </head>
5  <body>
6      <h1>Menghitung selisih dua buah waktu</h1>
7      <?php
8      /*
9       * Script ini akan mencari selisih antara waktu 10:34:45 dengan 12:25:31.
10      * Hasil selisih waktu dinyatakan dalam detik
11      */
12
13      // Waktu Pertama (12:25:31)
14      $jam1 = 12;
15      $menit1 = 25;
16      $detik1 = 31;
17
18      // Waktu Kedua (10:34:45)
19      $jam2 = 10;
20      $menit2 = 34;
21      $detik2 = 45;
22
23      // Menghitung total detik dari kedua waktu
24      $totalDetik1 = $jam1 * 3600 + $menit1 * 60 + $detik1; // menghitung total detik untuk waktu pertama
25      $totalDetik2 = $jam2 * 3600 + $menit2 * 60 + $detik2; // menghitung total detik untuk waktu kedua
26
27      // Menghitung selisih
28      $selisih = $totalDetik1 - $totalDetik2; // hitung selisih total detik dari kedua waktu
29
30      echo "<p>Selisih dari kedua waktu adalah " . $selisih . " detik</p>";
31  ?>
32 </body>
33 </html>

```

BAB IV

PEMBAHASAN

A. Pembahasan

Pada praktikum ini dipelajari konsep dasar *assignment* dan operator aritmatika dalam PHP. Operator aritmatika digunakan untuk melakukan operasi perhitungan seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Sementara itu, operator *assignment* digunakan untuk memberikan atau memperbarui nilai variabel secara efisien.

Melalui percobaan, implementasi operator *assignment* gabungan terbukti efektif dalam menyederhanakan sintaks saat memperbarui nilai variabel. Selain itu, hasil pengujian pada operator aritmatika menegaskan bahwa PHP mematuhi aturan prioritas operator (perkalian/pembagian dieksekusi sebelum penjumlahan/pengurangan). Dengan menggabungkan *assignment* dan operator aritmatika, peserta memahami cara kerja operator dalam proses pengolahan data, memungkinkan pembuatan program yang mampu menghitung nilai secara otomatis dan dinamis sesuai kebutuhan logika program.

A. Soal

1

```

Soal1.php
1  <?php
2  $saldoAwal = 1000000;
3  $bunga = 0.0025;
4  $bulan = 11;
5
6  // Menghitung total bunga yang diperoleh: (Saldo Awal * Bunga Per Bulan) * Jumlah Bulan
7  $totalBunga = $saldoAwal * $bunga * $bulan;
8
9  // Menghitung saldo Akhir
10 $saldoAkhir = $saldoAwal + $totalBunga;
11
12 echo "Saldo akhir setelah " . $bulan . " bulan adalah : Rp. " . $saldoAkhir . ",-";
13 ?>

```

2

```

Soal2.php
1  <?php
2  $jumlahUang = 1575250;
3
4  // Menghitung Pecahan Rp. 100.000,- ($a)
5  $a = floor($jumlahUang / 100000);
6  $sisas = $jumlahUang % 100000;
7
8  // Menghitung Pecahan Rp. 50.000,- ($b)
9  $b = floor($sisas / 50000);
10 $sisas = $sisas % 50000;
11
12 // Menghitung Pecahan Rp. 20.000,- ($c)
13 $c = floor($sisas / 20000);
14 $sisas = $sisas % 20000;
15
16 // Menghitung Pecahan Rp. 5.000,- ($d)
17 $d = floor($sisas / 5000);
18 $sisas = $sisas % 5000;
19
20 // Menghitung Pecahan Rp. 100,- ($e)
21 $e = floor($sisas / 100);
22 $sisas = $sisas % 100;
23
24 // Menghitung Pecahan Rp. 50,- ($f)
25 $f = floor($sisas / 50);
26
27 echo "Jumlah Rp. 100.000 : " . $a . "<br />";
28 echo "Jumlah Rp. 50.000 : " . $b . "<br />";
29 echo "Jumlah Rp. 20.000 : " . $c . "<br />";
30 echo "Jumlah Rp. 5.000 : " . $d . "<br />";
31 echo "Jumlah Rp. 100 : " . $e . "<br />";
32 echo "Jumlah Rp. 50 : " . $f . "<br />";
33 ?>

```

B. Hasil

1.

Saldo akhir setelah 11 bulan adalah : Rp. 1027500,-

2.

Jumlah Rp. 100.000 : 15
Jumlah Rp. 50.000 : 1
Jumlah Rp. 20.000 : 1
Jumlah Rp. 5.000 : 1
Jumlah Rp. 100 : 2
Jumlah Rp. 50 : 1

BAB V PENUTUP

A. Kesimpulan

Berdasarkan praktikum mengenai *assignment* dan operator aritmatika dalam PHP, dapat ditarik beberapa kesimpulan utama:

1. Operator *Assignment* berfungsi sebagai pondasi untuk inisialisasi dan pembaruan nilai variabel. Penggunaan operator *assignment* gabungan terbukti meningkatkan efisiensi dan keringkasn kode dalam proses pembaruan nilai.
2. Operator Aritmatika adalah inti dari semua proses perhitungan numerik dalam PHP. Operator Modulus sangat krusial untuk menentukan sisa hasil bagi.
3. Pelaksanaan operasi yang kompleks harus mempertimbangkan prioritas operator (seperti perkalian/pembagian dieksekusi sebelum penjumlahan/pengurangan). Tanda kurung wajib digunakan untuk memanipulasi urutan eksekusi dan memastikan akurasi hasil perhitungan.
4. Penguasaan kedua jenis operator ini merupakan prasyarat wajib dalam membangun logika program PHP yang fungsional, dinamis, dan mampu melakukan manipulasi data numerik secara otomatis.

DAFTAR PUSTAKA

Anonim. (2024). *Modul Praktikum Dasar Pemrograman Web*

Sutopo, B., & Permana, A. (2020). *Konsep Dasar Pemrograman Web dengan PHP dan MySQL*.