



JTi - POLINEMA

DASAR PHP

PERTEMUAN 4

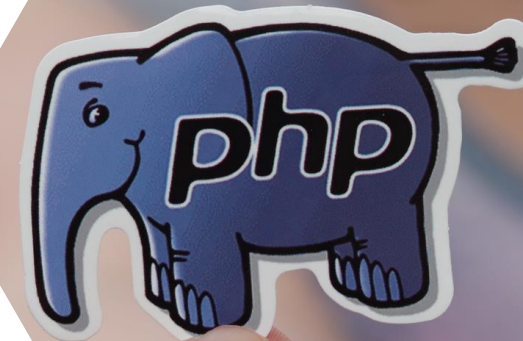
jti.polinema.ac.id

```
$s->rule_exists( $role_id )  
    $access == false )  
    // Remove the rule  
    $details['access'] =  
    $this->_sql->delete(  
    } else {  
        // Update the rule  
        $this->_sql->update(  
    }  
    foreach( $this->rules as  
        if ( $details['role_  
            if ( $access ==  
                unset( $this  
            } else {  
                $this->rules
```

> PHP

PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman server side scripting yang bersifat open source. PHP menjalankan instruksi pemrograman saat proses runtime. PHP merupakan bahasa pemrograman server-side, maka script dari PHP nantinya akan diproses di server.

Jenis server yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, dan LiteSpeed. Selain itu, PHP juga merupakan bahasa pemrograman yang bersifat open source.



PHP

Hypertext Preprocessor

> PHP



1. Cenderung mudah dipelajari

dibanding beberapa bahasa pemrograman populer lain, PHP lebih mudah dipelajari.



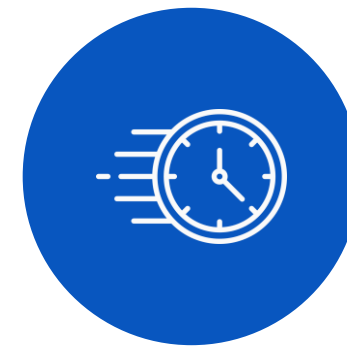
2. Materi belajar yang melimpah

PHP yang “cukup tua” menyebabkan banyak sekali dokumentasi, panduan, dan komunitas aktif bertebaran di jagat maya. Jadi, tak perlu takut jika Anda mengalami kesulitan.



3. PHP bersifat open-source

siapa pun bisa menggunakan PHP tanpa mengeluarkan biaya sepeserpun.



4. Kecepatan tinggi

PHP terbukti bisa meningkatkan kecepatan loading dibanding bahasa lain. Misalnya, lebih cepat tiga kali daripada Python pada beberapa kasus.



5. Banyaknya pilihan database

PHP bisa digunakan di hampir semua jenis database. Mulai dari MySQL, hingga non-relational database seperti Redis



6. Multi-platform

PHP bisa Anda gunakan di macam-macam operating system. Mulai dari Windows, Linux, hingga MacOS.

