

LAPORAN PRAKTIKUM
PERANCANGAN DAN PEMROGRAMAN WEB

MODUL IX
(NAMA MODUL)



Oleh:

(TSAQIF HISYAM SAPUTRA)

(2311104024)

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
DIREKTORAT KAMPUS PURWOKERTO
UNIVERSITAS TELKOM
2025

BAB I

PENDAHULUAN

1.1 Dasar Teori

PHP (Hypertext Preprocessor)

PHP adalah bahasa pemrograman server-side yang banyak digunakan untuk pengembangan web. PHP memiliki berbagai fitur yang memudahkan pengolahan data, termasuk:

Operator Aritmatika:

- Digunakan untuk melakukan operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian.
- Contoh: ``+``, ``-``, ``*``, ``/``, ``%``

Struktur Kontrol (Conditional Statement):

- ``if``, ``elseif``, ``else`` digunakan untuk membuat keputusan berdasarkan kondisi tertentu.
- Operator perbandingan: ``>`` (lebih besar), ``>=`` (lebih besar sama dengan), ``<`` (lebih kecil), ``<=`` (lebih kecil sama dengan), ``==`` (sama dengan).

Array dan Fungsi Bawaan PHP:

- Array adalah struktur data yang dapat menyimpan banyak nilai dalam satu variabel.
- Fungsi bawaan PHP untuk manipulasi array:
 - o ``max()`` - mencari nilai maksimum
 - o ``min()`` - mencari nilai minimum
 - o ``array_sum()`` - menjumlahkan semua elemen array
 - o ``count()`` - menghitung jumlah elemen array
 - o ``rsort()`` - mengurutkan array secara descending
 - o ``array_filter()`` - memfilter elemen array berdasarkan kondisi

Fungsi Format Angka:

- `number_format($angka, $desimal)` - memformat angka dengan jumlah desimal tertentu
- `round($angka, $presisi)` - membulatkan angka ke presisi tertentu

Rumus Konversi Suhu:

1. Celcius ke Fahrenheit: $F = C \times 9/5 + 32$
2. Fahrenheit ke Celcius: $C = (F - 32) \times 5/9$
3. Celcius ke Kelvin: $K = C + 273.15$

1.2 Tujuan

1. Memahami penggunaan operator aritmatika dalam PHP untuk melakukan konversi suhu.
2. Memahami penggunaan struktur kontrol `if-elseif-else` untuk membuat logika diskon bertingkat.
3. Memahami penggunaan array dan fungsi bawaan PHP untuk manipulasi data.
4. Mampu mengimplementasikan fungsi `number_format()` untuk memformat output angka.
5. Mampu menggunakan fungsi array seperti `max()`, `min()`, `array_sum()`, `count()`, dan `rsort()`.

1.3 Manfaat

1. Manfaat Akademis:

- Meningkatkan pemahaman tentang sintaks dan logika pemrograman PHP.
- Melatih kemampuan problem-solving dalam pemrograman.
- Memahami best practice dalam penulisan kode yang bersih dan terstruktur.

2. Manfaat Praktis:

- Mampu membuat aplikasi konversi suhu yang dapat digunakan dalam kehidupan sehari-hari.
- Mampu membuat sistem perhitungan diskon yang umum digunakan dalam aplikasi e-commerce.
- Mampu mengolah dan menganalisis data menggunakan array PHP.

3. Manfaat Pengembangan Skill:

- Meningkatkan kemampuan coding dengan PHP.
- Memahami penggunaan fungsi bawaan PHP yang efisien.
- Melatih kemampuan debugging dan testing kode.

BAB II

HASIL PRAKTIKUM

2.1 Kasus 1: Program Konversi Suhu

Pada percobaan pertama, praktikum difokuskan pada penerapan operator aritmatika dalam PHP untuk melakukan konversi satuan suhu. Program diimplementasikan dengan mendeklarasikan variabel awal sebesar 30°C dan 86°F. Logika konversi disusun menggunakan rumus standar fisika, yaitu Celcius ke Fahrenheit menggunakan rumus $(C * 9/5) + 32$, Fahrenheit ke Celcius menggunakan $(F - 32) * 5/9$, dan Celcius ke Kelvin dengan menambahkan konstanta 273.15.

Berdasarkan hasil eksekusi program, konversi suhu berjalan dengan akurat. Input suhu 30°C berhasil dikonversi menjadi 86.00°F dan 303.15 K. Sebaliknya, input 86°F berhasil dikembalikan menjadi nilai 30.00°C. Hal yang perlu digarisbawahi pada implementasi ini adalah penggunaan fungsi bawaan PHP `number_format($variabel, 2)`. Fungsi ini memastikan bahwa setiap hasil perhitungan ditampilkan dengan format dua digit desimal di belakang koma, sehingga output menjadi lebih rapi dan memenuhi standar presisi yang diminta dalam modul tugas.

PRAKTIKUM PHP - 3 STUDI KASUS

KASUS 1: Program Konversi Suhu

Hasil Konversi Suhu:

Konversi	Nilai Awal	Hasil
Celcius ke Fahrenheit	30°C	86.00°F
Fahrenheit ke Celcius	86°F	30.00°C
Celcius ke Kelvin	30°C	303.15 K

2.2 Kasus 2: Kalkulator Diskon Bertingkat

Studi kasus kedua menerapkan struktur kontrol percabangan `if-elseif-else` untuk membuat kalkulator diskon otomatis berdasarkan total belanja. Logika program dibungkus dalam fungsi `hitungDiskon()` yang memproses input angka dan mengembalikan array berisi detail pembayaran. Penentuan diskon dilakukan secara bertingkat dari persentase terbesar untuk menghindari kesalahan logika. Ketentuan yang diterapkan adalah diskon 30% untuk belanja minimal Rp 1.000.000, diskon 20% untuk belanja minimal Rp 500.000, dan diskon 10% khusus untuk belanja di atas Rp 100.000 (menggunakan operator `>`).

