KOMDIGI
Kementerian Komunikasi dan Digital
Republik Indonesia

digitalent

VSGA Vocational School Graduate Academy

# PERTEMUAN 9

## PELATIHAN JUNIOR WEB DEVELOPER

**Menyusun Fungsi, File atau Sumber Daya Pemrograman yang Lain dalam Organisasi yang Rapi**

(Saluran Daring, [Tanggal Penyampaian])

# Menyusun fungsi, File atau Sumber Daya Pemrograman yang Lain dalam Organisasi yang Rapi

**Deskripsi Singkat mengenai Topik:**
Pelatihan ini berhubungan dengan pengorganisasian sumber daya hasil pemrograman seperti source code, file referensi, dokumentasi, dan lainnya dalam organisasi yang rapi.

**Tujuan Pelatihan:**
Peserta pelatihan memiliki sikap, pengetahuan dan keterampilan yang dibutuhkan dalam mengorganisasikan sumber daya hasil pemrograman seperti source code, file referensi, dokumentasi, dan lainnya dalam organisasi yang rapi sehingga memudahkan diri sendiri maupun orang lain dalam mengakses dan menggunakan sumber daya tersebut.

**Materi yang akan disampaikan:**
1. Pemilihan library
2. Integrasi Library
3. Pembaharuan Library

**Tugas:**
Memasang library dan menggunakan dalam program

**Outcome/Capaian Pelatihan:**
1. Peserta dapat melakukan pemilihan unit-unit reuse yang potensial
2. Peserta dapat melakukan integrasi library atau komponen pre-existing dengan source code yang ada
3. Peserta dapat melakukan pembaharuan library atau komponen pre-existing yang digunakan

# Unit Kompetensi

| | | |
|---|---|---|
| **KODE UNIT** | : | **J.620100.015.01** |
| **JUDUL UNIT** | : | **Menyusun Fungsi, *File* atau Sumber Daya Pemrograman yang Lain dalam Organisasi yang Rapi** |
| **DESKRIPSI UNIT** | : | Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan dalam mengorganisasikan sumber daya hasil pemrogramannya seperti *source code*, *file* referensi, dokumentasi, dll dalam organisasi yang rapi sehingga memudahkan diri sendiri maupun orang lain dalam mengakses dan menggunakan sumber daya tersebut. |

# Elemen Kompetensi & Kriteria Unjuk Kerja

| ELEMEN KOMPETENSI | KRITERIA UNJUK KERJA |
|---|---|
| 1. Mengelola sumber daya pemrograman sesuai karakter | 1.1 Nama *file*, fungsi, variabel, konstanta, dan sumber daya pemrograman lain dibuat sesuai konteks.<br>1.2 Setiap fungsi/prosedur/program dilengkapi dengan penulisan komentar di awal mengenai deskripsi fungsi/ prosedur/program tersebut; *initial state* dan *final state*; *author* (pembuat); versi dan/atau tanggal.<br>1.3 Badan *source code* dilengkapi dengan komentar/keterangan yang cukup, yang memberikan penjelasan atas baris-baris intruksi. |
| 2. Mengorganisasikan sumber daya pemrograman sesuai konteks | 2.1 *Folder* dan sub–sub *folder* disusun sesuai konteks dan isinya<br>2.2 *File* "*readme*" dibuat, mengandung penjelasan mengenai struktur/hirarki *folder* serta penjelasan mengenai sumber daya pemrograman. |

# Konteks Variabel

- Yang dimaksud dengan **standar pemrograman** antara lain mengenai penulisan nama file, nama fungsi, nama variabel, nama konstanta, atau penulisan komentar.
- Yang dimaksud dengan **sumber daya pemrograman** adalah berbagai file yang digunakan untuk kegiatan pemrogaman seperti: file source code, file referensi, panduan penggunaan source code, panduan modifikasi source code, atau file basis data. Deskripsi struktur data utama yang dipilih; daftar class yang dipakai.

# Mengorganisasikan Folder dan Sub-Folder

Mengorganisasikan Sumber Daya Pemrograman

# Mengapa perlu Mengorganisasikan Project?

- PHP memberi Anda banyak kebebasan, hampir tidak ada aturan.
- Jika tidak melakukan organisasi atau pengaturan pada project, baik struktur folder, sub-folder, atau aturan penamaan, pada batasan tertentu kita pasti akan mendapatkan project dengan kode program yang “berantakan” dan sulit “dibaca” (dikenal dengan istilah: spaghetti code).
- Hal ini sering terjadi pada proyek PHP “cepat” yang kemudian berkembang secara organik.

# Membuat File Read Me

Mengorganisasikan Sumber Daya Pemrograman

# Deskripsi

- File README adalah sebuah berkas yang berisi informasi tentang berkas lain di dalam direktori atau arsip perangkat lunak komputer. Berkas ini berbentuk dokumentasi, biasanya berkas teks sederhana yang bernama **READ.ME**, **README.TXT**, **README.md** (untuk berkas teks yang menggunakan markah markdown), **README.1ST** – atau hanya **README**.
- Nama berkas pada umumnya ditulis dalam huruf besar. Pada sistem operasi mirip Unix, hal ini membuat berkas README mudah diperhatikan.

