

LAPORAN TUGAS PEMROGRAMAN WEB
TUGAS KESATU—*Praktikum Modul Kesatu: PHP*



Nama : Theo Jesen Naftali Sirait

NRP : 3125600106

Dosen Pengajar : Prof. M. Udin Harun Al Rasyid S. Kom, Ph. D

PROGRAM STUDI S. Tr. TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA (PENS)

TAHUN 2026

BAB I

PENDAHULUAN

2.1. Dasar Teori

1.1. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side scripting* yang bersifat *open source*. Sebagai bahasa *server-side*, skrip PHP dieksekusi di server web sebelum hasilnya dikirimkan ke peramban atau *browser* pengguna dalam bentuk HTML murni.

Secara objektif, PHP dirancang khusus untuk pengembangan web guna menciptakan konten dinamis, seperti sistem *login*, pengolahan *database*, dan manajemen sesi.

1.2. Tipe Data PHP

<i>Integer</i>	Digunakan untuk menyimpan angka utuh tanpa pecahan atau desimal.
<i>String</i>	Digunakan untuk menyimpan deretan karakter, baik berupa huruf, angka (yang tidak dihitung), maupun simbol.
<i>Float</i>	Juga dikenal sebagai Double, digunakan untuk menyimpan angka yang memiliki bagian desimal atau pecahan.
<i>Boolean</i>	Tipe data paling sederhana yang hanya memiliki dua kemungkinan nilai: Benar atau Salah.

1.3. Sintaks PHP

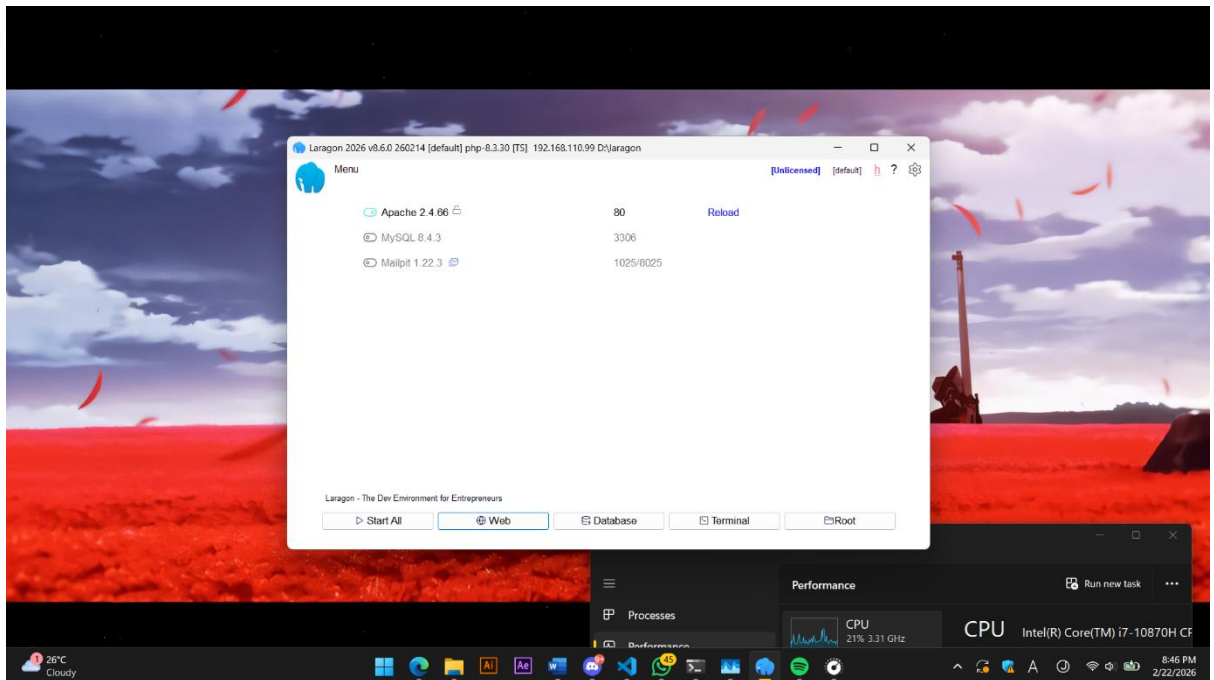
<code><?php ... ?></code>	Tag pembuka dan penutup kode PHP
<code>// dan /**/</code>	Komentar: tidak dieksekusi, hanya untuk catatan
<code>\$nama</code>	Tanda \$ digunakan untuk mendeklarasikan variabel
<code>=</code>	Operator assignment (mengisi nilai ke variabel)
<code>Echo</code>	Construct untuk menampilkan output (bisa multiple parameter)
<code>print</code>	Function untuk menampilkan output (hanya satu parameter)
<code>.</code>	Operator concatenation (penggabung string)
<code>"\$umur tahun"</code>	Variabel dalam string petik dua akan diparsing nilainya
<code>{\$tinggi_badan}</code>	Kurung kurawal memisahkan variabel dari teks di sekitarnya
<code>var_dump()</code>	Function untuk menampilkan informasi tipe data dan nilai
<code>+, /</code>	Operator aritmatika (penjumlahan, pembagian)
<code>>=</code>	Operator perbandingan (lebih besar atau sama dengan)