> Sintaks Dasar PHP

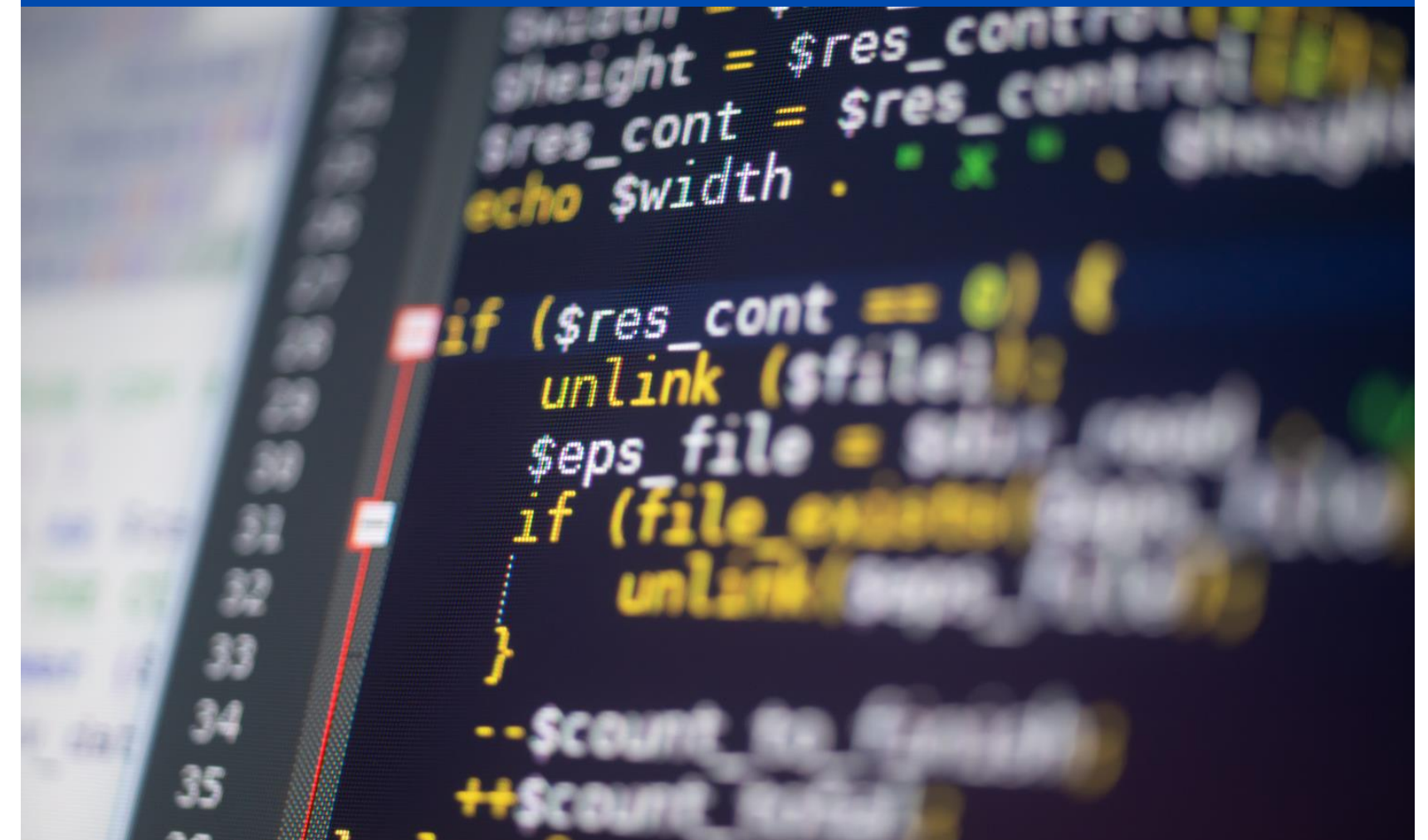
Sintaks dasar PHP dibuka dengan <?php dan ditutup dengan ?> terlihat di contoh berikut:

```
<?php  
    echo "Selamat datang";  
?>
```

Sintaks PHP bersifat case sensitive.

```
<?php  
    $alamat = "Yogyakarta";  
    echo $Alamat;  
?>
```

> Sintaks Dasar PHP



> Sintaks Dasar PHP

Anda juga bisa menuliskan komentar sebagai penjelasan dari kode yang ditulis. Komentar di PHP menggunakan `//` atau `/*` dan tidak akan dimunculkan sebagai output di browser.

```
<?php
// ini contoh penggunaan komentar
echo "Apa Kabar?";
/*
Nah ini juga contoh komentar
yang ditulis
lebih dari satu baris
*/
?>
```

