

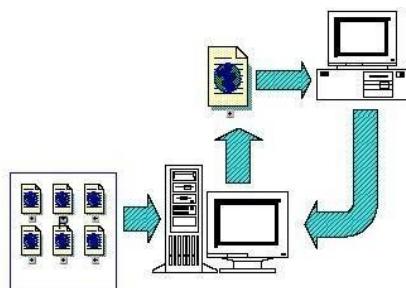
MODUL 9. PHP

Tujuan Praktikum

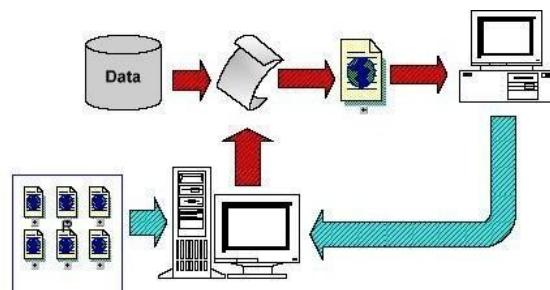
1. Mahasiswa mampu memahami konsep dan implementasi PHP pada web.
2. Mahasiswa mampu memahami sintaks, elemen, dan fungsi pada PHP.

9.1. Web Server dan Server Side Scripting

Web Server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima permintaan (*request*) berupa halaman web melalui HTTP atau HTTPS dari *client* yang dikenal dengan *web browser* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.



Gambar 9-1. Arsitektur web standar



Gambar 9-2. Arsitektur web dinamis

Beberapa *web server* yang banyak digunakan antara lain seperti berikut:

1. Apache Web Server (<https://httpd.apache.org/>)
2. Internet Information Service, IIS (<https://www.iis.net/>)
3. Xitami Web Server
4. Sun Java System Web Server

Server Side Scripting merupakan sebuah teknologi *scripting* atau pemrograman web dimana *script* (program) dikompilasi atau diterjemahkan di *server*. Dengan *server side scripting*, memungkinkan untuk menghasilkan halaman web yang dinamis.

Beberapa contoh *Server Side Scripting (Programming)* :

1. ASP (Active Server Page) dan ASP.NET
2. ColdFusion (<http://www.adobe.com/products/coldfusion-family.html>)
3. Java Server Pages (<http://www.oracle.com/technetwork/java/javaee/jsp/index.html>)

4. Perl (<https://www.perl.org/>)
5. Python (<https://www.python.org/>)
6. PHP (<http://www.php.net/>)

Keistimewaan PHP sebagai bahasa pemrograman berbasis *web* adalah :

1. Cepat
2. *Free*
3. Mudah dipelajari
4. *Multi-platform*
5. Dukungan *technical support*
6. Banyaknya komunitas PHP
7. Aman

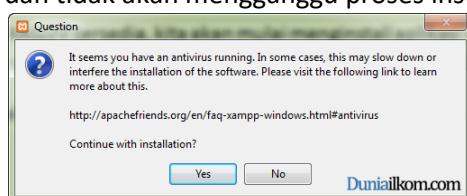
9.2. Instalasi Apache, PHP dan MySQL dengan XAMPP

Proses instalasi Apache, PHP dan MySQL terkadang menjadi kendala tersendiri bagi yang ingin memulai belajar pemrograman *web*. Namun saat ini sudah tersedia banyak aplikasi paket yang mencakup ketiga aplikasi tersebut. Beberapa diantaranya adalah sebagai berikut:

1. XAMPP (versi windows) dan LAMPP (versi linux) yang dapat diunduh di <https://www.apachefriends.org/download.html>
2. WAMP Server
3. APPServ
4. PHPTriad

Pada modul ini, akan digunakan aplikasi paket XAMPP sebagai sarana pendukung dan pembelajaran pemrograman *web*.