<code>if (kondisi) { ... }</code>	Struktur kondisi jika kondisi bernilai true
<code>elseif</code>	Alternatif kondisi jika if sebelumnya false
<code>else</code>	Blok yang dijalankan jika semua kondisi false
<code>switch (\$variabel)</code>	Seleksi berdasarkan nilai variabel
<code>case "nilai":</code>	Membandingkan nilai variabel dengan konstanta
<code>break</code>	Menghentikan eksekusi switch
<code>default</code>	Dijalankan jika tidak ada case yang cocok
<code>\$array = [nilai1, nilai2]</code>	Cara modern membuat array indexed
<code>\$array = ["key" => "value"]</code>	Array asosiatif dengan key custom
<code>for (inisialisasi; kondisi; increment)</code>	Perulangan dengan jumlah iterasi pasti
<code>\$i++</code>	Increment: menambah nilai variabel 1
<code>count(\$array)</code>	Function menghitung jumlah elemen array
<code>foreach (\$array as \$value)</code>	Perulangan khusus untuk array (hanya nilai)
<code>foreach (\$array as \$key => \$value)</code>	Perulangan array dengan key dan value
<code>while (kondisi) { ... }</code>	Perulangan selama kondisi true
<code>define("NAMA", "value")</code>	Membuat konstanta global
<code>const NAMA = "value"</code>	Cara lain membuat konstanta (lebih cepat)
<code>function namaFungsi(\$param) { }</code>	Deklarasi fungsi
<code>return</code>	Mengembalikan nilai dari fungsi
<code>\$nama = "default"</code>	Parameter default jika tidak dikirim argumen
<code>NamaKonstanta</code>	Konstanta dipanggil tanpa tanda \$
<code>\$_SERVER</code>	Superglobal array berisi informasi server
<code>\$_SERVER["PHP_SELF"]</code>	Nama file script yang sedang dieksekusi
<code>\$_SERVER["REMOTE_ADDR"]</code>	Alamat IP pengunjung
<code>preg_match(\$pola, \$string)</code>	Function mencocokkan string dengan pola regex
<code>/^...\$/</code>	Delimiter regex: ^ awal string, \$ akhir string
<code>[a-zA-Z0-9_-]+</code>	Character class: huruf, angka, titik underscore, strip (min 1x)
<code>@</code>	Karakter literal (harus ada @)
<code> \.</code>	Escape karakter untuk titik literal
<code>{2,}</code>	Kuantifier: minimal 2 karakter (untuk domain .com, .id, dll)