Semua kode PHP yang ditulis harus disimpan dengan file ekstensi **.php**

> Sintaks Dasar PHP



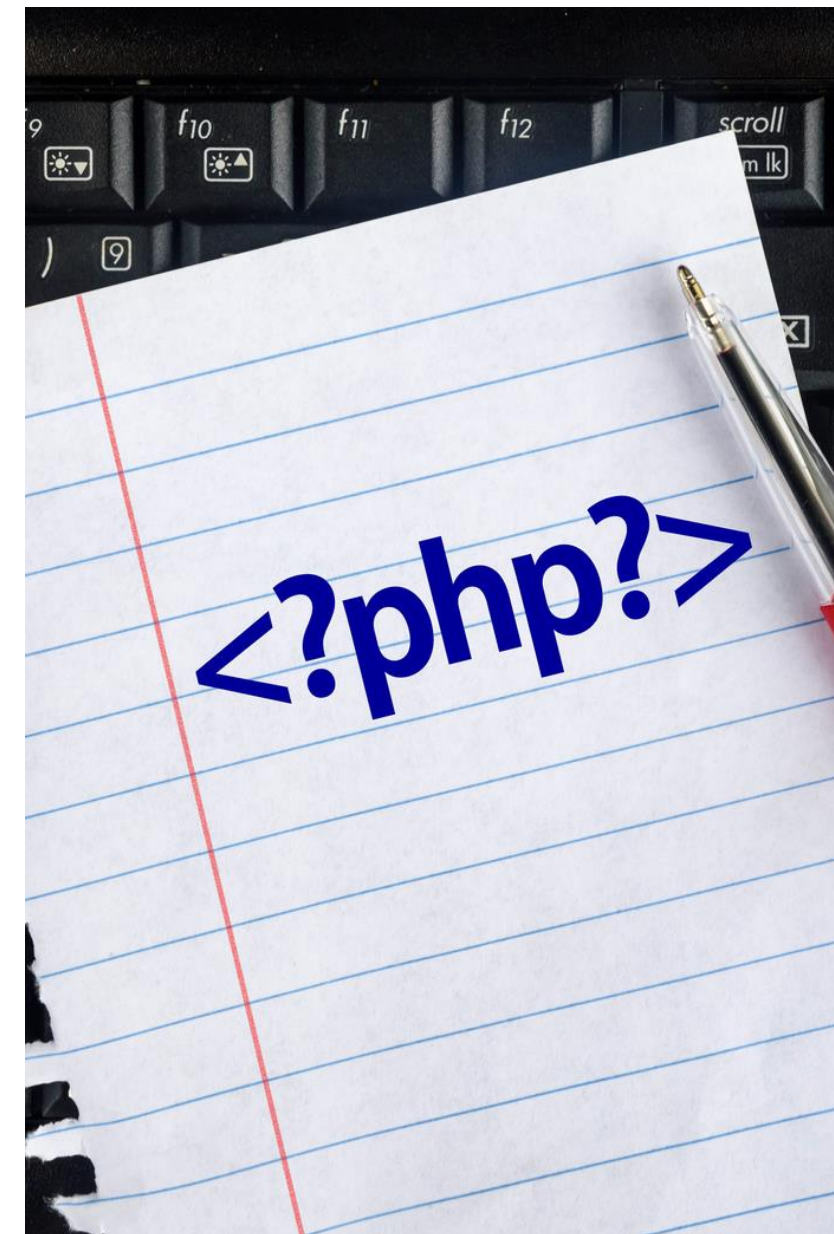
Penulisan Kode-Kode PHP

PHP Native

Native adalah penulisan kode PHP dari nol ketika melakukan perancangan sebuah website. PHP Native sering digunakan oleh developer yang memiliki keahlian coding cukup baik atau mereka yang ingin membuat kerangka alur yang unik dengan fungsionalitas tinggi.

PHP Framework

Ketika menggunakan framework, developer dapat memanfaatkan kerangka pengelolaan website yang sudah jadi. Artinya, tidak perlu membuatnya dari awal sehingga memudahkan pekerjaan. Framework adalah kerangka kerja yang dapat membantu developer bekerja lebih efisien dan menyelesaikan pengembangan website lebih cepat.



> Variabel Pada PHP



1. Deklarasi Variabel

Anda bisa mendeklarasikan variabel dengan mengawali nama variabel dengan tanda dollar (\$) diikuti dengan nama variabel. Misalnya: **\$namaVariabel**



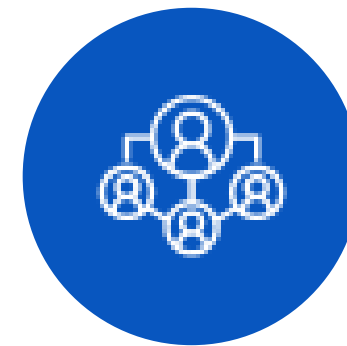
2. Aturan Penamaan Variabel

Nama variabel harus dimulai dengan huruf atau garis bawah (_) dan boleh mengandung huruf, angka, atau garis bawah. Namun, ada perbedaan antara huruf besar dan kecil (case-sensitive). Contoh: **\$nama, \$umur, \$salamat**



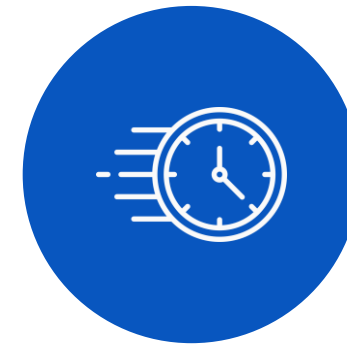
3. Penugasan Nilai

Anda dapat memberikan nilai ke variabel dengan menggunakan operator penugasan (=). Contoh: **\$nama = "John";**



4. Tipe Data

PHP adalah bahasa pemrograman yang tipe datanya dinamis, artinya Anda tidak perlu menginisialisasi tipe data saat mendeklarasikan variabel. Variabel dapat menyimpan berbagai jenis data, seperti string, angka, boolean, array, dan banyak lagi.



