

# Modul Pelatihan

## Vocational School Graduate Academy

### Tema

Digital Talent Scholarship 2020



## KATA PENGANTAR

Era Digitalisasi pada Industri 4.0 di Indonesia saat ini dihadapkan pada tantangan hadirnya permintaan dan penawaran talenta digital dalam mendukung perkembangan ekosistem industri teknologi. Tantangan tersebut perlu dihadapi salah satunya melalui kegiatan inovasi dan inisiasi dari berbagai pihak dalam memajukan talenta digital Indonesia, baik dari pemerintah maupun mitra kerja pemerintah yang dapat menyiapkan angkatan kerja muda sebagai talenta digital Indonesia. Kementerian Komunikasi dan Informatika melalui Badan Penelitian dan Pengembangan Sumber Daya Manusia sejak tahun 2018-2019, telah menginisiasi Program Digital Talent Scholarship yang telah berhasil dianugerahkan kepada 26.000 penerima pelatihan di bidang teknologi informasi dan komunikasi. Program Digital Talent Scholarship ini ditujukan untuk memberikan pelatihan dan sertifikasi tema-tema bidang teknologi informasi dan komunikasi, diharapkan menjadi bagian untuk memenuhi permintaan dan penawaran talenta digital Indonesia.

Tahun ini, Program Digital Talent Scholarship menargetkan pelatihan peningkatan kompetensi bagi 60.000 peserta yang bertujuan untuk meningkatkan keterampilan dan daya saing SDM bidang teknologi informasi dan komunikasi sebagai bagian dari program pembangunan prioritas nasional. Program pelatihan DTS 2020 ditujukan untuk meningkatkan keterampilan, keahlian angkatan kerja muda Indonesia, masyarakat umum dan aparatur sipil negara di bidang teknologi informasi dan komunikasi sehingga dapat meningkatkan produktivitas dan daya saing bangsa di era Industri 4.0.

Program DTS 2020 secara garis besar dibagi menjadi Tujuh akademi, yaitu: Fresh Graduate Academy (FGA), Program pelatihan berbasis kompetensi bersama GlobalTech yang ditujukan kepada peserta pelatihan bagi lulusan S1 bidang TIK dan MIPA, dan terbuka bagi penyandang disabilitas; Vocational School Graduate Academy (VSGA), Program pelatihan berbasis kompetensi nasional yang ditujukan kepada peserta pelatihan bagi lulusan SMK dan Pendidikan Vokasi bidang TI, Telekomunikasi, Desain, dan Multimedia; Coding Teacher Academy (CTA), Program pelatihan merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan kepada peserta pelatihan bagi Guru setingkat SMA/SMK/MA/SMP/SD di bidang pemrograman. Online Academy (OA), Program pelatihan OA merupakan program pelatihan Online di bidang Teknologi Informasi yang ditujukan kepada peserta pelatihan bagi Masyarakat umum, ASN, mahasiswa, dan pelaku industri; Thematic Academy (TA), Program pelatihan TA merupakan program pelatihan multisektor bagi pengembangan sumberdaya manusia yang ditujukan kepada peserta pelatihan dari jenjang dan multidisiplin yang berbeda; Regional Development Academy (RDA), Program pelatihan RDA merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan untuk meningkatkan kompetensi ASN di Kawasan Prioritas Pariwisata dan 122 Kabupaten Prioritas Pembangunan. Digital Entrepreneurship Academy (DEA), Program pelatihan DEA merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan kepada talenta digital di bidang Usaha Mikro, Kecil, dan Menengah (UMKM).

Selamat mengikuti Pelatihan Digital Talent Scholarship, mari persiapkan diri kita menjadi talenta digital Indonesia.

Jakarta, 2020  
Kepala Badan Penelitian dan Pengembangan Sumber Daya Manusia  
Kementerian Komunikasi dan Informatika Republik Indonesia  
**Dr. Ir. Basuki Yusuf Iskandar, MA**

## Pendahuluan

Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membuat website.

### A. Tujuan Umum

Setelah mempelajari modul ini peserta latih diharapkan mampu membuat website sederhana.

### B. Tujuan Khusus

Adapun tujuan mempelajari unit kompetensi melalui buku informasi Pelatihan Web Developer ini guna memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan dalam Menulis kode dengan prinsip sesuai guidelines dan best practices.

## Latar belakang

Unit kompetensi ini dinilai berdasarkan tingkat kemampuan dalam merancang website. Adapun penilaian dilakukan dengan menggabungkan serangkaian metode untuk menilai kemampuan dan penerapan pengetahuan pendukung penting. Penilaian dilakukan dengan mengacu kepada Kriteria Unjuk Kerja (KUK) dan dilaksanakan di Tempat Uji Kompetensi (TUK), ruang simulasi atau workshop dengan cara:

- 1.1 Lisan
- 1.2 Wawancara
- 1.3 Tes tertulis
- 1.4 Demonstrasi
- 1.5 Metode lain yang relevan.

## Deskripsi Pelatihan

Menerapkan coding-guidelines dan best practices dalam penulisan program.