BAB II

PERCOBAAN

2.1. Percobaan Pertama—Instalasi Server PHP (Laragon)

2.1.1. *Output:*



2.1.2. Penjelasan:

Laragon adalah *local server environment* yang digunakan untuk menjalankan skrip PHP di komputer lokal (*Windows*). Fungsi utamanya adalah menyediakan paket lengkap yang terdiri dari Apache/Nginx (*Web Server*), MySQL (*Database*), dan PHP *Interpreter*. Tanpa server lokal seperti Laragon, kode PHP tidak dapat dieksekusi karena PHP memerlukan *environment* khusus untuk memproses skrip sebelum ditampilkan di browser.

2.2. Percobaan Kedua—Pengenalan Dasar (*variables*, Tipe Data, *echo*)

2.2.1. *Source Code*:

```
<?php
echo "<style>html {text-align: center;font-family:'Sans Serif', Calibri, sans-serif;}</style>";

// ini single line comment

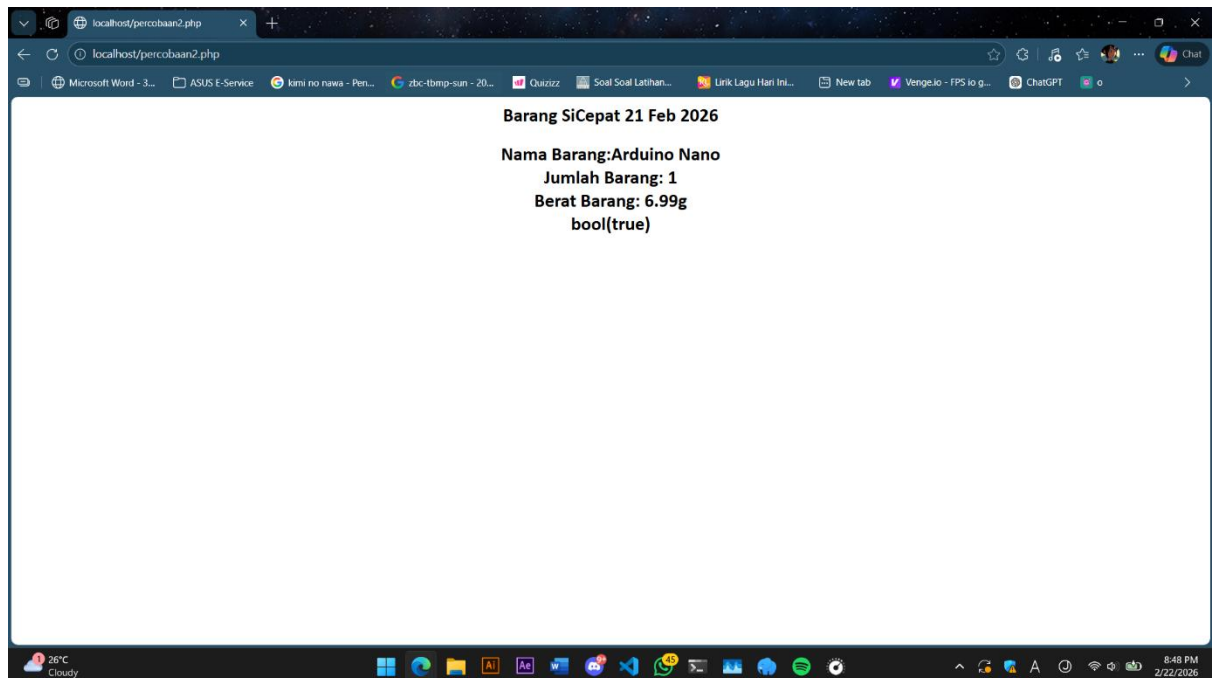
/*Kalo ini Multiline comment
coment */

// Declaration berdasarkan tipe data
$productName = "Arduino Nano";
$quantity = 1;
$weight = 6.99;
$isDelivered = true;

// Output
echo "<h2> Barang SiCepat 21 Feb 2026 <h2>";
print "Nama Barang:". $productName. "<br>"; //concatenation
echo "Jumlah Barang: $quantity<br>"; //Variabel di dalam string
echo "Berat Barang: {$weight}g<br>"; //kurawal untuk menentukan akhir dari variabel

// data type check
var_dump($isDelivered); //return bool value
?>
```

2.2.2. Output:



2.2.3. Penjelasan:

Pengenalan dasar dimulai dengan memahami variabel sebagai wadah penyimpanan data sementara yang ditandai dengan simbol dolar (\$) dan memiliki sifat *case-sensitive*. Setiap variabel membawa tipe data tertentu, seperti *string* untuk teks, *integer* untuk bilangan bulat, atau *boolean* untuk nilai logika, yang menentukan bagaimana data tersebut akan diproses oleh mesin. Sementara itu, perintah *echo* digunakan sebagai instruksi dasar untuk mencetak atau menampilkan hasil pengolahan data tersebut ke layar pengguna dalam format teks atau HTML.