5. Mengakses Nilai Variabel

Anda dapat mengakses nilai variabel dengan menggunakan nama variabelnya. Contoh: **\$nama**.

Penerapan Script Variabel

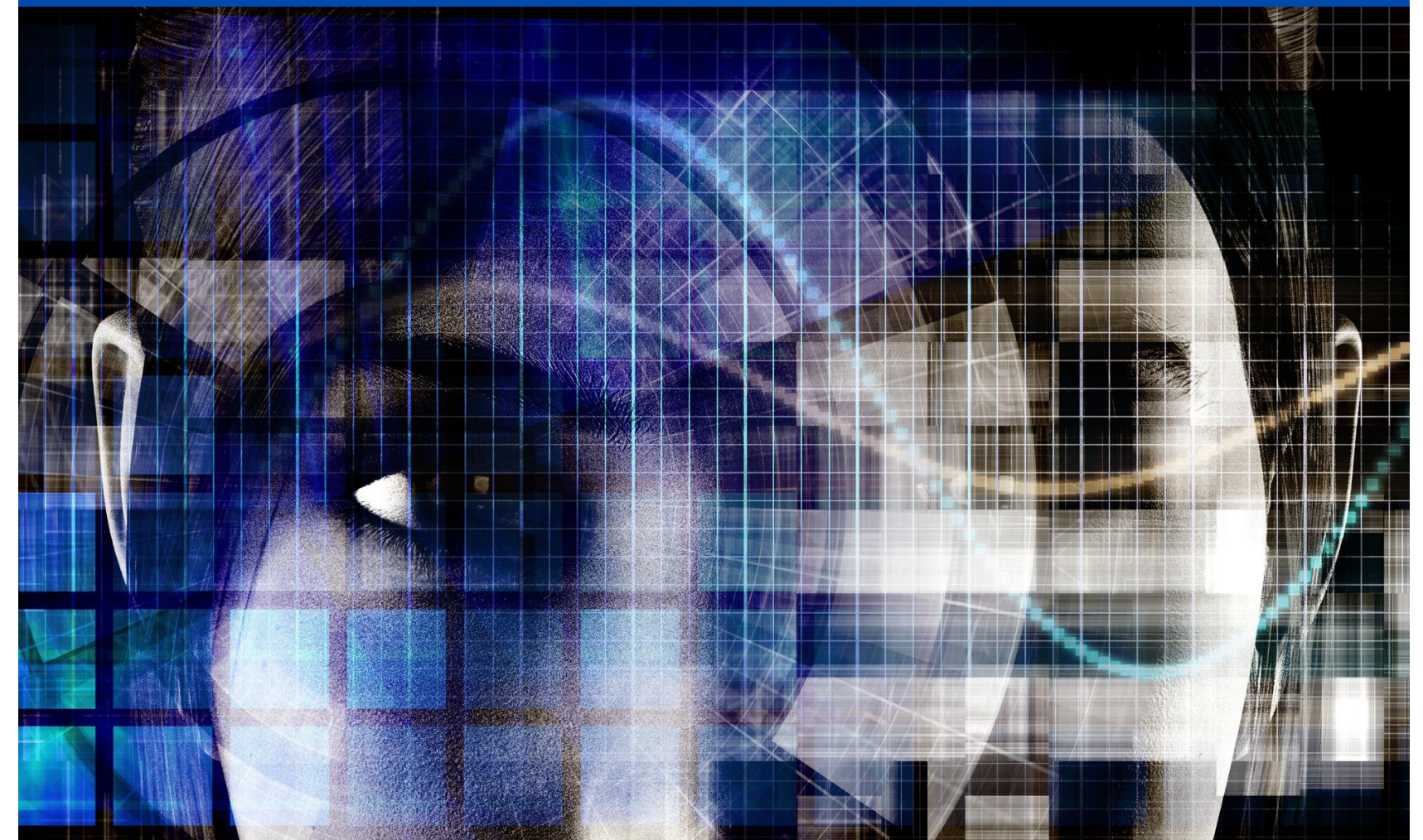
Dalam bentuk string

```
<?php
    $nama = "John";
    $pekerjaan = "Programmer";
    echo "Halo, nama saya $nama dan saya
    seorang $pekerjaan.";
?>
```