## Tujuan Pembelajaran

### A. Tujuan Umum

Setelah mempelajari modul ini peserta latih diharapkan mampu membuat aplikasi berbasis web sederhana.

### B. Tujuan Khusus

Adapun tujuan mempelajari unit kompetensi melalui buku informasi Pelatihan Junior Web Deveoper ini guna memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan sebagai berikut:

1. Menulis kode dengan prinsip sesuai guidelines dan best practices, termasuk diantaranya adalah menerapkan coding-guidelines dan best practices dalam penulisan program (kode sumber) dan menggunakan ukuran performasi dalam menuliskan kode sumber.

### Kompetensi Dasar

Mampu menjelaskan *websites development* dengan benar

### Indikator Hasil Belajar

Dapat menulis kode dengan prinsip sesuai guidelines dan best practices, termasuk diantaranya adalah menerapkan coding-guidelines dan best practices dalam penulisan program (kode sumber) dan menggunakan ukuran performasi dalam menuliskan kode sumber.

## INFORMASI PELATIHAN

|  |  |
|--|--|
| Akademi  | Vocational School Graduate Academy   |
| Mitra Pelatihan  | Perguruan Tinggi   |
| Tema Pelatihan   | <b>Web Developer</b>   |
| Sertifikasi  | <i>Sertifikasi Kompetensi Junior Web Developer dari BNSP</i>   |
| Persyaratan Sarana Peserta/spesifikasi device Tools/media ajar yang akan digunakan | Laptop dengan spesifikasi:<br>1. RAM minimal 2 GB (disarankan 4 GB)<br>2. Laptop dengan 32/64-bit processor<br>3. Laptop dengan Operating System Windows 7,8,10<br>4. Laptop dengan konektivitas WiFi dan memiliki Webcam<br>5. Akses Internet Dedicated 126 kbps per peserta per perangkat<br>6. Sudah terinstall Software XAMPP dan Text Editor Sublime Text |
| Aplikasi yang akan di gunakan selamat pelatihan                                    | Sublime Text, Web Browser, XAMPP   |
| Tim Penyusun   | <ul style="list-style-type: none"><li>• Dr. Ir. Eko Kuswardono Budiardjo, M.Sc. (Universitas Indonesia);</li><li>• Ir. Windy Gambetta MBA (ITB);</li></ul>   |

|  |   |
|--|---|
|  | <ul style="list-style-type: none"> <li>• I Komang Sugiarta, S.Kom., MMSI (Universitas Gunadarma);</li> <li>• Agus Suwondo, SKom., MKom. (Politeknik Negeri Semarang);</li> <li>• Airlangga Adi Hermawan (Vokasi UGM);</li> <li>• Alfrets Wauran, ST.,MCSE (Politeknik Negeri Manado);</li> <li>• Devit Suwardiyanto, S.Si., M.T. (Politeknik Negeri Banyuwangi);</li> <li>• Dyah Puspito Dewi Widowati (BPPTIK Cikarang);</li> <li>• Freska Rolansa (Politeknik Negeri Pontianak);</li> <li>• Hamdani Arif (Politeknik Negeri Batam);</li> <li>• Hermawan Arief Putranto, ST, MT (Politeknik Negeri Jember);</li> <li>• I Nyoman Eddy Indrayana, S.Kom., MT (Politeknik Negeri Bali);</li> <li>• Iklima Ermis Ismail, S.Kom., M.Kom. (Politeknik Negeri Jakarta);</li> <li>• Marion Erwin Dien, S.Kom, M.Cs (Politeknik Negeri Ambon);</li> <li>• Nicodemus M.Setiohardjo,S.Kom,M.Cs (Politeknik Negeri Kupang);</li> <li>• Rheo Malani (Politeknik Negeri Samarinda);</li> <li>• Salahuddin, ST, M.Cs. (Politeknik Negeri Lhokseumawe);</li> <li>• Subandi, ST., M.Kom (Politeknik Negeri Banjarmasin);</li> </ul> |
|--|---|

## INFORMASI PEMBELAJARAN

| Unit Kompetensi  | Materi pembelajaran                   | Kegiatan pembelajaran | Durasi Pelatihan                          | Rasio Praktek : Teori | Sumber pembelajaran |
|--|---------------------------------------|-----------------------|---|-----------------------|---------------------|
| Menulis kode dengan prinsip sesuai guidelines dan best practices | Modul dan Slide dalam membuat website | Daring/Online         | Live Class 2 JP<br>LMS 4 JP<br>@ 45 Menit | 60:40                 |                     |

### Materi Pokok

Penulisan kode dengan prinsip sesuai guidelines dan *best practices*.

