**LOGICAL TEST**

**1. ARRAY LOOP (1\_array\_loop.php)**

Lanjutkan skrip berikut dengan menggunakan perulangan **while** untuk menampilkan semua data pada array ke web browser:



Program :

<?php

$aplikasi[1] = 'gtAkademik';

$aplikasi[2] = 'gtFinansi';

$aplikasi[3] = 'gtPerizinan';

$aplikasi[4] = 'eCampuz';

$aplikasi[5] = 'eOviz';

// Inisialisasi variabel indeks

$index = 1;

// Selama indeks kurang dari atau sama dengan jumlah elemen dalam array

while ($index <= count($aplikasi)) {

    // Mendapatkan nilai aplikasi berdasarkan indeks

    $value = $aplikasi[$index];

    // Menampilkan data ke browser

    echo "Aplikasi ke-$index: $value<br>";

    // Meningkatkan indeks

    $index++;

}

Output :

A computer screen with a white background

Description automatically generated with medium confidence

Pembahasan :

Kode program 1\_array\_loop.php adalah penggunaan perulangan `while` dalam PHP untuk menampilkan semua data dalam array. Mari kita bahas langkah demi langkah:

1. Mendefinisikan array `$aplikasi` yang berisi beberapa nama aplikasi. Setiap elemen dalam array memiliki indeks numerik, dimulai dari 1.

2. Menginisialisasi variabel `$index` dengan nilai 1. Ini akan digunakan sebagai indeks untuk mengakses elemen-elemen dalam array.

3. Kemudian, memulai perulangan `while`. Perulangan ini akan berlanjut selama nilai `$index` kurang dari atau sama dengan jumlah elemen dalam array, yang dihitung dengan `count($aplikasi)`.

4. Di dalam perulangan, mengambil nilai aplikasi dari array menggunakan indeks saat ini, yaitu `$aplikasi[$index]`, dan menyimpannya dalam variabel `$value`.

5. Selanjutnya, mencetak pesan ke browser yang berisi nomor aplikasi (indeks) dan nama aplikasi yang sesuai.

6. Setelah mencetak nilai, dapat meningkatkan nilai `$index` dengan `$index++` agar perulangan dapat melanjutkan ke elemen berikutnya.

7. Proses ini diulangi hingga seluruh elemen dalam array `$aplikasi` telah ditampilkan.

Hasil keluaran program ini akan mencetak semua nama aplikasi beserta nomor urutnya ke browser. Perulangan `while` digunakan di sini untuk secara dinamis menampilkan elemen-elemen dalam array tanpa harus menentukan panjang array secara eksplisit. Ini adalah salah satu cara penggunaan perulangan dalam PHP untuk mengakses dan memproses elemen dalam array.

**2. MYSQL QUERY (2\_mysql\_query.sql)**

Buatlah **satu buah query** untuk mencari **nama seorang mahasiswa** yang memiliki **nilai**

**mata kuliah tertinggi** pada mata kuliah dengan kode **“MK303”**.

Table: tb\_mahasiswa



Table: tb\_mahasiswa\_nilai



Table: tb\_matakuliah



Output :

A screenshot of a computer program

Description automatically generated

Pembahasan :

Hasil output dapat dilakukan dengan membuat Query SQL relasi hubungan antar tabel tb\_mahasiswa, tb\_mahasiswa\_nilai dan tb\_matakuliah lalu mencari dan menambil data sesuai permintaan kuosioner berdasarkan hubungan ralasi antar tabel atau membuat Query seperti gambar diatas.

**3. OPERATOR LOGIC (****3\_operator\_logic.php)**

Buatlah sebuah fungsi yang berisi algoritma untuk operasi pembagian bilangan bulat positif. **Tidak diperbolehkan menggunakan operator maupun function pembagian** **seperti:**

● **“div”**

● **“%”**

● **“/”**

● **“intdiv()”**

**● “bcdiv()”**

Output hasil bagi adalah bilangan bulat saja, tidak perlu pecahan desimal.

Contoh: 7 : 2 menghasilkan output 3 atau 8 : 4 menghasilkan output 2.

Program :

<?php

function divisionWithoutOperator($numerator, $denominator) {

    // Inisialisasi hasil bagi

    $result = 0;

    // Lakukan pengurangan berulang-ulang hingga numerator lebih kecil dari denominator

    while ($numerator >= $denominator) {

        $numerator -= $denominator;

        $result++;

    }

    return $result;

}

// Contoh penggunaan

$hasil1 = divisionWithoutOperator(7, 2);

$hasil2 = divisionWithoutOperator(8, 4);

echo "7 : 2 menghasilkan output : $hasil1" . PHP\_EOL; // Output: 3

echo "atau" . PHP\_EOL; // Baris pemutus

echo "8 : 4 menghasilkan output : $hasil2" . PHP\_EOL; // Output: 2

?>

Output :

A screenshot of a computer

Description automatically generated

Pembahasan :

Program 3\_operator\_logic.php adalah tentang pembagian bilangan bulat positif tanpa menggunakan operator pembagian seperti `/` atau fungsi pembagian bawaan seperti `div`. Ini adalah contoh cara melakukan operasi pembagian bilangan bulat dengan operator pengurangan dalam PHP. Berikut pembahasan singkatnya:

1. Fungsi `divisionWithoutOperator` menerima dua parameter, yaitu `$numerator` (pembilang) dan `$denominator` (penyebut). Tujuan fungsi ini adalah untuk menghitung hasil pembagian bulat dari `$numerator` oleh `$denominator` tanpa menggunakan operator pembagian.