Dalam bentuk angka

```
<?php
    $umur = 25;
    $tinggi = 175.5;
    echo "Saya berusia $umur tahun dan tinggi
    saya $tinggi cm.";
?>
```

 Important role in the company



Konstanta Pada PHP

Konstanta adalah nilai yang tetap dan tidak dapat diubah selama jalannya program. Mereka sering digunakan untuk menyimpan nilai-nilai yang tidak boleh berubah selama eksekusi program.



1. Deklarasi Konstanta

Konstanta dideklarasikan menggunakan fungsi `define()` dengan dua parameter: nama konstanta dan nilainya. Contoh: `define("PI", 3.14);`



2. Aturan Penamaan Konstanta

Konstanta umumnya ditulis dalam huruf besar dan menggunakan garis bawah (`_`) sebagai pemisah kata jika nama konstanta terdiri dari beberapa kata. Contoh: `MAX_VALUE`.



3. Tipe Data Konstanta

Konstanta juga bisa menyimpan berbagai jenis data, seperti string, angka, boolean, atau array. Tapi, sekali konstanta didefinisikan, nilainya tidak dapat diubah selama program berjalan.



4. Mengakses Konstanta

Anda dapat mengakses konstanta dengan menggunakan nama konstantanya. Contoh: `echo PI;`

Penerapan Script Konstanta

```
<?php  
define("PI", 3.14159265);  
$radius = 5;  
$luas_lingkaran = PI * ($radius * $radius);  
echo "Luas lingkaran dengan jari-jari $radius adalah  
$luas_lingkaran."  
?>
```



> Tipe Data Pada PHP

Tipe data adalah klasifikasi jenis data atau bentuk dari suatu data. Ia menjelaskan suatu data: dari jenis apakah ia tersusun? Apakah bilangan riil? Atau kah bilangan pecahan? Atau kah ia data yang tersusun dari bentuk karakter? Intinya tipe data adalah klasifikasi jenis dari data yang kita ingin simpan dalam sebuah variabel.

> Tipe Data Pada PHP





Macam-Macam Tipe Data Dalam PHP

Tipe Data	Keterangan
Integer	Berisi bilangan bulat
Float	Berisi bilangan desimal
Boolean	Berisi 2 nilai saja: true dan false
String	Berisi data teks yang diapit oleh tanda " atau ''
Array	Berisi himpunan data
Object	Digunakan untuk membuat objek dari kelas yang telah didefinisikan sebelumnya
NULL	Digunakan untuk menyatakan bahwa suatu variabel tidak memiliki nilai
Callable	Digunakan untuk menyimpan referensi ke fungsi atau metode yang dapat dipanggil

Operator Pada PHP

Operator dalam pemrograman adalah simbol atau tanda yang digunakan untuk melakukan operasi pada data. Operator memungkinkan Anda untuk melakukan berbagai tindakan seperti operasi matematika, perbandingan, logika, dan lainnya pada nilai-nilai yang ada dalam program.

What we do



Jenis-Jenis Operator Pada PHP

Jenis Operator	Simbol	Deskripsi dan Contoh Penggunaan
Aritmatika	+	Penjumlahan: \$hasil = \$angka1 + \$angka2;
	-	Pengurangan: \$hasil = \$angka1 - \$angka2;
	*	Perkalian: \$hasil = \$angka1 * \$angka2;
	/	Pembagian: \$hasil = \$angka1 / \$angka2;
	%	Sisa Bagi: \$sis = \$angka1 % \$angka2;
Pembanding	==	Sama dengan: \$hasil = \$nilai1 == \$nilai2;
	!=	Tidak sama dengan: \$hasil = \$nilai1 != \$nilai2;
	<	Lebih kecil dari: \$hasil = \$nilai1 < \$nilai2;
	>	Lebih besar dari: \$hasil = \$nilai1 > \$nilai2;
	<=	Lebih kecil dari atau sama dengan: \$hasil = \$nilai1 <= \$nilai2;
	>=	Lebih besar dari atau sama dengan: \$hasil = \$nilai1 >= \$nilai2;