- a. Persiapan Instalasi
 1. Pastikan komputer anda tidak terinstall *web server* lain seperti IIS atau PWS karena dapat menyebabkan bentrok dengan *web server* Apache yang akan dipasang. Namun jika anda tetap ingin mempertahankannya, setelah installasi *web server* Apache selesai, anda dapat mengkonfigurasinya secara manual untuk mengganti nomor *port* yang akan digunakan oleh Apache.
 2. Download source XAMPP versi 5.6.32 (yang akan digunakan di modul ini) pada <https://www.apachefriends.org/download.html> dan tersedia untuk sistem operasi Windows, Linux dan Mac.
- b. Proses Instalasi
 1. Jalankan *file installer* XAMPP
 2. Akan ditampilkan jendela instalasi XAMPP. Pilih **Yes**. Peringatan tersebut menunjukkan bahwa antivirus sedang berjalan dan tidak akan mengganggu proses instalasi.



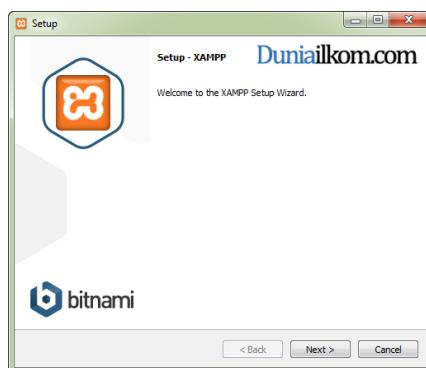
Gambar 9-3. Peringatan antivirus pada instalasi XAMPP

3. Setelah itu akan muncul jendela peringatan UAC (*User Account Control*). Klik **OK**.



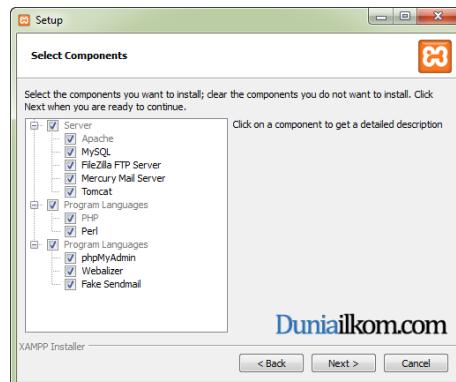
Gambar 9-4. Peringatan UAC pada instalasi XAMPP

4. Akan muncul jendela awal instalasi. Klik **Next** untuk melanjutkan instalasi.



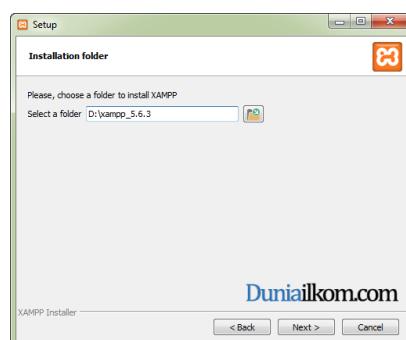
Gambar 9-5. Jendela awal instalasi XAMPP

5. Jendela berikutnya adalah “**Select Component**”. Pada bagian ini kita bisa memilih aplikasi apa saja yang akan dipasang. Setelah selesai dipilih, klik **Next** untuk melanjutkan.



Gambar 9-6. Jendela pemilihan komponen pada XAMPP

6. Selanjutnya adalah memilih lokasi dimana XAMPP akan dipasang. Jika sudah ditentukan lokasinya, klik next **Next**.



Gambar 9-7. Jendela pemilihan lokasi instalasi

7. Tampilan berikutnya adalah jendela “**Bitnami for XAMPP**”. Hilangkan centang pada bagian “Learn more about Bitnami for XAMPP”. Klik **Next**.