2. Fungsi ini menginisialisasi variabel `$result` dengan nilai 0. Variabel ini akan digunakan untuk menyimpan hasil pembagian.

3. Selama `$numerator` lebih besar atau sama dengan `$denominator`, fungsi akan memasuki perulangan `while`. Di dalam perulangan, `$numerator` akan dikurangi sebanyak `$denominator`, dan nilai `$result` akan ditingkatkan. Ini akan terus berlanjut hingga `$numerator` lebih kecil dari `$denominator`.

4. Hasil akhir yang disimpan di dalam `$result` adalah hasil pembagian bulat antara `$numerator` dan `$denominator`.

5. Di bagian berikutnya, fungsi `divisionWithoutOperator` dipanggil dengan contoh penggunaan. Hasil pembagian antara 7 dan 2 disimpan dalam `$hasil1`, dan hasil pembagian antara 8 dan 4 disimpan dalam `$hasil2`.

6. Hasil dari kedua operasi pembagian ditampilkan menggunakan pernyataan `echo`. Hasil pembagian pertama (7 : 2) adalah 3, dan hasil pembagian kedua (8 : 4) adalah 2.

Program ini memberikan contoh bagaimana melakukan operasi pembagian bilangan bulat dengan cara yang sederhana tanpa menggunakan operator pembagian. Namun, metode ini tidak efisien untuk bilangan yang sangat besar, karena melibatkan pengurangan berulang kali.

**4. LOOPING CONDITION (****4\_looping\_condition.php)**

Buatlah sebuah program yang mencetak baris angka dari 1 sampai dengan 50, dengan ketentuan sebagai berikut:

● Jika angka yang akan tercetak merupakan kelipatan tiga, maka program akan mencetak kata “Foo”

● Jika angka yang akan tercetak merupakan kelipatan lima, maka program akan mencetak kata “Bar”

● Jika angka yang akan tercetak merupakan kelipatan tiga dan lima sekaligus, maka program akan mencetak kata “FooBar”.

Program :

<?php

$fooCount = 0;

$barCount = 0;

$fooBarCount = 0;

for ($i = 1; $i <= 50; $i++) {

    if ($i % 3 === 0 && $i % 5 === 0) {

        echo "FooBar ";

        $fooBarCount++;

    } elseif ($i % 3 === 0) {

        echo "Foo ";

        $fooCount++;

    } elseif ($i % 5 === 0) {

        echo "Bar ";

        $barCount++;

    } else {

        echo $i . " ";

    }

}

echo "</br>";

echo "\n";

echo "Jumlah Foo: " . $fooCount . "\n";

echo "Jumlah Bar: " . $barCount . "\n";

echo "Jumlah FooBar: " . $fooBarCount . "\n";

?>

Output :



Pembahasan :

Program 4\_looping\_condition.php adalah program PHP yang mencetak angka dari 1 hingga 50 sesuai dengan aturan berikut:

- Jika angka adalah kelipatan 3, maka cetak "Foo."

- Jika angka adalah kelipatan 5, maka cetak "Bar."

- Jika angka adalah kelipatan 3 dan 5 sekaligus, maka cetak "FooBar."

Berikut adalah pembahasan singkatnya:

1. Variabel `$fooCount`, `$barCount`, dan `$fooBarCount` digunakan untuk menghitung berapa kali kata "Foo," "Bar," dan "FooBar" muncul.

2. Perulangan `for` digunakan untuk mengiterasi angka dari 1 hingga 50.

3. Di dalam perulangan, ada beberapa kondisi `if` yang menentukan apa yang akan dicetak sesuai dengan aturan yang telah disebutkan.

4. Jika angka adalah kelipatan 3 dan 5 (angka yang habis dibagi 3 dan 5), maka program akan mencetak "FooBar" dan menambahkan 1 ke `$fooBarCount`.

5. Jika angka adalah kelipatan 3 (angka yang habis dibagi 3), maka program akan mencetak "Foo" dan menambahkan 1 ke `$fooCount`.

6. Jika angka adalah kelipatan 5 (angka yang habis dibagi 5), maka program akan mencetak "Bar" dan menambahkan 1 ke `$barCount`.

7. Jika angka tidak memenuhi salah satu kondisi di atas, program akan mencetak angka itu sendiri.

8. Setelah perulangan, program mencetak total jumlah "Foo," "Bar," dan "FooBar" yang telah dicetak.

Program ini mencetak output sesuai dengan aturan yang diberikan dan menghitung berapa kali setiap kata ("Foo," "Bar," "FooBar") muncul dalam rentang angka dari 1 hingga 50.