Jenis-Jenis Operator Pada PHP

Logika	&& / and	AND (dan): \$hasil = \$a && \$b;
	/ or	OR (atau): \$hasil = \$a \$b;
	! / not	NOT (negasi): \$hasil = !\$a;
Penugasan	=	Penugasan biasa: \$a = \$b;
	+=	Penugasan dengan penjumlahan: \$a += \$b;
	-=	Penugasan dengan pengurangan: \$a -= \$b;
	*=	Penugasan dengan perkalian: \$a *= \$b;
	/=	Penugasan dengan pembagian: \$a /= \$b;
	%=	Penugasan dengan sisa bagi: \$a %= \$b;
Increment/Decrement	++	Increment (menambah 1): \$a++;
	--	Decrement (mengurangi 1): \$a--;

Jenis-Jenis Operator Pada PHP

Bitwise	&	AND bitwise: \$hasil = \$a & \$b;
		OR bitwise: \$hasil = \$a \$b;
	^	XOR bitwise: \$hasil = \$a ^ \$b;
	~	NOT bitwise: \$hasil = ~\$a;
Pemangkatan	**	Pangkat: \$hasil = \$angka ** \$pangkat;
Identik	===	Identik (nilai dan tipe data sama): \$hasil = \$a === \$b;
	!==	Tidak Identik (nilai atau tipe data tidak sama): \$hasil = \$a !== \$b;
Ternary	? :	Operator kondisional (ternary): \$hasil = (\$nilai > 10) ? "Lebih besar" : "Lebih kecil";
Array	[]	Akses elemen array: \$nilai = \$array[0];
	+	Penggabungan array: \$array3 = \$array1 + \$array2;

➤ Struktur Kontrol Pada PHP

Pada dasarnya, pemrograman adalah tentang mengendalikan alur eksekusi program. Struktur kontrol adalah alat yang digunakan untuk memutuskan bagaimana kode akan dieksekusi berdasarkan kondisi-kondisi tertentu.

Pernyataan Kondisional

Pernyataan kondisional digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Ada beberapa jenis pernyataan kondisional dalam PHP:

➤ What we do



➤ Pernyataan Kondisional

➤ if : Digunakan untuk menjalankan blok kode jika kondisi tertentu benar.

```
if ($nilai > 90) {  
    echo "Nilai Anda sangat baik!";  
}
```


➤ Pernyataan Kondisional

➤ **else** : Digunakan bersamaan dengan **if** untuk menjalankan blok kode jika kondisi tidak benar.

```
if ($nilai > 90) {  
    echo "Nilai Anda sangat baik!";  
} else {  
    echo "Anda perlu belajar lebih keras."  
}
```

> Pernyataan Kondisional

> elseif : Digunakan untuk menambahkan kondisi tambahan setelah if.

```
if ($nilai > 90) {  
    echo "A+";  
} elseif ($nilai > 80) {  
    echo "A";  
} else {  
    echo "Belajar lebih keras.";  
}
```

> Pernyataan Kondisional

> **switch:** Digunakan untuk mengevaluasi ekspresi dan menjalankan kode berdasarkan nilai yang cocok.

```
switch ($hari) {  
    case "Senin":  
        echo "Hari kerja.";  
        break;  
    case "Sabtu":  
    case "Minggu":  
        echo "Akhir pekan.";  
        break;  
    default:  
        echo "Hari tidak valid.";  
}
```


> Perulangan

Perulangan memungkinkan Anda untuk mengulang kode beberapa kali berdasarkan kondisi tertentu. Ada tiga jenis perulangan yang paling umum digunakan dalam PHP:

> **for:** Digunakan untuk mengulang kode sejumlah tertentu kali.

```
for ($i = 1; $i <= 5; $i++) {  
    echo "Iterasi ke-$i <br>";  
}
```

> Perulangan

> **while:** Digunakan untuk mengulang kode selama kondisi tertentu benar.

```
$counter = 1;  
while ($counter <= 5) {  
    echo "Iterasi ke-$counter <br>";  
    $counter++;  
}
```

> Perulangan

> **do-while:** Mirip dengan while, tetapi akan menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi.

```
$counter = 1;  
do {  
    echo "Iterasi ke-$counter <br>";  
    $counter++;  
} while ($counter <= 5);
```


➤ Pernyataan Pengendalian Aliran

Pernyataan ini memungkinkan Anda mengontrol aliran eksekusi program dengan cara tertentu:

➤ **break:** Digunakan untuk menghentikan perulangan atau switch saat ini.