### Sub Materi Pokok

Menerapkan coding-guidelines dan best practices dalam penulisan program (kode sumber)

Menggunakan ukuran performansi dalam menuliskan kode sumber  
Menggunakan tipe data dan control program

## **MENULIS KODE DENGAN PRINSIP SESUAI GUIDELINES DAN BEST PRACTICES**

### **A. Pengetahuan yang diperlukan dalam menulis kode dengan prinsip sesuai guidelines dan best practices**

#### **1. Menerapkan coding- guidelines dan best practices dalam penulisan program (kode sumber)**

##### **Coding-standard HTML**

- Semua kode HTML yang akan anda tulis harus valid dan terstruktur.
- Sebelum mulai menulis kode program, anda harus memeriksa spesifikasi proyek yang anda kerjakan dengan spesifikasi HTML yang anda gunakan.
- Spesifikasi yang sekarang biasa digunakan adalah HTML5 Document Type Definition.

```
<!DOCTYPE HTML>
```

##### **Gunakan lower case untuk nama**

- Elemen dan atribut harus ditulis menggunakan huruf lower case.

```
<!-- Correct -->  
<input name="name" type="text" />  
  
<!-- Wrong -->  
<input name="name" TYPE="text" />
```

## Tag Penutup

- Baik elemen yang kosong maupun yang tidak kosong, harus memiliki tag penutup yang sesuai.
- Elemen yang tidak kosong

```
<h1>My title</h1>
<p>Some text</p>
```

- Elemen yang kosong

```
<span></span>
```

## Tag Penutup (2)

- Khusus untuk elemen dengan *single tag* harus menggunakan > di akhir statement.

```
<br>
<hr>

```

## Elemen Bersarang

Elemen bersarang (nested element) harus ditulis dengan benar, sesuai dengan susunan tag-nya.

- **Penulisan yang benar**

```
<!-- Correct -->
<div>
  <p>Some text</p>
</div>
```

- **Penulisan yang salah**

```
<!-- Wrong -->
<div>
  <p>Some text</div>
</p>
```

### **Nilai dari Atribut**

- Menuliskan nilai (value) dari sebuah atribut harus menggunakan tanda petik dua [“ ”], walaupun nilai tersebut berupa nilai numerik.

```
<!-- Correct -->
<input name="age" type="text" size="3" />

<!-- Wrong -->
<input name=age type=text size=3 />
```

### **Indentasi**

- Gunakan 2 spasi untuk indentasi kode.
- Gunakan indentasi secara konsisten untuk memudahkan dalam membaca kode.
- Ketika ada sebuah elemen yang didalamnya terdapat lebih dari satu baris kode, berikan indentasi pada konten elemen, antara tag awal dan tag akhir, untuk memudahkan dalam melihat di mana elemen dimulai dan berakhir.



```

<div class="container">
  <header class="header">
    <h1>Site Name<span></span></h1>
  </header>
  <!-- / header -->
  <hr>
  <nav class="navigation">
    <ul>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </nav>
  <!-- / navigation -->
</div>
<!-- / container -->

```

## Encoding dan Charset

- Gunakan encoding standard untuk dokumen HTML anda, yaitu UTF-8 Normalization Form C (NFC).

```

<meta charset="utf-8" />

```

- Beberapa contoh untuk encoding character yang biasa digunakan

```

&amp;
&copy;
&raquo;
&gt;

```

## HTML Anchors

- Saat Anda ingin memberi tautan ke bagian di dalam dokumen HTML, gunakan atribut ID.

```
<a href="#section">link</a>
<div id="section"></div>
```

- Jika tidak mungkin menggunakan ID (misalnya karena platform ASP.NET), gunakan atribut *named anchors*.

```
<a href="#section">link</a>
<a name="section"></a>
```

## Komentar

Tuliskan komentar pada bagian akhir setelah tag penutup dari bagian HTML, hindari penulisan komentar pada awal bagian HTML.

```
<ol class="accessibility-nav">
  <li><a href="#navigation">Skip to navigation</a></li>
  <li><a href="#content">Skip to content</a></li>
  <li><a href="#sidebar">Skip to sidebar</a></li>
</ol>
<!-- / accessibility-nav -->

<p>
  <a href="#" title="Go to homepage"><em>Home</em></a>
</p>
<!-- / breadcrumb -->
```

## Aksesibilitas

- Gunakan h1 - h6 untuk mengidentifikasi heading
- Gunakan elemen list (ul, ol, li) untuk mengelompokkan tautan
- Berikan definisi singkatan dengan menggunakan elemen abbr dan akronim
- Gunakan atribut bahasa pada elemen html untuk mengidentifikasi bahasa default suatu dokumen HTML

## Table

1. Gunakan elemen table `<table></table>` untuk menyajikan informasi dalam bentuk tabel
2. Gunakan atribut `scope` untuk mengaitkan sel header dan sel data dalam tabel data
3. Gunakan atribut `summary` dari elemen tabel untuk memberikan gambaran umum tentang data table

```
</tr><tr>
  <td>2.</td>
  <td scope="row">Clive Lloyd</td>
  <td>410-306-1420</td>
  <td>410-306-5400</td>
  <td>Baltimore</td>
</tr><tr>
  <td>3.</td>
  <td scope="row">Gordon Greenidge</td>
  <td>281-564-6720</td>
  <td>281-511-6600</td>
  <td>Houston</td>
```

## Form

- Dalam sebuah form selalu sediakan tombol submit.
- Gunakan atribut `alt` pada gambar yang digunakan sebagai tombol kirim.
- Gunakan elemen label untuk mengaitkan label teks dengan kontrol formulir.
- Gunakan atribut `judul` untuk mengidentifikasi kontrol formulir ketika elemen label tidak dapat digunakan.