2.3. Percobaan Ketiga—Operator dan Struktur Kondisi(*if..else* & *switch*)

2.3.1. Source Code:

```
<?php
echo "<style>html {text-align: center;font-family:'Sans Serif', Calibri, sans-serif;}</style>";
$deliveredItemsLastMonth = 2500;
$deliveredItemsThisMonth = 2000;

// mean
$totalItemsDelivered = ($deliveredItemsLastMonth + $deliveredItemsThisMonth) /2;

echo "Total Barang Terkirim: $totalItemsDelivered <br>";

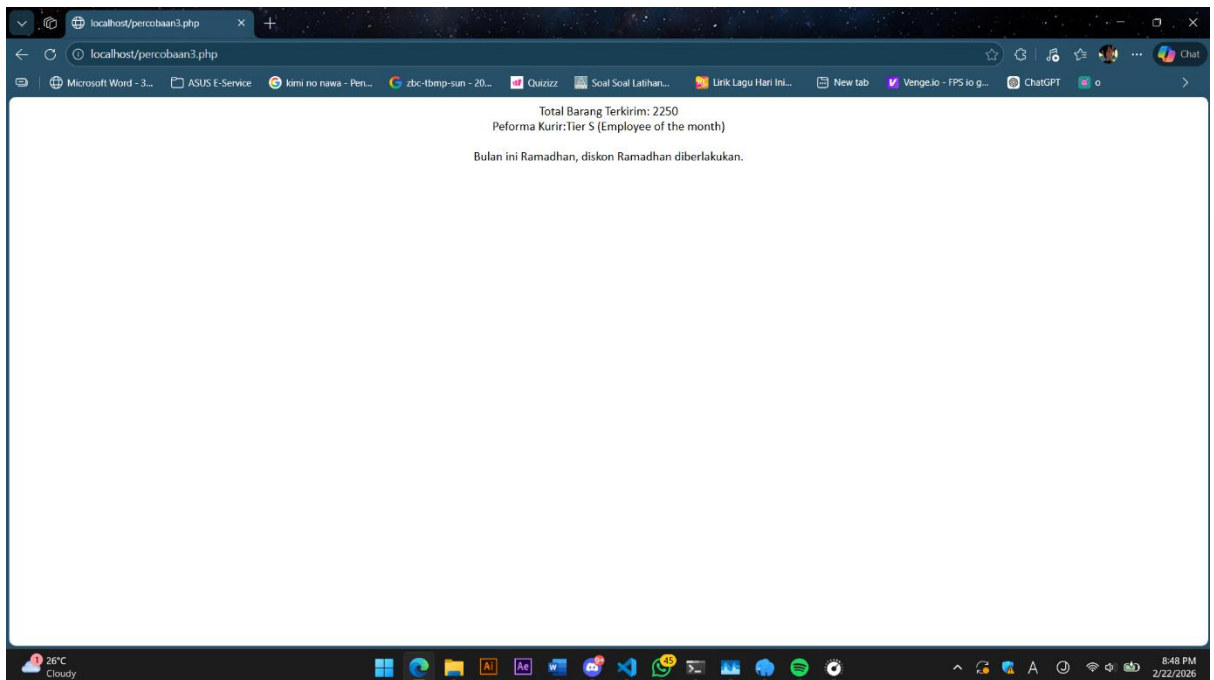
// Menggunakan conditional state
echo "Peforma Kurir:";

if ($totalItemsDelivered >= 2000)
    echo "Tier S (Employee of the month)";
elseif ($totalItemsDelivered >= 1500)
    echo "Tier A (Approved)";
elseif ($totalItemsDelivered >= 1000)
    echo "Tier B (Approved)";
elseif ($totalItemsDelivered >= 500)
    echo "Tier C (Warning Letter)";
else
    echo "Tier D (Dipecat)";

echo "<br><br>";

// Switch State
$bulan = "Ramadhan";
switch($bulan){
    case "Ramadhan":
        echo "Bulan ini $bulan, diskon $bulan diberlakukan.";
        break;
    case "Adven":
        echo "Bulan ini $bulan, diskon $bulan diberlakukan.";
        break;
    default:
        echo "Tidak ada Diskon bulan ini.";
};
?>
```

2.3.2. Output:



2.3.3. Penjelasan:

Operator digunakan untuk melakukan manipulasi data, mulai dari operasi aritmatika hingga perbandingan logika yang menghasilkan nilai benar atau salah. Logika tersebut kemudian diterapkan dalam struktur kondisi seperti *if..else* dan *switch* untuk menentukan arah alur program berdasarkan kriteria yang diberikan. Struktur *if..else* sangat fleksibel untuk menangani rentang kondisi yang kompleks, sedangkan *switch* digunakan sebagai alternatif yang lebih rapi ketika program harus memilih salah satu dari banyak nilai konstanta yang sudah ditentukan.