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

Iterasi ke-1

Iterasi ke-2

Iterasi ke-3

Iterasi ke-4

➤ Pernyataan Pengendalian Aliran

➤ **continue:** Digunakan untuk melanjutkan iterasi berikutnya dalam perulangan.

```
for ($i = 1; $i <= 5; $i++) {  
    if ($i == 3) {  
        continue; // Melanjutkan ke iterasi berikutnya saat $i = 3  
    }  
    echo $i;  
}
```

Iterasi ke-1

Iterasi ke-2

Iterasi ke-4

Iterasi ke-5

Penggunaan Array Pada PHP

Array Satu Dimensi

Array satu dimensi adalah struktur data yang digunakan untuk menyimpan sejumlah nilai dalam satu variabel. Anda dapat mendeklarasikan array satu dimensi dalam PHP menggunakan beberapa metode:

Metode 1: Menggunakan array()

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

Metode 2: Menggunakan tanda kurung siku [] (Mulai dari PHP 5.4+)

```
$buah = ["apel", "jeruk", "pisang", "mangga"];
```

Anda dapat mengakses nilai dalam array dengan menggunakan indeks (mulai dari 0 untuk indeks pertama):

```
echo $buah[0]; // Output: apel  
echo $buah[2]; // Output: pisang
```


Penggunaan Array Pada PHP

Array Satu Dimensi

Anda dapat menambahkan nilai ke dalam array menggunakan indeks tertentu atau dengan metode `[]` untuk menambahkan nilai baru ke akhir array:

```
$buah[] = "anggur"; // Menambahkan "anggur" ke akhir array  
$buah[4] = "durian"; // Menambahkan "durian" pada indeks 4
```

Anda dapat mengubah nilai dalam array dengan mengakses indeksnya dan menggantinya:

```
$buah[1] = "semangka"; // Mengubah nilai "jeruk" menjadi "semangka"
```

Anda dapat melakukan iterasi (perulangan) melalui semua elemen dalam array menggunakan loop `for`, `foreach`, atau `while`. Berikut contohnya dengan `foreach`:

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

Penggunaan Array Pada PHP

Array Dua Dimensi

Array dua dimensi adalah struktur data yang digunakan untuk menyimpan data dalam bentuk tabel dua dimensi, yang mirip dengan baris dan kolom dalam spreadsheet. Berikut adalah penjelasan singkat tentang array dua dimensi dan contoh penerapannya dalam PHP:

1. Deklarasi Array Dua Dimensi

Anda dapat mendeklarasikan array dua dimensi dalam PHP dengan menyusun array satu dimensi dalam array lainnya:

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

```
$matrix = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
];
```

Penggunaan Array Pada PHP

Array Dua Dimensi

2. Mengakses Nilai dalam Array Dua Dimensi

Anda dapat mengakses nilai dalam array dua dimensi dengan menggunakan dua indeks, satu untuk baris dan satu lagi untuk kolom:

```
echo $matrix[0][0]; // Output: 1 (baris 1, kolom 1)  
echo $matrix[1][2]; // Output: 6 (baris 2, kolom 3)
```

3. Menambahkan dan Mengubah Nilai dalam Array Dua Dimensi

Anda dapat menambahkan atau mengubah nilai dalam array dua dimensi dengan mengakses elemen menggunakan indeks:

```
$matrix[1][1] = 10; // Mengubah nilai pada baris 2, kolom 2 menjadi 10  
$matrix[2][] = 11;  // Menambahkan nilai 11 ke akhir array pada baris 3
```


Penggunaan Array Pada PHP

Array Dua Dimensi

4. Iterasi Melalui Array Dua Dimensi

Anda dapat melakukan iterasi melalui semua elemen dalam array dua dimensi menggunakan nested loop, seperti loop **for**:

```
for ($i = 0; $i < count($matrix); $i++) {  
    for ($j = 0; $j < count($matrix[$i]); $j++) {  
        echo $matrix[$i][$j] . " ";  
    }  
    echo "<br>";  
}
```

1	2	3
4	5	6
7	8	9

Penggunaan Array Pada PHP

Array Multidimensi

Array multidimensi adalah struktur data yang memungkinkan Anda untuk menyimpan data dalam lebih dari dua dimensi. Ini adalah pengembangan dari array dua dimensi yang telah dijelaskan sebelumnya.