*Sumber: id.wikipedia.org*

# Jenis

- Petunjuk konfigurasi.
- Petunjuk instalasi.
- Petunjuk pengoperasian.
- Sebuah berkas manifes.
- Informasi hak cipta dan perizinan.
- Informasi kontak untuk distributor atau pemrogram.
- Bug yang diketahui.
- Troubleshooting.
- Kredit dan ucapan terima kasih.
- Sebuah catatan perubahan/changelog (biasanya untuk pemrogram).
- Bagian berita (biasanya untuk pengguna).

*Sumber: id.wikipedia.org*

# Mengapa perlu Membuat File Read Me?

- File README merupakan alat yang sangat berharga bagi pengembang dan pengguna proyek PHP.
- File README membantu menciptakan lingkungan yang transparan, memudahkan kolaborasi, dan memberikan dukungan yang diperlukan untuk memahami dan menggunakan proyek dengan efektif.
- Dengan membaca README, pengguna dan pengembang dapat lebih mudah memahami tujuan, fungsionalitas, dan cara penggunaan proyek PHP.

- Lebih detil, dapat dilihat pada tabel di slide berikut:

# Mengapa perlu Membuat File Read Me?

| Manfaat README dalam Proyek PHP | Bagi Programmer | Bagi Pengguna |
|---|---|---|
| **Dokumentasi yang Jelas** | Memberikan informasi tentang struktur dan fungsi proyek tanpa melihat kode. | Memudahkan pengguna memahami cara menggunakan proyek. |
| **Panduan Instalasi yang Lengkap** | Membantu pengembang dan kontributor mengonfigurasi dan menjalankan proyek. | Memberikan langkah-langkah jelas dalam menginstal proyek. |
| **Kontribusi dan Kolaborasi** | Menyediakan informasi cara berkontribusi dan aturan pengembangan. | Mendorong pengguna untuk berpartisipasi dalam pengembangan proyek. |
| **Pemahaman Struktur dan Organisasi** | Memberikan pandangan umum tentang struktur direktori dan komponen utama. | Memudahkan pengguna menavigasi dan memahami organisasi proyek. |
| **Lisensi dan Hak Cipta** | Memberikan kejelasan hukum tentang cara proyek dapat digunakan. | Memberikan pemahaman tentang batasan dan hak pengguna. |
| **Informasi Kontak dan Dukungan** | Memudahkan kontributor atau pengguna untuk menghubungi pengembang. | Menyediakan rute komunikasi untuk pertanyaan atau pelaporan masalah. |
| **Histori Perubahan dan Versi** | Menyajikan riwayat perubahan, pembaruan, dan pemecahan masalah. | Memungkinkan pengguna memahami perubahan dari versi ke versi. |
| **Panduan Penggunaan dan Contoh** | Memberikan contoh penggunaan dan pengimplementasian kode. | Mempermudah pengguna memahami cara menggunakan fitur tertentu. |
| **Penjelasan Ketergantungan dan Konfigurasi** | Menyajikan daftar ketergantungan dan konfigurasi yang diperlukan. | Memudahkan pengguna menyiapkan lingkungan yang dibutuhkan. |

# Panduan Penulisan yang Baik

- Jelas dan Singkat
  - Informasi disampaikan dengan jelas dan tidak berlebihan.
- Terstruktur
  - Dibagi menjadi bagian-bagian yang terstruktur untuk kemudahan navigasi.
- Relevan
  - Memuat informasi yang relevan dan diperlukan.
- Dilengkapi dengan Contoh
  - Memberikan contoh penggunaan atau tangkapan layar jika diperlukan.

# Summary

- File, fungsi, prosedur, variabel, konstanta, dan sumberdaya yang kita buat, harus diberi nama.
- Memberikan nama yang baik akan membantu membuat kode lebih mudah dimengerti dan dipelihara.
- Praktik ini membantu dalam meningkatkan kolaborasi dan kejelasan kode dalam pengembangan perangkat lunak.
- Memberikan komentar yang baik dan informatif dalam kode program sangat penting untuk mempermudah pemahaman dan pemeliharaan kode.
- Komentar membantu dalam mengkomunikasikan ide dan tujuan, sehingga membuat kode lebih mudah dipahami oleh diri sendiri dan orang lain yang membaca kode tersebut.

# Referensi


- Clean Code: A Handbook of Agile Software Craftsmanship by Robert C. Martin
- PHP Best Practices - Comments: https://phpbestpractices.org/#comments
- PHP FIG - PSR-1: Basic Coding Standard: https://www.php-fig.org/psr/psr-1
- PHP FIG - PSR-12: Extended Coding Style: https://www.php-fig.org/psr/psr-12
- PHP Naming Conventions: https://www.php-fig.org/bylaws/psr-naming-conventions/
- PHPDoc - Manual: https://docs.phpdoc.org
- https://id.wikipedia.org/
- https://kotakode.com/

KOMDIGI
Kementerian Komunikasi dan Digital
Republik Indonesia

digitalent

# Terima Kasih