Pengujian dilakukan dengan beberapa skenario nilai belanja, yaitu Rp 80.000, Rp 150.000, Rp 500.000, dan Rp 1.200.000. Hasil output menunjukkan bahwa logika percabangan berfungsi dengan benar. Pada belanja Rp 80.000, program tidak memberikan diskon karena tidak memenuhi syarat minimum. Pada belanja Rp 150.000, diskon 10% aktif karena nilai belanja lebih besar dari Rp 100.000. Sementara itu, belanja Rp 500.000 dan Rp 1.200.000 masing-masing mendapatkan potongan 20% dan 30% karena memenuhi syarat penggunaan operator `>=` (lebih besar sama dengan). Perhitungan akhir `totalBayar` juga terbukti valid setelah dikurangi nominal diskon yang sesuai.

KASUS 2: Kalkulator Diskon Bertingkat

Hasil Perhitungan Diskon:

Total Belanja	Diskon (%)	Nominal Diskon	Total Bayar Bersih
Rp 80.000	0%	Rp 0	Rp 80.000
Rp 150.000	10%	Rp 15.000	Rp 135.000
Rp 500.000	20%	Rp 100.000	Rp 400.000
Rp 1.200.000	30%	Rp 360.000	Rp 840.000

Ketentuan Diskon:

- Belanja **> Rp 100.000** → Diskon 10%
- Belanja **>= Rp 500.000** → Diskon 20%
- Belanja **>= Rp 1.000.000** → Diskon 30%

2.3 Kasus 3: Manipulasi Array Nilai

Percobaan ketiga berfokus pada pengelolaan struktur data array menggunakan data nilai mahasiswa: [75, 89, 65, 90, 85, 70, 98, 65, 69, 70, 12]. Implementasi ini bertujuan untuk menguji efisiensi penggunaan fungsi bawaan (built-in functions) PHP dalam pengolahan data statistik sederhana tanpa perlu melakukan perulangan manual yang kompleks. Fungsi `max()` dan `min()` digunakan untuk mencari nilai ekstrim, sedangkan kombinasi `array_sum()` dan `count()` digunakan untuk menghitung nilai rata-rata kelas.

Hasil eksekusi program menunjukkan data statistik yang akurat, di mana nilai tertinggi yang ditemukan adalah 98 dan nilai terendah adalah 12, dengan rata-rata kelas sebesar 71.64. Selain statistik dasar, program juga melakukan penyaringan data untuk menghitung jumlah kelulusan dengan standar nilai minimal 70. Menggunakan perulangan dengan kondisi `if ($nilai >= 70)`, program berhasil mengidentifikasi bahwa terdapat 7 nilai yang memenuhi syarat lulus (termasuk nilai batas 70). Terakhir, fungsi `rsort()` berhasil mengurutkan array secara *descending* (menurun), sehingga data tersaji rapi dari nilai terbesar (98) hingga terkecil (12).

KASUS 3: Manipulasi Array Nilai Mahasiswa

Data Array Nilai:

[75, 89, 65, 90, 85, 70, 98, 65, 69, 70, 12]

Hasil Manipulasi Array:

Operasi	Hasil
Nilai Tertinggi (max)	98
Nilai Terendah (min)	12
Rata-rata Nilai	71.64
Jumlah Mahasiswa Lulus (nilai >= 70)	7 mahasiswa
Urutan Nilai (Tertinggi ke Terendah)	[98, 90, 89, 85, 75, 70, 70, 69, 65, 65, 12]

Detail Mahasiswa Lulus (Nilai >= 70):

Nilai yang lulus: [75, 89, 90, 85, 70, 98, 70]

BAB III

KESIMPULAN & SARAN

3.1 Kesimpulan

Dapat disimpulkan bahwa penerapan logika dasar PHP mampu menyelesaikan permasalahan komputasi sederhana dengan efektif dan akurat. Program konversi suhu berhasil menerapkan format presisi desimal menggunakan fungsi `number_format()`, sedangkan kalkulator diskon menunjukkan ketepatan logika percabangan `if-elseif` dalam menangani batasan nilai belanja menggunakan operator perbandingan `>` dan `>=` yang spesifik. Selain itu, pemanfaatan fungsi bawaan (*built-in functions*) PHP dalam manipulasi array terbukti sangat efisien untuk mengolah data statistik nilai mahasiswa, mulai dari pencarian nilai ekstrim hingga pengurutan data, sehingga struktur kode program menjadi lebih ringkas dan mudah dipahami.

3.2 Saran

Untuk pengembangan program di masa mendatang, disarankan agar kode tidak lagi bergantung pada nilai variabel statis (*hardcoded*), melainkan diintegrasikan dengan antarmuka pengguna berbasis form HTML yang memungkinkan input data secara dinamis dan interaktif. Selain itu, penambahan fitur validasi input sangat penting untuk mencegah *error* akibat kesalahan tipe data yang dimasukkan pengguna, serta perluasan fungsionalitas seperti penambahan opsi satuan suhu lainnya atau visualisasi grafik pada data array akan membuat aplikasi ini lebih informatif dan meningkatkan pengalaman pengguna (*User Experience*).