2.4. Percobaan Keempat—Perulangan (*loops*) dan *array*

2.4.1. *Source Code:*

```
<?php
echo "<style>html {text-align: center; display:flex; justify-content:center;}</style>";
// Array indexed
$jabodetabekSector = ["Jakarta", "Bogor", "Depok", "Tangerang", "Bekasi"];

echo "<h3>Daftar Wilayah Jabodetabek</h3>";
echo "<ul>";

for ($i=0; $i < count($jabodetabekSector); $i++) {
    echo "<li> $jabodetabekSector[$i] </li>";
}

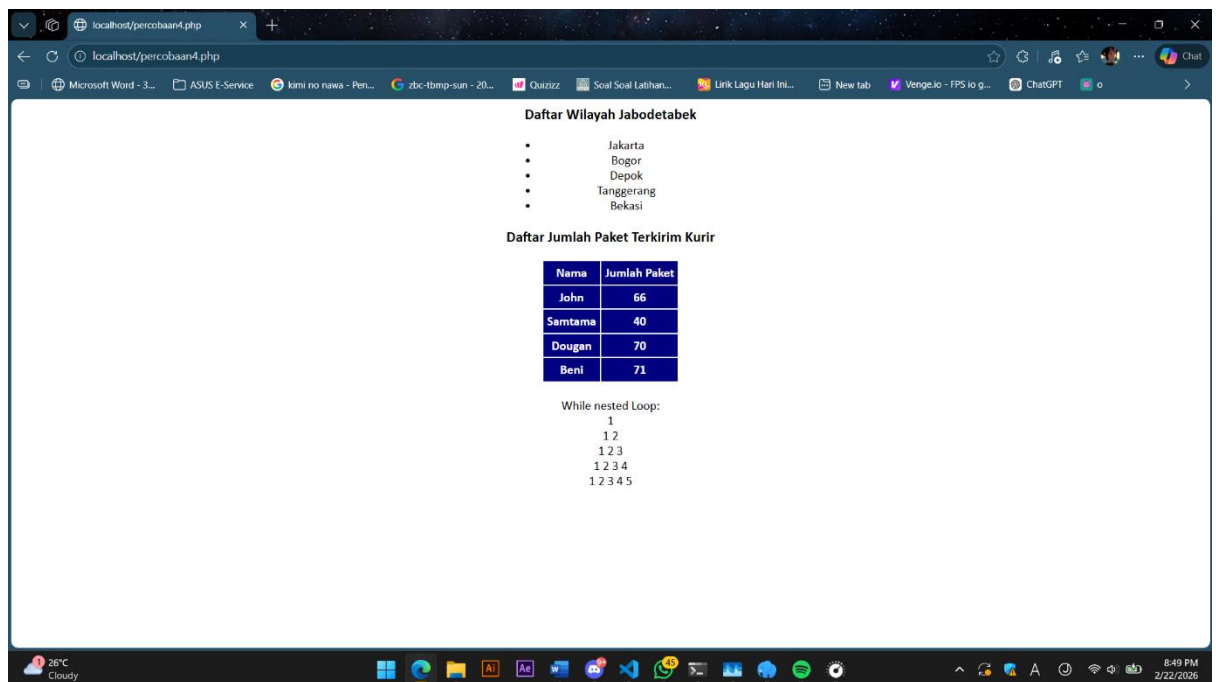
echo "</ul>";

// Array Associative
$itemDeliveredbyPorter = [
    "John" => 66,
    "Samtama" => 40,
    "Dougan" => 70,
    "Beni" => 71];

echo "<h3> Daftar Jumlah Paket Terkirim Kurir</h3> ";
echo "<table border= '0.5' cellpadding= '5'>";
echo "<tr><style>html{font-family:'Sans Serif, Calibri, sans-serif;}table{margin:auto;}tr{text-align: center; background-color: navy; color: white; font-weight:bold;}</style><th>Nama</th><th>Jumlah Paket</th></tr>";
foreach ($itemDeliveredbyPorter as $nama => $jumlahItem) {
    echo "<tr><td>$nama</td><td>$jumlahItem</td></tr>";
}
echo "</table>";

// while loop untuk increment
echo "<br>While nested Loop: <br>";
$counter = 1;
for ($i=$counter; $i <=5 ; $i++) {
    for ($j=$counter;$j <= $i;$j++){
        echo "$j ";
    }
    echo "<br>";
}
?>
```

2.4.2. Output:



2.4.3. Penjelasan:

Perulangan atau *loops* berfungsi untuk menjalankan blok kode secara berulang selama kondisi tertentu terpenuhi, yang sangat efektif untuk mengotomatisasi tugas yang repetitif. Mekanisme ini sering dipadukan dengan *array*, yaitu struktur data yang mampu menyimpan banyak nilai dalam satu nama variabel tunggal melalui sistem indeks. Dengan menggunakan perulangan seperti *foreach*, pengembang dapat menelusuri setiap elemen di dalam array secara efisien untuk ditampilkan atau diolah lebih lanjut tanpa harus menulis kode secara manual satu per satu.

