

1. Konvensi Penulisan Kode Program PBL IF

Konvensi Penulisan Kode / Coding convention adalah aturan atau panduan yang digunakan untuk menulis kode pemrograman dengan cara yang konsisten dan mudah dibaca. Tujuannya adalah agar kode yang ditulis oleh berbagai programmer dalam satu tim terlihat seragam, lebih mudah dipahami, dan lebih gampang dikelola atau diperbaiki. Misalnya, coding convention bisa mengatur cara memberi nama variabel, fungsi, cara penulisan komentar, atau cara mengatur indentasi dalam kode. Dengan mengikuti coding convention, kerja sama dalam tim jadi lebih lancar, dan kemungkinan kesalahan dalam kode bisa dikurangi.

Konvensi penulisan ini berlaku untuk semua program yang dibuat oleh mahasiswa pada tugas kuliah, praktikum maupun PBL. File program yang disubmit (baik praktikum maupun PBL) tanpa mengikuti konvensi ini dapat dianggap tidak mensubmit tugas atau dikurangi nilainya.

1. Komentar Pada Kode

- Komentar pada Setiap File: Setiap file program harus diawali dengan komentar yang menjelaskan fungsi dari file tersebut, pembuat dan waktu pembuatan.
- Format komentar di awal file:

```
php Copy code

// Nama File: [nama_file.php]
// Deskripsi: [Penjelasan singkat tentang fungsi atau tujuan dari file ini]
// Dibuat oleh: [Nama Pembuat] - NIM: [NIM Mahasiswa]
// Tanggal: [Tanggal Pembuatan]
```

```
php Copy code

// Nama File: user_manager.php
// Deskripsi: File ini mengelola semua fungsi yang berhubungan dengan pengguna seperti log
// Dibuat oleh: John Doe - NIM: 12345678
// Tanggal: 09 Agustus 2024
```

- Berikan komentar pada kode yang sulit dipahami atau memiliki logika yang kompleks.
- Gunakan komentar satu baris (//) untuk menjelaskan apa yang dilakukan setiap bagian kode.
- Berikan komentar blok (/* ... */) untuk mendokumentasikan fungsi atau metode yang penting.

Contoh:

```
<?php
// Fungsi untuk menambahkan pengguna baru ke dalam database
function addUser($name, $email, $password) {
    // Enkripsi password sebelum disimpan
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);

    // Query untuk menyimpan data pengguna
    $sql = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";
```

2. Penamaan:

- Variabel dan Fungsi: Gunakan camelCase (contoh: `userName`, `getData()`).
- Kelas: Gunakan PascalCase (contoh: `UserManager`).
- Konstanta: Gunakan huruf besar dengan underscore (`UPPER_SNAKE_CASE`) (contoh: `MAX_USER_LIMIT`).

3. HTML

- Indentasi: Gunakan indentasi 2 atau 4 spasi untuk elemen yang bersarang.
- Penamaan Kelas: Gunakan nama kelas CSS yang jelas dan deskriptif (contoh: `.menu-bar`, `.button-primary`).
- Penulisan Tag: Semua tag harus ditutup, baik yang memerlukan penutupan (`<div></div>`) maupun yang tidak (``).

4. CSS

- Selectors: Gunakan selector yang jelas dan sederhana (contoh: `.header`, `.footer-link`).
- Pengelompokan Properti: Urutkan properti CSS secara logis:
 - Box model: `margin`, `padding`, `width`, `height`.
 - Tipografi: `font-size`, `color`, `text-align`.
 - Visual: `background`, `border`.
- Satuan: Gunakan rem atau em untuk ukuran font dan spasi. Hindari penggunaan px kecuali sangat diperlukan.

5. JavaScript

- Indentasi dan Spasi:
 - Gunakan indentasi 2 atau 4 spasi.
 - Tambahkan spasi sebelum dan sesudah operator (`=`, `+`, `-`).
- Deklarasi Variabel: Gunakan `let` untuk variabel yang dapat berubah, dan `const` untuk variabel yang tidak berubah.
- Penamaan Fungsi: Gunakan nama yang deskriptif untuk fungsi (contoh: `calculateTotal()`, `fetchData()`).
- Error Handling: Gunakan `try...catch` untuk menangani error pada operasi penting.

6. RDBMS (MySQL, Postgre, SQLServer atau lainnya)

- Penamaan Tabel dan Kolom: Gunakan snake_case untuk nama tabel dan kolom (contoh: `user_data`, `order_items`).
- Primary Keys: Setiap tabel harus memiliki primary key. Contoh: (`id`)
- Struktur Database: Pastikan relasi yang dibuat benar, tidak redundan