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

Nomor Dokumen SKKNI



MENTERI KETENAGAKERJAAN
REPUBLIK INDONESIA

KEPUTUSAN MENTERI KETENAGAKERJAAN
REPUBLIK INDONESIA
NOMOR 282 TAHUN 2016
TENTANG
PENETAPAN STANDAR KOMPETENSI KERJA NASIONAL INDONESIA
KATEGORI INFORMASI DAN KOMUNIKASI GOLONGAN POKOK AKTIVITAS
PEMROGRAMAN, KONSULTASI KOMPUTER DAN KEGIATAN YANG
BERHUBUNGAN DENGAN ITU (YBDI) BIDANG *SOFTWARE DEVELOPMENT*
SUBBIDANG PEMROGRAMAN

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. 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. 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 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.

Materi

Memberi Nama

Mengelola Sumber Daya Pemrograman

Pendahuluan

- 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.
- Nama yang baik memberikan petunjuk tentang tujuan dan fungsionalitas dari suatu variable, konstanta, fungsi atau prosedur.
- Dalam kode program, pemberian nama yang baik membantu menjadikan kode lebih mudah dibaca, dimengerti, dan dikelola.

Pemberian Nama yang Baik

- Deskriptif:
  - Nama harus mencerminkan tugas atau tujuan utama fungsi atau prosedur.
- Singkat dan Jelas:
  - Pilih nama yang singkat namun jelas menggambarkan aksi atau hasil yang diharapkan.
- Pakai Kata Kerja:
  - Gunakan kata kerja yang memberikan indikasi bahwa fungsi melakukan suatu tindakan.
- Hindari Singkatan yang Membingungkan:
  - Jika perlu, gunakan singkatan yang umum dan dapat dimengerti oleh programmer lain.
- Ikuti Konvensi Penamaan:
  - Gunakan konvensi penamaan yang konsisten, seperti camelCase atau snake_case, sesuai dengan standar komunitas PHP.
- Konsisten dalam Gaya Penamaan:
  - Gunakan gaya penamaan yang konsisten di seluruh program.

Tips

- Jangan Takut Menulis Nama yang Panjang:
  - Lebih baik memiliki nama yang panjang dan jelas daripada nama yang ambigu.
- Gunakan Bahasa atau istilah yang Baku:
  - Jika memungkinkan, utamakan menggunakan bahasa Inggris, agar bisa dipahami oleh banyak pihak
  - Hindari penggunaan bahasa lokal atau singkatan yang tidak universal.
- Hindari Nama Variabel yang Terlalu Umum:
  - Nama variabel yang terlalu umum seperti **`$data`** atau **`$temp`** dapat membingungkan.
- Pilih Nama yang Menyiratkan Tipe Data atau Isi Variabel:
  - Misalnya, **`$angka`**, **`$string`**, atau **`$arrayData`**.
- Perbarui Nama Saat Kode Berkembang:
  - Jika fungsionalitas berubah, pastikan nama tetap mencerminkan tujuan saat ini.
  - Jika tujuan atau fungsi variabel berubah, pastikan untuk memperbarui nama variabel.

Contoh

- Variabel untuk Menyimpan Usia:

```
$usiaPengguna = 25;
```

- Variabel untuk Nama Pengguna:

```
$namaPengguna = "JohnDoe";
```

- Variabel untuk Menghitung Total Belanja:

```
$totalBelanja = 150.75;
```

Contoh

- Konstanta untuk Nilai PI:

```
define("PI", 3.14);
```

- Konstanta untuk Panjang Maksimum Nama Produk:

```
define("MAX_NAMA_PRODUK", 50);
```

- Konstanta untuk Warna Utama:

```
define("WARNA_UTAMA", "#3498db");
```

Contoh

- Konstanta untuk Status Aktif:

```
define("STATUS_AKTIF", 1);
```

- Konstanta untuk Versi Aplikasi:

```
define("VERSI_APLIKASI", "1.2.3");
```

Contoh

- Fungsi untuk Menambah Dua Angka:

```
function tambahAngka($angka1, $angka2) {
    // ...
}
```

- Fungsi untuk Memeriksa Kevalidan Email:

```
function validasiEmail($email) {
    // ...
}
```

Contoh

- Fungsi untuk Menghitung Diskon:

```
function hitungDiskon($totalHarga, $persentaseDiskon) {
    // ...
}
```

- Prosedur untuk Menampilkan Pesan Selamat Datang:

```
function sambutPengguna($nama) {
    // ...
}
```

Contoh

- Fungsi untuk Menggabungkan Dua String:

```
function gabungString($str1, $str2) {
    // ...
}
```

- Fungsi untuk Menghitung Luas Persegi Panjang:

```
function hitungLuasPersegiPanjang($panjang, $lebar) {
    // ...
}
```

Memberi Komentar

Mengelola Sumber Daya Pemrograman

Pendahuluan

- Komentar adalah bagian kritis dalam penulisan kode.
- Komentar akan memberikan informasi tambahan yang membantu programmer atau orang lain yang membaca kode untuk memahami tujuan dan cara kerja suatu bagian dari program.
- Dalam fungsi atau prosedur, komentar membantu menjelaskan logika, alur kontrol, dan tujuan dari setiap blok kode.

Penulisan Komentar

- Komentar Satu Baris:
  - Gunakan tanda “//” untuk komentar satu baris

```
// Ini adalah komentar satu baris
```

- Komentar Multibaris:
  - Gunakan tanda “/* */” untuk komentar yang melibatkan beberapa baris

```
/*
   Ini adalah komentar
   lebih dari satu baris
*/
```

Tujuan Penggunaan Komentar

- Deskripsi
  - Memberikan penjelasan tujuan umum dari suatu bagian, fungsi, atau prosedur

```
/**
 * Fungsi untuk menghitung jumlah total barang
 * dalam keranjang belanja.
 */
function hitungTotalBarang() {
   // ... logika perhitungan
}
```

Tujuan Penggunaan Komentar

- Penjelasan Parameter
  - Memberikan penjelasan setiap parameter dan tipe data pada fungsi, atau prosedur

```
/**
 * Fungsi untuk menggabungkan dua string.
 *
 * @param string $str1 String pertama.
 * @param string $str2 String kedua.
 * @return string Hasil penggabungan kedua string.
 */
function gabungString($str1, $str2) {
    // ... logika penggabungan
}
```

Tujuan Penggunaan Komentar

- Keterangan Logika Internal
  - Memberikan penjelasan langkah-langkah atau keputusan logika yang diambil dalam fungsi atau prosedur

```
/**
 * Fungsi untuk memeriksa apakah angka genap.
 *
 * @param int $angka Angka yang diperiksa.
 * @return bool True jika genap, false jika ganjil.
 */
function cekGenap($angka) {
    // ... logika pemeriksaan
}
```

Contoh Komentar pada Fungsi

```
/**
 * Fungsi untuk menghitung luas segitiga.
 *
 * @param float $alas Panjang alas segitiga.
 * @param float $tinggi Tinggi segitiga.
 * @return float Luas segitiga.
 */
function hitungLuasSegitiga($alas, $tinggi) {
    // Rumus luas segitiga
    $luas = 0.5 * $alas * $tinggi;

    return $luas;
}
```

Contoh Komentar pada Prosedur

```
/**
 * Prosedur untuk mencetak pesan selamat datang.
 *
 * @param string $nama Nama pengguna.
 */
function sambutPengguna($nama) {
    // Cetak pesan selamat datang
    echo "Selamat datang, $nama!";
}
```

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