2.5. Percobaan Kelima—Fungsi dan Konstanta

2.5.1. *Source Code:*

```
<?php
echo "<style>html {text-align: center;font-family:'Sans Serif', Calibri, sans-serif;}</style>";
// Konstanta
define("SERVICE_NAME", "SiCepat");
const TOTAL_TIME = 10080;

// membuat function
const itemBesar = 10000;
const itemSedang =7000;
const itemKecil =5000;

function totalPendapatanKurir ($jumlahItem, $itemsType){
    $total = $jumlahItem * $itemsType;
    return $total;
}

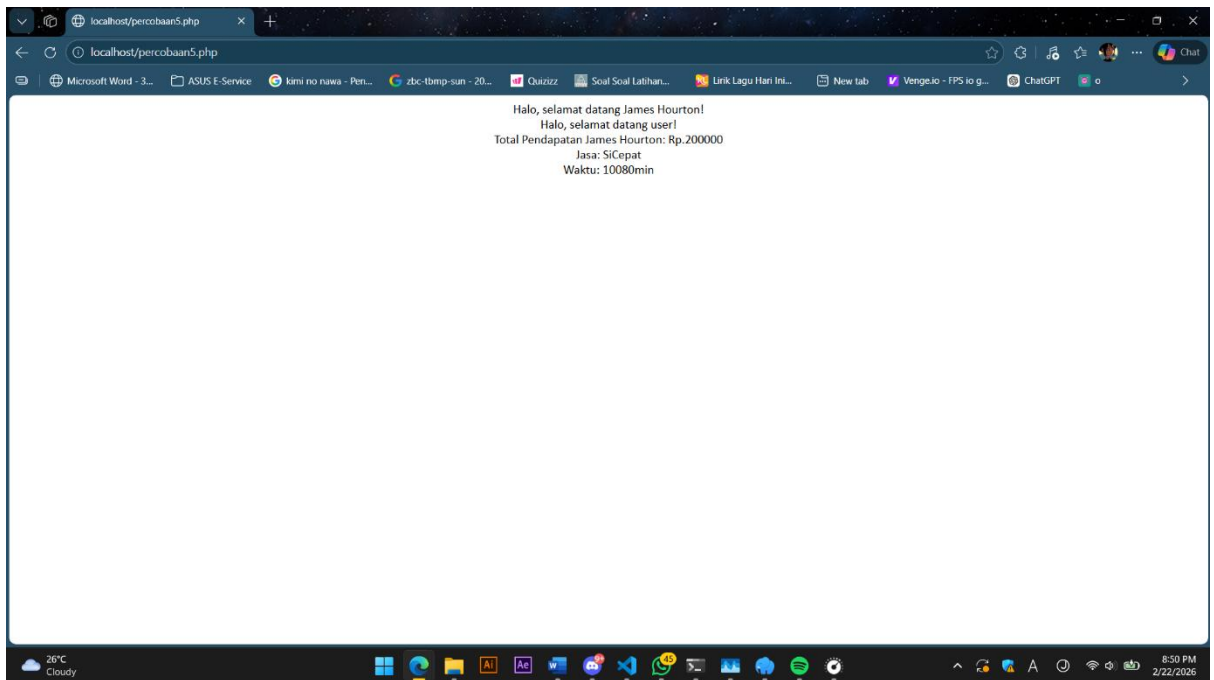
// Membuat fungsi parameter default
function dashboardIntroClient($name="user"){
    return "Halo, selamat datang ".$name . "!";
}

// menggunakan fungsi
echo dashboardIntroClient("James Hourton")."<br>";
echo dashboardIntroClient()."<br>"; //menggunakan value default

$totalPendapatanJamesHourton = totalPendapatanKurir(20, itemBesar);
echo "Total Pendapatan James Hourton: Rp. " . $totalPendapatanJamesHourton."<br>";

echo "Jasa: ". SERVICE_NAME."<br>";
echo "Waktu: ". TOTAL_TIME. "min";
?>
```

2.5.2. Output:



2.5.3. Penjelasan:

Fungsi merupakan blok kode terpisah yang dirancang untuk melakukan tugas spesifik dan dapat dipanggil berulang kali, sehingga meningkatkan efisiensi serta keterbacaan kode (*reusability*). Di sisi lain, konstanta digunakan untuk menyimpan nilai tetap yang tidak akan berubah selama seluruh siklus eksekusi program berlangsung. Berbeda dengan variabel biasa, konstanta didefinisikan menggunakan fungsi *define()* atau kata kunci *const* dan tidak memerlukan simbol \$ dalam penulisannya, menjadikannya ideal untuk menyimpan pengaturan konfigurasi yang bersifat permanen.