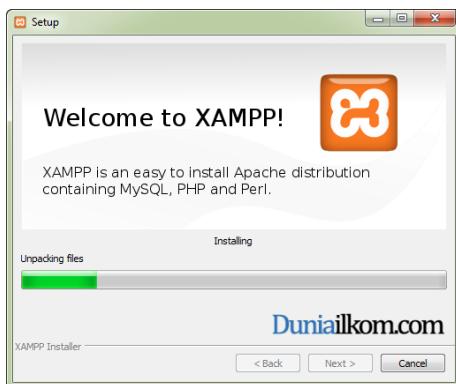


Gambar 9-8. Jendela instalasi Bitnami pada XAMPP

8. Jendela berikutnya adalah konfirmasi untuk mulai memasang XAMPP. Klik **Next** dan proses instalasi akan berlangsung.



Gambar 9-9. Jendela konfirmasi instalasi



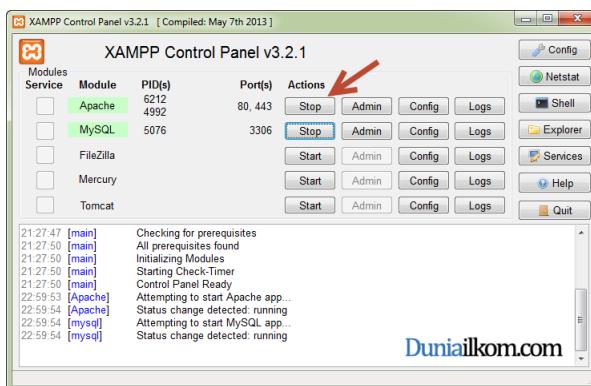
Gambar 9-10. Jendela proses instalasi

9. Jika jendela “**Completing the XAMPP Setup Wizard**” telah tampil, maka proses instalasi XAMPP telah selesai. Centang bagian “**Do you want to start the Control Panel now?**” untuk menjalankan XAMPP.



Gambar 9-11. Jendela instalasi XAMPP selesai

10. Akan muncul jendela Control Panel dari XAMPP. Pada bagian ini, kita bisa mengaktifkan *service* apa saja yang akan dijalankan dengan cara menekan tombol **Start** pada setiap *service* yang ada.



Gambar 9-12. Jendela ketika service pada XAMPP dijalankan

11. Untuk mengujinya, buka *web browser* dan ketikkan alamat **localhost** pada *address bar*, kemudian tekan **enter**. Akan tampil jendela XAMPP yang menunjukkan bahwa proses instalasi dan *service* Apache berjalan dengan baik.



Gambar 9-13. Tampilan ketika localhost dijalankan pada browser

9.3. Pengenalan PHP

Merupakan singkatan rekursif dari **PHP : Hypertext Preprocessor**. Pertama kali diciptakan oleh **Rasmus Lerdorf** pada tahun 1994. PHP sendiri harus ditulis diantara tag :

- <? dan ?>

- <?php dan ?>
- <script language="php"> dan </script>
- <% dan %>

Setiap satu *statement* (perintah) biasanya diakhiri dengan titik-koma (;). PHP juga **case sensitive** untuk nama *identifier* yang dibuat oleh *user* sedangkan *identifier* bawaan dari PHP **tidak case sensitive**. Contoh program yang ditulis dengan bahasa PHP

```
<?php
    echo "Hello World!";
?>
```

Simpan *file* tersebut dengan nama **hello.php** pada direktori **htdocs** yang ada di folder XAMPP. Kemudian, jalankan pada *browser* dengan mengetikkan alamat <http://localhost/hello.php>. Hasilnya akan muncul di *web browser* seperti berikut:



Gambar 9-14. Tampilan ketika menjalankan file PHP pada browser

Tambahan: Komentar dalam PHP

Komentar dalam PHP ada beberapa macam sesuai seperluannya, misalkan

```
<?php
// Ini adalah komentar satu baris

# Ini juga komentar satu baris (gaya Unix shell)

/*
    Ini adalah komentar
    lebih dari satu baris
    atau multi-line comment
*/

echo "Hello World"; // komentar bisa di akhir
statement
?>
```

9.4. Variabel

Variabel digunakan untuk menyimpan sebuah *value* (nilai), data atau informasi. Nama variabel pada PHP diawali dengan tanda \$. Panjang dari suatu variabel tidak terbatas dan variabel tidak perlu dideklarasikan terlebih dahulu sebelumnya. Setelah tanda \$, dapat diawali dengan huruf atau *under-score* (_). Karakter berikutnya bisa terdiri dari huruf, angka dan atau karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).

Variabel pada PHP bersifat *case sensitive* artinya besar kecilnya suatu karakter berpengaruh pada variabel tersebut. Suatu karakter pada PHP tidak boleh mengandung spasi.