## Coding-standard untuk CSS

### Format CSS

- Semua dokumen CSS harus menggunakan dua spasi untuk indentasi dan file tidak boleh memiliki dua spasi tambahan (whitespace).
- Gunakan tanda kutip ganda.
- Gunakan notasi steno hanya jika diperlukan.
- Beri spasi setelah: dalam deklarasi properti.
- Letakkan spasi sebelum {dalam deklarasi aturan.
- Gunakan kode warna hex # 000 kecuali menggunakan rgba ().
- Selalu berikan properti fallback untuk browser lama.
- Gunakan satu baris per deklarasi properti.
- Selalu ikuti aturan dengan satu baris spasi.
- Selalu mengutip konten url () dan @import ().
- Jangan membuat indentasi blok.

### Format CSS (Contoh)

```
.media {
  overflow: hidden;
  color: #fff;
  background-color: #000; /* Fallback value */
  background-image: linear-gradient(black, grey);
}

.media .img {
  float: left;
  border: 1px solid #ccc;
}

.media .img img {
  display: block;
}

.media .content {
  background: #fff url("../images/media-background.png") no-repeat;
}
```

### Pemberian Nama

Semua nama yang digunakan untuk id, kelas dan atribut harus menggunakan huruf kecil, dan menggunakan tanda hubung apa bila diperlukan pemisahan.

```

/* GOOD */
.dataset-list {}

/* BAD */
.datasetlist {}
.datasetList {}
.dataset_list {}

```

## Komentar

Gunakan komentar untuk menjelaskan apa pun yang memiliki makna ganda atau tidak mudah dipahami orang lain yang membaca kode CSS anda.

```

.prose p {
  font-size: 1.1666em /* 14px / 12px */;
}

.ie7 .search-form {
  /*
    Force the item to have layout in IE7 by setting display to block.
    See: http://reference.sitepoint.com/css/haslayout
  */
  display: inline-block;
}

```

## Coding Standard untuk PHP

### Tag PHP

- Tag Pembuka HARUS berada pada barisnya sendiri dan HARUS diikuti oleh baris kosong.

Salah karena <?php tidak pada jalurnya sendiri.

```
< ? php print_welcome_message ();
```

Salah karena <?php tidak diikuti oleh baris kosong

```
<? php  
print_welcome_message ();
```

BENAR

```
<? php  
  
print_welcome_message ();
```

### Tag Penutup

- BENAR

Tidak menggunakan ?>

```
<? php  
  
print_welcome_message ();
```

- SALAH

Salah karena ?> digunakan.

```
<? php  
  
print_welcome_message ();  
  
? >
```

Tag Penutup TIDAK HARUS digunakan dalam file PHP, namun digunakan apabila ada kode PHP yang dituliskan dalam HTML (inline).

### Tag Pembuka dan Penutup (inline PHP)

- BENAR

Karena tag <?php dan ?> berada dalam satu baris

```
< div >
    < h1 > <? php print_welcome_message (); ?> </ h1 >
</ div >
```

- SALAH

Salah karena tag <?php dan ?> tidak dalam satu baris

```
< div >
    < h1 > <? php
    print_welcome_message ();
    ?> </ h1 >
</ div >
```

### Tag pembuka pendek (short open tag)

Tag terbuka pendek sebaiknya TIDAK digunakan.

- BENAR

```
<? php

print_welcome_message ();
```

- SALAH

```
< ?

print_welcome_message ();
```

### Tag echo pendek (short echo tag)

- Tag gema pendek sebaiknya digunakan dalam file PHP yang berada di dalam HTML (inline).
- Masih dapat diterima,

```
< div >
    < p > <? php echo get_welcome_message (); ?> </ p >
</ div >
```

- Namun sebaiknya

```
< div >
    < p > <? = get_welcome_message (); ?> </ p >
</ div >
```

### Akhir dari File

- Bagian ini menjelaskan bagaimana setiap file PHP harus diakhiri.
- Cara penulisan komentar akhir file:
  - HARUS dimasukkan di akhir file
  - HARUS berada di jalurnya sendiri
  - HARUS didahului dan diakhiri oleh garis-garis kosong

```
<? php

print_welcome_message ();

// EOF
```

### Namespace

- Deklarasi Namespace HARUS menjadi pernyataan pertama dan HARUS diikuti oleh baris kosong.



- Nama namespace HARUS dimulai dengan huruf kapital dan HARUS menjadi camelcase.
- Bila ada lebih dari satu namespace HARUS menggunakan sintaks kurung kurawal.
- Konstanta ajaib (magic constant) HARUS digunakan untuk referensi nama namespace.

Contoh:

```
<? php

namespace MyCompany \ Model {
// ModuleOne body }

namespace MyCompany \ Lihat {
$ welcome_message = __NAMESPACE__ . ' \ ' . get_welcome_message (); }

// EOF
```

### Komentar

- Komentar satu baris HARUS menggunakan dua garis miring ke depan ( // ).
- Komentar multi-baris HARUS menggunakan format blokir ( /\* ..... \*/ )
- Komentar header HARUS menggunakan format blokir.
- Komentar pembagi HARUS menggunakan format blok dengan 75 tanda bintang di antaranya.
- Komentar HARUS ada di jalur mereka sendiri.
- Blok kode HARUS dijelaskan atau dirangkum.
- Angka ambigu HARUS diklarifikasi.
- Variabel eksternal HARUS diklarifikasi.