2.6. Percobaan Keenam—*Superglobals* dan *Reguler Expression (RegEx)*

2.6.1. *Source Code:*

```
<?php
echo "<style>html {text-align: center;font-family:'Sans Serif', Calibri, sans-serif;}</style>";
// Superglobals
echo "<h3>Informasi Server:</h3>";
echo "file name: " . $_SERVER['PHP_SELF'] . "<br>";
echo "IP address Client: " . $_SERVER['REMOTE_ADDR'] . "<br>";
echo "User Agent: " . $_SERVER['HTTP_USER_AGENT'] . "<br>";
echo "<hr>";

// latihan regex[validasi passowrd]

$password1 = "Naftali@123";
$password2 = "NaftaliSirait";

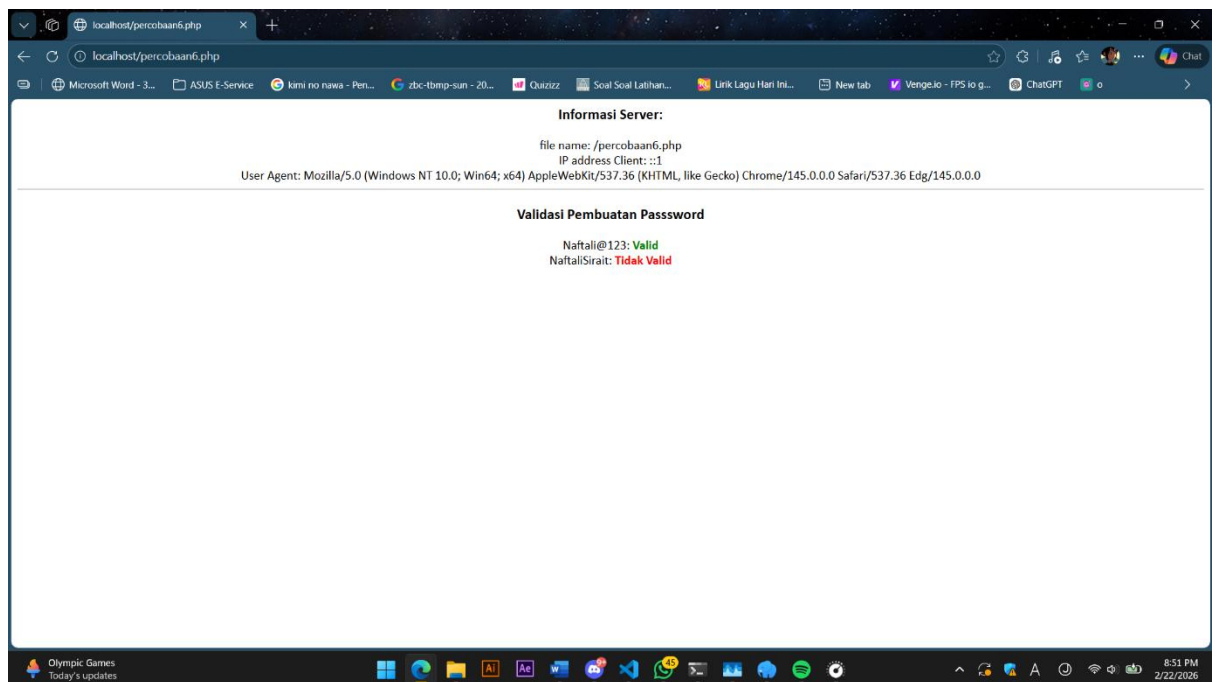
// validator
$pola_password = "/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@!%*?&.-])[A-Za-z\d@$!%*?&.-]{7,}$/";

echo "<h3>Validasi Pembuatan Passsword</h3>";

// pass1
if (preg_match($pola_password,$password1))
    echo "$password1: <span style='font-weight:bold'><span style='color:green'>Valid</span></span><br>";
else
    echo "$password1: <span style='font-weight:bold'><span style='color:red'>Tidak Valid</span></span><br>";

// pass2
if (preg_match($pola_password,$password2))
    echo "$password2: <span style='font-weight:bold'><span style='color:green'>Valid</span></span><br>";
else
    echo "$password2: <span style='font-weight:bold'><span style='color:red'>Tidak Valid</span></span><br>";
?>
```

2.6.2. Output:



2.6.3. Penjelasan:

Superglobals adalah variabel bawaan PHP yang memiliki cakupan global, sehingga dapat diakses dari bagian mana pun dalam skrip tanpa deklarasi khusus, seperti `$_GET` dan `$_POST` yang berfungsi menangkap data input dari pengguna. Untuk memastikan data tersebut valid, digunakan *Regular Expression (RegEx)* sebagai metode pencarian pola karakter yang kompleks dalam sebuah *string*. Kombinasi keduanya memungkinkan pengembang untuk membangun aplikasi yang interaktif sekaligus aman melalui proses validasi format data, seperti pengecekan alamat email atau pola kata sandi yang sesuai standar.