Berikut adalah contoh penggunaan variabel pada PHP:

```

<?php
    $nim = "2211104042";
    $nama = "Cabine";

    echo "NIM : " . $nim;
    echo "Nama : " . $nama;

?>

```

Pada PHP, tipe data dari suatu variabel tidak didefinisikan langsung oleh *programmer*, akan tetapi secara otomatis akan ditentukan oleh interpreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu:

Tipe Data	Penjelasan Singkat	Contoh Kode PHP
Boolean	Bernilai true atau false, dipakai untuk logika	\$isLoggedIn = true;
Integer	Bilangan bulat tanpa desimal	\$umur = 20;
Float	Bilangan desimal	\$tinggi = 165.5;
String	Teks yang terdiri dari karakter	\$nama = "Dewi";
Array	Menyimpan banyak nilai dalam satu variabel	\$buah = ["Apel", "Jeruk"];
Object	Data yang berasal dari class (OOP)	class Mhs { public \$nama = "Dewi"; }
Resource	Referensi ke sumber eksternal seperti file atau DB	\$f = fopen("data.txt", "r");
NULL	Variabel tanpa nilai	\$x = NULL;

9.5. Konstanta

Konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta pada PHP, dapat menggunakan fungsi `define()` yang telah tersedia pada PHP. Berikut adalah contohnya :

```

<?php
    define("NAMA", "Baharuddin");
    define("NIM", "1301165454");
    echo "Nama : " . NAMA;
    echo "NIM : " . NIM;
?>

```

9.6. Operator dalam PHP

Ada beberapa jenis operator pada PHP, yaitu:

Tabel 8-1. Operator pada PHP

Jenis Operator	Operator	Contoh	Keterangan
Arimatika	+	\$a + \$b	Pertambahan
	-	\$a - \$b	Pengurangan
	*	\$a * \$b	Perkalian
	/	\$a / \$b	Pembagian
	%	\$a % \$b	Modulus, sisa hasil bagi
Penugasan	=	\$a = 4	Variabel \$a akan diisi oleh 4

Jenis Operator	Operator	Contoh	Keterangan
Bitwise	&	\$a & \$b	<i>Bitwise AND</i>
		\$a \$b	<i>Bitwise OR</i>
	^	\$a ^ \$b	<i>Bitwise XOR</i>
	~	~\$a	<i>Bitwise NOT</i>
	<<	\$a << \$b	<i>Shift Left</i>
	>>	\$a >> \$b	<i>Shift Right</i>
Perbandingan	==	\$a == \$b	Sama dengan
	===	\$a === \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan
Logika	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a \$b	TRUE jika \$a atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$b	TRUE jika \$b FALSE
String	.	\$a . \$b	Penggabungan string \$a dan \$b
Increment	++\$a	\$a = 5; echo ++\$a;	Pre-increment: nilai dinaikkan dulu (jadi 6), lalu ditampilkan
	\$a++	\$a = 5; echo \$a++;	Post-increment: nilai ditampilkan dulu (5), baru dinaikkan menjadi 6
Decrement	--\$a	\$a = 5; echo --\$a;	Pre-decrement: nilai dikurangi dulu (jadi 4), lalu ditampilkan
	\$a--	\$a = 5; echo \$a--;	Post-decrement: nilai ditampilkan dulu (5), lalu dikurangi menjadi 4

9.7. Struktur Kondisi

Struktur kondisi pada PHP sama halnya dengan bahasa pemrograman lainnya seperti Java. Berikut adalah contoh penulisan struktur kondisi **if-then** pada PHP:

```
if (kondisi) {
    statement-jika-kondisi-TRUE;
} else {
    Statement-jika-kondisi-FALSE; }
```

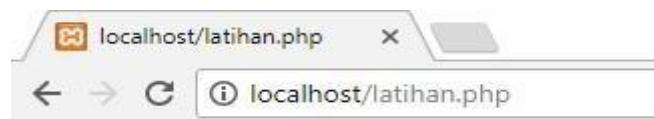
Selain struktur kondisi **if-then**, terdapat pula struktur kondisi **switch-case** seperti berikut:

```
switch ($var) {
    case '1' : statement-1; break;

    case '2' : statement-2; break;
    . . .
```