### Keyword include/require once

- Biasakan selalu menggunakan include\_once atau require\_once
- Tidak perlu menggunakan parenthesis / tanda kurung buka-kurung tutup

- Tujuan penggunaan include harus terdokumentasikan menggunakan komentar.

```
<?php

// Provides XYZ framework
require_once 'some-file.php';

// EOF
```

### Formatting

- Panjang baris tidak boleh melebihi 80 karakter, kecuali baris tersebut berupa text.
- Indentasi baris harus menggunakan tab.
- Baris kosong (dua kali enter) harus ditambahkan diantara dua kode logical block
- Meratakan teks harus menggunakan spasi
- Dua spasi kosong tidak boleh digunakan setelah pernyataan (statement) atau dua koma atau baris baru.
- Kata kunci (keyword) harus menggunakan huruf kecil (lowercase)
- Penggunaan operator HARUS diawali dan diakhiri oleh spasi.
- Operator unary HARUS disertakan ke variabel atau integer mereka.
- Penggabungan 2 statement atau lebih menggunakan tanda titik HARUS diawali dan diakhiri dengan spasi.
- Sebaiknya menggunakan tanda petik tunggal.
- Sebaiknya tidak menggunakan tanda petik ganda.

```
<?php

$my_movies = array(
    'Slumdog Millionaire',
    'Silver Linings Playbook',
    'The Lives of Others',
    'The Shawshank Redemption'
);

$has_all_movies = true;

foreach($my_movies as $my_movie) {
    if(!in_array($my_movie, $movies)) {
        $has_all_movies = false;
    }
}

if($has_all_movies) {
    // if body
}

// EOF
```

- Contoh penulisan kode yang salah, karena perataan text harus menggunakan spasi, bukan tab.

```
<?php

$movie_quotes = array(
    'slumdog_millionaire' => 'When somebody asks me a question, I tell them the answer.',
    'silver_linings_playbook' => 'I opened up to you, and you judged me.',
    'the_lives_of_others' => 'To think that people like you ruled a country.',
    'the_shawshank_redemption' => 'Get busy living, or get busy dying.'
);

// EOF
```

- Contoh penulisan kode yang benar, karena indentasi text menggunakan tab dan perataan text menggunakan spasi.

```
<?php

$movie_quotes = array(
    'slumdog_millionaire' => 'When somebody asks me a question, I tell them the answer.',
    'silver_linings_playbook' => 'I opened up to you, and you judged me.',
    'the_lives_of_others' => 'To think that people like you ruled a country.',
    'the_shawshank_redemption' => 'Get busy living, or get busy dying.'
);

// EOF
```

## Fungsi

- Nama fungsi harus menggunakan huruf kecil dan kata-kata harus dipisahkan oleh garis bawah

Contoh: function welcome\_message() {

- Awalan fungsi HARUS dimulai dengan kata kerja  
Contoh: get\_, add\_, update\_, delete\_, convert\_, dll
- Pemanggilan fungsi tidak boleh menggunakan spasi antara nama fungsi dan tanda kurung terbuka  
Contoh: func();
- Argumen didalam fungsi harus dituliskan sbb:
- Tidak boleh menggunakan spasi sebelum tanda koma.
- Harus menggunakan spasi setelah tanda koma.
- Boleh menggunakan baris baru untuk argument yang Panjang.
- Setiap argument harus dituliskan di baris yang berbeda
- Harus menggunakan satu kali indentasi untuk argument dibawahnya
- Harus diurutkan dari yang paling dibutuhkan sampai argument pilihan.
- Harus diurutkan berdasarkan argumen yang paling penting ke argument yang tidak begitu penting.
- Harus menggunakan standar deskriptif.
- HARUS menyertakan petunjuk jenis.  
misalnya func(\$arg1, \$arg2 = 'asc', \$arg3 = 100);
- Deklarasi fungsi HARUS didokumentasikan menggunakan pedoman
  - phpDocumentor dan HARUS meliputi:
    - Deskripsi Singkat
    - Deskripsi Panjang, jika perlu
    - @access: private atau protected (diasumsikan public)
    - @ penulis: nama penulis
    - @ global: menggunakan fungsi variabel global, jika ada
    - @param: Parameter dengan tipe data, nama variabel, dan deskripsi
    - @ return: Kembalikan tipe data, jika ada
- Contoh:

```

<? php

/ **
 * Dapatkan foto dari penulis blog
 *
 * Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut id volutpat
 * orci. Etiam pharetra bukan turpis ultrices. Vitae Pellentesque risus
 * sagittis, vehicula massa eget, tincidunt ligula.
 *
 * @access private
 * @author Firstname Lastname
 * @global object $ post
 * @param int $ id ID Penulis
 * @param string $ jenis Jenis foto
 * @par int $ lebar Foto lebar px
 * @ par int $ tinggi Foto tinggi dalam px
 * @return objek Foto
 * / fungsi my_function ( $ id , $ type , $ width , $ height ) { // ... return $ Photo ; }

// EOF