BAB III

TUGAS PRAKTIKUM

3.1. Aplikasi Daftar Nilai Mahasiswa Sederhana

3.1.1. *Source Code:*

```
<?php
$mahasiswa = [
    ["nama" => "Albert", "nim" => "31256000101", "uts" => 70, "uas" => 94, "tugas" =>
    91],
    ["nama" => "Nikola", "nim" => "31256000102", "uts" => 93, "uas" => 98, "tugas" =>
    29],
    ["nama" => "Isaac", "nim" => "31256000103", "uts" => 87, "uas" => 95, "tugas" =>
    94],
    ["nama" => "Feynman", "nim" => "31256000104", "uts" => 88, "uas" => 86, "tugas" =>
    81],
    ["nama" => "Heisenberg", "nim" => "31256000105", "uts" => 40, "uas" => 5, "tugas" =>
    91],
    ["nama" => "Schrodinger", "nim" => "31256000106", "uts" => 1, "uas" => 19, "tugas" =>
    9],];
$head = ['ID', 'Nama', 'NIM', 'UTS', 'UAS', 'Tugas', 'Rata-rata', 'Grade'];

// calc function
function countCGPA($uts, $uas, $tugas){
    $nilaiAkhir = 0.2 * $uts + 0.3 * $uas + 0.5 * $tugas;
    return $nilaiAkhir;
}

// gradeValidator
function gradeValidator($nilaiAkhir){
    if ($nilaiAkhir >= 85 && $nilaiAkhir <= 100)
        return "A";
    elseif($nilaiAkhir >= 70 && $nilaiAkhir < 85)
        return "B";
    elseif($nilaiAkhir >= 60 && $nilaiAkhir < 70)
        return "C";
    elseif($nilaiAkhir >= 50 && $nilaiAkhir < 60)
        return "D";
    elseif($nilaiAkhir >= 0 && $nilaiAkhir < 50)
        return "E";
    else
        return "Nilai tidak bisa divalidasi!";
};
```

```
// output
echo "<style>html{display:flex;justify-content:center;margin:auto;font-family:'Sans Serif, Calibri, sans-serif;table{margin:auto; text-align:center;background-color:red;}th{color:white;}tr{color:#000000;}</style>";
echo "<h1>Data Mahasiswa Harvard Institute of Zurich</h1>";
echo "<table cellpadding= '10'>";
echo "<thead style='background-color:#f97316'><tr>";
for ($i=0;$i<count($head);$i++){
    echo "<th>{$head[$i]}</th>";
};
echo "</tr></thead>";

foreach ($mahasiswa as $i => $data) {
    $nilaiAkhir = countCGPA($data['uts'],$data['uas'],$data['tugas']);
    $colorGradeValidator = ($nilaiAkhir < 70) ? 'color:red':'color:#22c55e';
    $rowValidator = ($i%2 == 0) ? 'background-color:#fef9c3' : 'background-color:#fde68a';
    echo "<tr
style=$rowValidator><td>".$i+1."</td><td>{$data['nama']}</td><td>{$data['nim']}</td><td>{$data['uts']}</td
><td>{$data['uas']}</td><td>{$data['tugas']}</td><td>{$nilaiAkhir}</td><td style=$colorGradeValidator><span
style='font-weight:bold'>".gradeValidator($nilaiAkhir)."</span></td></tr>";
};
echo "</table>";
?>
```

3.1.2. Output:

ID	Nama	NIM	UTS	UAS	Tugas	Rata-rata	Grade
1	Albert	31256000101	70	94	91	87.7	A
2	Nikola	31256000102	93	98	29	62.5	C
3	Isaac	31256000103	87	95	94	92.9	A
4	Feynman	31256000104	88	86	81	83.9	B
5	Heisenberg	31256000105	40	5	91	55	D
6	Schrodinger	31256000106	1	19	9	10.4	E

3.1.3. Penjelasan

Skrip ini menggunakan *Array* Multidimensi *\$mahasiswa* untuk menyimpan sekumpulan data terstruktur, di mana setiap elemen di dalamnya merupakan *array* asosiatif yang memuat informasi nama, NIM, serta nilai akademik mahasiswa. Secara objektif, penggunaan struktur ini memudahkan pengelolaan data massal yang memiliki atribut seragam. Untuk memproses data tersebut, dibuat dua fungsi kustom (*user-defined functions*): *countCGPA()* yang berfungsi melakukan kalkulasi matematis dengan bobot tertentu (20% UTS, 30% UAS, 50% Tugas), dan *gradeValidator()* yang menerapkan struktur kondisi bersyarat (*if..elseif*) untuk mengonversi nilai angka menjadi nilai huruf (A hingga E) berdasarkan rentang yang telah ditetapkan.

Proses penyajian data dilakukan dengan mengintegrasikan logika PHP ke dalam elemen tabel HTML. Penggunaan perulangan *foreach* memungkinkan sistem untuk menelusuri setiap baris data mahasiswa secara otomatis tanpa harus menulis kode secara repetitif. Di dalam perulangan tersebut, diterapkan logika *Ternary Operator* untuk mengatur estetika tampilan secara dinamis, seperti *rowValidator* untuk memberikan warna belang-belang (zebra cross) pada baris tabel dan *colorGradeValidator* untuk memberikan penanda warna merah secara otomatis pada mahasiswa yang mendapatkan nilai di bawah standar (di bawah 70). Hal ini menunjukkan bagaimana PHP dapat digunakan secara efektif untuk menghasilkan antarmuka web yang dinamis dan informatif berdasarkan hasil pengolahan logika di sisi server.

DOKUMENTASI

naftali.it.student.pens.ac.id