Berikut adalah contoh ketika *statement* kondisi **if-then** dijalankan :

```
$nilai = 80;
if ($nilai > 50) {
    echo "Nilai Anda adalah " . $nilai . ". Selamat, Anda lulus";
} else {
    echo "Nilai Anda adalah " . $nilai . ". Maaf, Anda tidak lulus";
}
```



Gambar 9-15. Hasil penggunaan IF-THEN

Dan berikut ini adalah contoh ketika *statement* **switch-case** dijalankan:

```

$nilai = 80;

switch ($nilai) {
    case ($nilai > 50 && $nilai <= 60):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda C";
        break;

    case ($nilai > 60 && $nilai <= 70):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda BC";
        break;

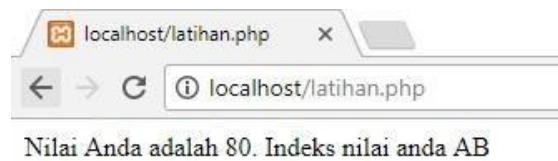
    case ($nilai > 70 && $nilai <= 75):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda B";
        break;

    case ($nilai > 75 && $nilai <= 80):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda AB";
        break;

    case ($nilai > 80 && $nilai <= 100):
        echo "Nilai Anda adalah $nilai. Indeks nilai anda A";
        break;

    default:
        echo "Nilai Anda adalah $nilai. Maaf, Anda tidak lulus";
        break;
}

```



Gambar 9-16. Hasil penggunaan **switch-case**

9.8. Perulangan (Looping)

Banyak jenis perulangan yang terdapat pada PHP. Adapun beberapa diantaranya adalah :

1. Perulangan for

```

for (init_awal, kondisi, counter) {
    statement;
}

```

2. Perulangan while

```

init_awal; while (kondisi) {
    statement;
    counter;
}

```

3. Perulangan do-while

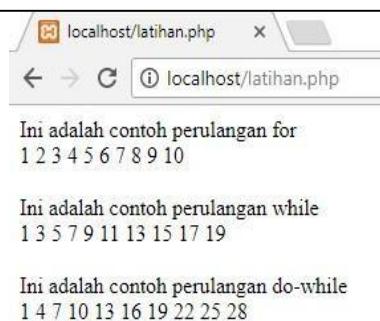
```
init_awal;  
do {  
    statement;    counter;  
} while (kondisi);
```

4. Perulangan foreach

```
foreach (array_expression as $value) {  
    statement;  
}
```

Berikut adalah contoh penggunaan perulangan :

```
<?php  
echo "Ini adalah contoh perulangan for";  
echo "<br>";  
for ($i = 1; $i <= 10; $i++) {  
    echo $i . " ";  
}  
  
echo "<br>";  
echo "<br>";  
echo "Ini adalah contoh perulangan while";  
echo "<br>";  
$i = 1;  
while ($i <= 20) {  
    echo $i . " ";  
    $i += 2;  
}  
  
echo "<br>";  
echo "<br>";  
echo "Ini adalah contoh perulangan do-while";  
echo "<br>";  
$i = 1;  
do {  
    echo $i . " ";  
    $i += 3;  
} while ($i < 30);  
?>
```



Gambar 9-17. Hasil penggunaan perulangan

9.9. Function

Dalam merancang kode program, kadang kita sering membuat kode yang melakukan tugas yang sama secara berulang-ulang, seperti membaca tabel dari *database*, menampilkan penjumlahan, dan lainlain. Tugas yang sama ini akan lebih efektif jika dipisahkan dari program utama, dan dirancang menjadi sebuah **fungsi**.

Fungsi dipanggil dengan menulis nama dari fungsi tersebut, dan diikuti dengan argumen (jika ada). Argumen ditulis di dalam tanda kurung, dan jika jumlah argumen lebih dari satu, maka diantaranya dipisahkan oleh karakter koma.