```

## Function Return

- Sebisamungkin segera di tampilkan.
- Variable yang menghasilkan return value Harus diinisialisasikan sebelumnya.
- HARUS diawali dengan baris baru, kecuali apabila function return berada didalam control statement.

```

<?php

function get_movies() {
    $movies = array();
    // ...

    return $movies;
}

// EOF

```

## Struktur Kontrol

- Kata kunci HARUS diikuti oleh spasi  
misalnya if (, switch (, do {, for (
- Setelah tanda kurung buka tidak boleh diikuti oleh spasi  
misalnya (\$expr,(\$i
- Tanda kurung tutup tidak boleh diawali dengan spasi  
misalnya \$expr), \$i++),\$value)

- Tanda Kurung Kurawal buka HARUS diawali dengan spasi dan HARUS diikuti oleh baris baru  
misalnya \$expr) { , \$i++) {
- Struktur tubuh HARUS di-indentasi sekali dan HARUS ditutup dengan kurung kurawal (tidak ada tulisan cepat)  
misalnya if (\$expr) { ↵ ... ↵ }
- Penutupan brace HARUS dimulai pada baris berikutnya yaitu ... ↵ }
- Struktur control yang bersarang tidak boleh melebihi tiga level  
misalnya if (\$expr1) { if (\$expr2) { if (\$expr3) { if (\$expr4) { ... } } } }

### If, Elseif, Else

- Biasakan menggunakan elseif sebagai pengganti else if
- elseif dan else HARUS dituliskan diantara } dan { pada satu baris

```
if ( $ expr1 ) {
// if body } elseif ( $ expr2 ) { // elseif body } else { // else body }
```

### Switch, Case

- Case statement HARUS di-indentasi sekali
- Badan case HARUS di indentasi dua kali
- Break kata kunci HARUS di-indentasi dua kali
- Case logic HARUS dipisahkan oleh satu baris kosong

```
<?php
switch ($expr) {
    case 0:
        echo 'First case, with a break';
        break;

    case 1:
        echo 'Second case, which falls through';
        // no break
    case 2:
    case 3:
    case 4:
        echo 'Third case, return instead of break';
        return;

    default:
        echo 'Default case';
        break;
}

// EOF
```

### For, foreach

```
<?php

for ($i = 0; $i < 10; $i++) {
    // for body
}

foreach ($iterable as $key => $value) {
    // foreach body
}

// EOF
```

## Try, catch

```
<?php

try {
    // try body
}
catch (FirstExceptionType $e) {
    // catch body
}
catch (OtherExceptionType $e) {
    // catch body
}

// EOF
```

## Class

- Pemberian nama file untuk class hanya boleh berisi satu definisi dan HARUS diawali dengan kata kunci class-

Contoh: class User → class-user.php, class Office → class-office.php

- Class namespace HARUS didefinisikan dan HARUS menyertakan nama vendor

Contohnya: namespace MyCompany\Model;, namespace MyCompany\View;, namespace MyCompany\Controller;

```
<?php

namespace MyCompany\Model;

class User
{
    // ...
}

// EOF
```

- Nama class HARUS dimulai dengan huruf kapital dan apabila terdiri dari 2 kata HARUS menjadi camelcase  
Contoh: MyCompany
- Harus ada dokumentasi class dan HARUS menggunakan model tag phpDocumentor  
Contoh: @author, @global, @package
- Definisi class HARUS menempatkan kurung kurawal pada baris tersendiri  
Contoh: class User { ... }

### Properti dari Class

- Penamaan variable harus mengikuti standard yang ada.
- HARUS menetapkan visibilitas dari variable.
- Nama variable tidak boleh diawali dengan garis bawah jika tipe variable private atau protected.

```
<?php
namespace MyCompany\Model;

class User
{
    public $var1;
    protected $var2;
    private $var3;
}

// EOF
```

### Properti dari Method

- Penamaan method harus mengikuti standard penulisan fungsi.
- HARUS menetapkan visibilitas dari method.
- Nama method tidak boleh diawali dengan garis bawah jika tipe method private atau protected.



```

<?php

namespace MyCompany\Model;

class User
{
    // ...

    public function get_var1() {
        return $this->var1;
    }

    protected function get_var2() {
        return $this->var2;
    }

    private function get_var3() {
        return $this->var3;
    }
}

// EOF

```

## Instansiasi sebuah Class

- HARUS mengikuti penamaan variable yang standard
- HARUS menyertakan tanda kurung

```

<?php

$office_program = new OfficeProgram();

// EOF

```

## Best-Practice untuk PHP

### Inisialisasi variable

- Inisialisasi variable harus dilakukan pertama kali sebelum penggunaan variable tersebut
- Boleh saja:

```

<?php

if ($expr) {
    // ....
}

$movies = array();
$movies = get_movies();

// EOF

```

- Namun sebaiknya gunakan ini:

```

<?php

$movies = array();

if ($expr) {
    // ....
}

$movies = get_movies();

// EOF