1. Deklarasi Array Multidimensi Tiga Dimensi

Anda dapat mendeklarasikan array multidimensi tiga dimensi dengan menambahkan array dalam array dalam array:

```
$threeDArray = array(  
    array(  
        array(1, 2, 3),  
        array(4, 5, 6)  
    ),  
    array(  
        array(7, 8, 9),  
        array(10, 11, 12)  
    )  
);
```

```
$threeDArray = [  
    [  
        [1, 2, 3],  
        [4, 5, 6]  
    ],  
    [  
        [7, 8, 9],  
        [10, 11, 12]  
    ]  
];
```

Penggunaan Array Pada PHP

Array Multidimensi

2. Mengakses Nilai dalam Array Multidimensi Tiga Dimensi

Anda dapat mengakses nilai dalam array multidimensi tiga dimensi dengan menggunakan tiga indeks, masing-masing untuk tingkat dimensi yang sesuai:

```
echo $threeDArray[0][0][0]; // Output: 1 (dimensi 1, dimensi 2, elemen pertama)
echo $threeDArray[1][1][2]; // Output: 12 (dimensi 2, dimensi 2, elemen ketiga)
```

3. Menambahkan dan Mengubah Nilai dalam Array Multidimensi Tiga Dimensi

Anda dapat menambahkan atau mengubah nilai dalam array multidimensi tiga dimensi dengan mengakses elemen menggunakan indeks:

```
$threeDArray[0][1][1] = 15; // Mengubah nilai pada dimensi 1, dimensi 3, elemen kedua
$threeDArray[1][0][] = 13;  // Menambahkan nilai 13 ke akhir array pada dimensi 1
```

Penggunaan Array Pada PHP

Array Multidimensi

4. Iterasi Melalui Array Multidimensi Tiga Dimensi

Untuk mengiterasi melalui array multidimensi tiga dimensi, Anda perlu menggunakan tiga tingkat nested loop:

```
foreach ($threeDArray as $dimensi1) {  
    foreach ($dimensi1 as $dimensi2) {  
        foreach ($dimensi2 as $elemen) {  
            echo $elemen . " ";  
        }  
        echo "<br>";  
    }  
    echo "<br>";  
}
```

1 2 3

4 5 6

7 8 9

10 11 12

Penggunaan Array Pada PHP

Array Asosiatif

Array asosiatif (associative array) adalah tipe array di PHP yang menggunakan pasangan "kunci" (key) dan "nilai" (value) untuk mengorganisir data. Berikut adalah penjelasan dan contoh penerapan array asosiatif dalam PHP:

1. Deklarasi Array Asosiatif

Anda dapat mendeklarasikan array asosiatif dengan menggunakan sintaks `array()` atau tanda kurung siku `[]`. Setiap elemen array asosiatif memiliki kunci (key) yang berpasangan dengan nilai (value)-nya. Berikut adalah contoh penggunaan array asosiatif dalam PHP:

```
// Menggunakan sintaks array()
$student = array(
    "nama" => "John",
    "usia" => 20,
    "kelas" => "12A"
);

// Menggunakan sintaks tanda kurung siku []
$student = [
    "nama" => "John",
    "usia" => 20,
    "kelas" => "12A"
];
```

Penggunaan Array Pada PHP

Array Asosiatif

2. Mengakses Nilai dalam Array Asosiatif

Anda dapat mengakses nilai dalam array asosiatif dengan menggunakan kunci yang sesuai:

```
echo $student["nama"]; // Output: John  
echo $student["usia"]; // Output: 20  
echo $student["kelas"]; // Output: 12A
```

3. Menambahkan dan Mengubah Nilai dalam Array Asosiatif

Anda dapat menambahkan atau mengubah nilai dalam array asosiatif dengan mengakses elemen menggunakan kunci:

```
$student["nilai"] = 95; // Menambahkan nilai baru ke dalam array  
$student["usia"] = 21; // Mengubah nilai usia
```

Penggunaan Array Pada PHP

Array Asosiatif

4. Iterasi Melalui Array Asosiatif

Anda dapat melakukan iterasi (perulangan) melalui semua elemen dalam array asosiatif menggunakan loop foreach. Loop ini memungkinkan Anda mengakses semua pasangan kunci-nilai dalam array asosiatif:

```
foreach ($student as $key => $value) {  
    echo "$key: $value <br>";  
}
```

nama: John
usia: 21
kelas: 12A
nilai: 95



JTi - POLINEMA

TERIMA KASIH

jti.polinema.ac.id