Bentuk umum pendefinisian fungsi pada PHP adalah sebagai berikut:

```
function nama_fungsi(parameter1, parameter2,  
... , n) {  
    statement;  
}
```

Contoh fungsi pada PHP tanpa menggunakan parameter dan *return value*:

```
<?php  
function cetakGenap()  
{  
    for ($i = 1; $i <= 100; $i++) {  
        if ($i % 2 == 0) {  
            echo "$i ";  
        }  
    }  
}  
  
//pemanggilan fungsi  
cetakGenap();  
?>
```

Contoh fungsi pada PHP menggunakan parameter dan tanpa *return value*:

```
<?php  
function cetakGenap($awal, $akhir)  
{  
    for ($i = $awal; $i <= $akhir; $i++) {  
        if ($i % 2 == 0) {  
            echo "$i ";  
        }  
    }  
}  
  
//pemanggilan fungsi  
$a = 10;  
$b = 50;  
echo "Bilangan ganjil dari $a sampai $b adalah : <br>";  
cetakGenap($a, $b);  
?>
```

Contoh fungsi pada PHP dengan *return value*:

```
<?php function luasSegitiga($alas, $tinggi)
{
    return 0.5 * $alas * $tinggi;
}

//pemanggilan fungsi
$a = 10;
$t = 50;
echo "Luas Segitiga dengan alas $a dan tinggi $t adalah : " . luasSegitiga($a,
$t);
?>
```

9.10. Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun *array* disebut elemen *array*, yang masing-masing elemen dapat diakses tersendiri melalui *index array*. *Index array* dapat berupa bilangan *integer* atau *string*.

Untuk mendeklarasikan atau mendefinisikan sebuah *array* di PHP bisa menggunakan keyword `array()`. Jumlah elemen *array* tidak perlu disebutkan saat deklarasi. Sedangkan untuk menampilkan isi *array* pada elemen tertentu, cukup dengan menyebutkan nama *array* beserta *index array*-nya.

Berikut adalah cara mendeklarasikan suatu *array* di PHP :

```
<?php
$arrKendaraan = ["Mobil", "Pesawat", "Kereta Api", "Kapal Laut"];
echo $arrKendaraan[0] . "<br>"; //Mobil
echo $arrKendaraan[2] . "<br>"; //Kereta Api

$arrKota = [];
$arrKota[] = "Jakarta";
$arrKota[] = "Medan";
$arrKota[] = "Bandung";
$arrKota[] = "Malang";
$arrKota[] = "Sulawesi";

echo $arrKota[1] . "<br>"; //Medan
echo $arrKota[2] . "<br>"; //Bandung
echo $arrKota[4] . "<br>"; //Sulawesi
?>
```

Cara mendeklarasikan suatu *array* pada PHP bisa dengan *index string* atau yang dinamakan dengan *array assosiatif*. Berikut adalah contoh pendeklarasian *array assosiatif*:

```
<?php
$arrAlamat = [
    "Rona" => "Banjarmasin",
    "Dhiva" => "Bandung",
    "Ilham" => "Medan",
    "Oku" => "Hongkong",
];

echo $arrAlamat["Dhiva"] . "<br>"; //Bandung
echo $arrAlamat['Oku'] . "<br>"; //Hongkong

$arrNim = [];
$arrNim["Rona"] = "11011112";
$arrNim["Dhiva"] = "11011101";
$arrNim["Ilham"] = "11011309";
$arrNim["Oku"] = "11014765";
$arrNim["Fadhlaln"] = "11011113";

echo $arrNim["Ilham"] . "<br>"; //11011309
echo $arrNim['Fadhlaln'] . "<br>"; //11011113
?>
```

9.11. Tugas (UNGUIDED)

1. Buat program konversi suhu yang dapat mengkonversi:

- Celcius ke Fahrenheit
- Fahrenheit ke Celcius
- Celcius ke Kelvin

Tampilkan menggunakan 2 desimal di belakang koma

2. Kalkulator Diskon:

Input: Total belanja

- Diskon 10% jika belanja \geq Rp 100.000
- Diskon 20% jika belanja \geq Rp 500.000
- Diskon 30% jika belanja \geq Rp 1.000.000

Tampilkan: total belanja, diskon, dan total bayar

3. Manipulasi Array:

Buat array nilai mahasiswa: [75, 89, 65, 90, 85, 70, 98, 65, 69, 70, 12]

Tampilkan:

- Nilai tertinggi
- Nilai terendah
- Rata-rata nilai
- Jumlah mahasiswa yang lulus (≥ 70)

Urutkan nilai dari tertinggi ke terendah