```

### **Inisialisasi/ Urutan deklarasi variable, method dan properties**

- HARUS diawali dengan global, ikuti dengan konstanta, dan diakhiri dengan variabel lokal
- HARUS diawali dengan properti kemudian diikuti dengan method dalam class
- HARUS diawali dengan public, diikuti protected, dan diakhiri dengan private method dalam class
- HARUS sesuai dengan abjad dalam kelompoknya

```

<?php

global $app_config,
       $cache,
       $db_connection;

define('ENVIRONMENT', 'PRODUCTION');
define('MODE', 1);

$id = 0;
$firstname = '';
$lastname = '';

// EOF

```

```

<?php

namespace MyCompany\Model;

class Office
{
    private $id;
    private $name;
    private $status;

    public function get_id() {
        // ...
    }

    protected function get_status() {
        // ...
    }

    private function get_name() {
        // ...
    }
}

// EOF

```

## Penanganan Galat/Error

- Penanganan error adalah proses menangkap kesalahan pada program untuk diambil tindakan yang sesuai.
- Metode Penanganan Error
  - Menggunakan Function "die()"
  - Menulis function Penanganan Error sendiri
  - Error reporting

## Function "die()"

- Misalkan terdapat kode sbb:

```

<?php

$file=fopen("hello.txt","
r");

?>

```

- Jika file tersebut tidak ditemukan, maka akan keluar pesan berikut:

```
Warning: fopen(hello.txt) [function.fopen]: failed to
open stream:
No such file or directory in
/var/www/webfolder/test.php on line 2
```

### Penggunaan function die();

- Memeriksa keberadaan file terlebih dahulu

```
<?php

if(!file_exists("hello.txt"))
{
    die("File tidak
ditemukan");
} else {

$file=fopen("hello.txt","r");
}

?>
```

- Jika file tidak ditemukan, maka akan muncul pesan:

```
File tidak ditemukan
```

- Program akan berhenti ketika memanggil function die(); dan menampilkan pesan yang dapat dipahami oleh pengguna

### Metode Menulis function Penanganan Error sendiri

- Kita dapat menulis fungsi sendiri untuk menangani kesalahan apa pun

```
<?php
//function penanganan error
function customError($errno,
$errorstr) {
    echo "<b>Error:</b> [$errno]
$errorstr";
}

//set error handler

set_error_handler("customError");

//picu error
echo($test);

?>
```

```
Error: [8] Undefined variable: test
```

- Fungsi yang dibuat harus mampu menangani minimal dua parameter (tingkat kesalahan dan pesan kesalahan) tetapi dapat menerima hingga lima parameter (opsional: file, nomor baris, dan konteks kesalahan):
- Syntax:

```
error_function(error_level,error_message,error_file,error_line,
error_context)
```

- Parameter Kesalahan:

| Parameter     | Deskripsi   |
|---------------|---|
| error_level   | Wajib. Menentukan tingkat laporan kesalahan untuk kesalahan yang ditentukan pengguna. Harus berupa angka nilai. Lihat tabel di slide berikutnya untuk daftar tingkat laporan kesalahan. |
| error_message | Wajib. Pesan error yang disampaikan ke pengguna.  |
| error_file    | Tidak wajib. Menampilkan nama file dimana error terjadi   |
| error_line    | Tidak wajib. Menampilkan nomor baris dimana error terjadi.  |
| error_context | Tidak Wajib. Menampilkan sebuah array yang mengandung setiap variabel, dan nilainya yang digunakan ketika terjadi error.  |

- Level Kesalahan:

| Nilai | Konstanta | Deskripsi   |
|-------|-----------|---|
| 2     | E_WARNING | Kesalahan yang tidak fatal. Eksekusi program tidak terhenti.  |
| 8     | E_NOTICE  | Pemberitahuan Run-time. Dalam kode mungkin terdapat sebuah kesalahan, tetapi dapat juga terjadi ketika kode berjalan normal |

|      |                     |  |
|------|---------------------|--|
| 256  | E_USER_ERROR        | Kesalahan fatal yang dibuat oleh programmer. Ini seperti sebuah E_ERROR yang di set oleh programmer menggunakan function trigger_error()               |
| 512  | E_USER_WARNING      | Peringatan yang tidak fatal yang dibuat oleh programmer. Ini seperti sebuah E_WARNING yang di set oleh programmer menggunakan function trigger_error() |
| 1024 | E_USER_NOTICE       | Pemberitahuan yang dibuat oleh programmer. Ini seperti sebuah E_NOTICE yang di set oleh programmer menggunakan function trigger_error()                |
| 4096 | E_RECOVERABLE_ERROR | Kesalahan fatal yg dapat ditangkap. Ini seperti sebuah E_ERROR Tapi yang ditangkap oleh programmer dg mendefinisikan handle (set_error_handler())      |
| 8191 | E_ALL               | Semua kesalahan dan peringatan   |

### **Penanganan pengecualian (exception)**

- Pengecualian digunakan untuk mengubah alur program yg normal jika terjadi kesalahan (error) tertentu. Kondisi tersebut disebut pengecualian.
- Pada dasarnya yang terjadi saat exception dipicu adalah:
  - Keadaan kode yang paling baru (sebelum terjadi kesalahan) disimpan
  - Eksekusi kode akan beralih ke fungsi handler exception (custom) yang telah ditentukan
  - Bergantung pada situasinya, handler kemudian dapat melanjutkan eksekusi dari status kode yang disimpan, mengakhiri eksekusi program atau melanjutkan program dari lokasi yang berbeda dalam kode.

### **Penggunaan Exception dasar**

- Penulisan Exception yang standard harus mengandung:

- try - Fungsi yang menggunakan exception harus berada di dalam blok "try". Jika exception tidak terpicu, kode akan dilanjutkan seperti biasa. Namun jika exception terpicu, exception akan di-"throw".
- throw - Ini adalah bagaimana Anda memicu exception. Setiap "throw" harus memiliki setidaknya satu "catch".
- catch - Blok "catch" mengambil exception dan membuat objek yang berisi informasi dari exception.

### Contoh penggunaan Exception

```
<?php
//create function with an exception
function checkNum($number) {
    if($number>1) {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}

//trigger exception in a "try" block
try {
    checkNum(2);
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e) {
    echo 'Message: ' . $e->getMessage();
}
?>
```

Apabila dijalankan akan menghasilkan

---

```
Message: Value must be 1 or below
```

---

### Error Reporting

- Fungsi `error_reporting()` menentukan kesalahan mana yang dilaporkan.
- PHP memiliki banyak level kesalahan, dan menggunakan fungsi ini menetapkan level kesalahan tersebut untuk kode program yang sedang dijalankan.
- Syntax:

---

```
error_reporting(Level);
```

---

### Error Reporting (contoh)

```
<?php
// Turn off error reporting
error_reporting(0);

// Report runtime errors
error_reporting(E_ERROR | E_WARNING | E_PARSE);

// Report all errors
error_reporting(E_ALL);

// Same as error_reporting(E_ALL);
ini_set("error_reporting", E_ALL);

// Report all errors except E_NOTICE
error_reporting(E_ALL & ~E_NOTICE);
?>
```

### Kinerja Situs Web

- Ketika berbicara tentang kinerja aplikasi web, kita berbicara tentang **kinerja web** atau **kinerja runtime**.
- **Kinerja web** sebagai ukuran waktu mulai dari saat pengguna akhir meminta konten hingga kapan konten itu tersedia di perangkat pengguna.
- **Kinerja runtime** sebagai indikasi seberapa responsif aplikasi terhadap input pengguna saat runtime.

### Menggunakan ukuran performansi dalam menuliskan kode sumber

#### Mengukur Kinerja Program (kecepatan)

Waktu Eksekusi = Waktu Berakhir – Waktu Mulai



```
<?php
    $awal = microtime(true);
    // Berhenti untuk sesaat atau berisi
    blok kode program yg diukur
    usleep(1000);
    $akhir = microtime(true);
    $waktu = $akhir - $awal;
    echo "Program berjalan selama $waktu
    detik\n";
?>
```

### Mengukur Kinerja Program (penggunaan memory)

```
<?php
    /*
        Blok kode program yang akan diukur
        penggunaan memorinya
    */
    $penggunaan_memory =
    memory_get_usage();
    echo "Program menggunakan memory sebesar
    $penggunaan_memory bytes\n";
?>
```

### Menghitung efisiensi program

### Menulis kode PHP dengan Efisien

- Menggunakan Fungsi bawaan PHP

Menggunakan fungsi bawaan PHP lebih cepat dibanding menulis fungsi sendiri.

Referensi fungsi bawaan PHP dapat dilihat di manual PHP <https://www.php.net/manual/en/funcref.php>

- Membuat Kelas hanya jika diperlukan

Kelas dan method dibuat jika memang benar-benar diperlukan (digunakan kembali di banyak kode).

- Menutup Koneksi

Menutup koneksi basisdata setelah tidak dibutuhkan akan menghemat memory.

- Menggunakan petik tunggal

Ketika menggunakan string, penggunaan petik tunggal ( ' ' ) lebih cepat dari pada petik ganda ( " " ). Petik ganda akan memeriksa keberadaan variabel di dalamnya, sehingga butuh waktu lebih lama.

#### Lebih lambat

```
$awal = microtime(true);  
$i = 0;  
while($i < 1000) {  
    $tmp[] = "";  
    ++$i;  
}
```

```
$akhir = microtime(true);  
$waktu = $akhir - $awal;  
echo "Membutuhkan $waktu  
detik\n";
```

#### Lebih cepat

```
$awal = microtime(true);  
$i = 0;  
while($i < 1000) {  
    $tmp[] = '';  
    ++$i;  
}
```

```
$akhir = microtime(true);  
$waktu = $akhir - $awal;  
echo "Membutuhkan $waktu  
detik\n";
```

- **Mengurangi perhitungan yang tidak perlu**

Menghitung dan memberikan nilai ke variabel di awal, lebih cepat dari pada menghitungnya beberapa kali dimana perhitungan tersebut diperlukan.

#### Lebih lambat

```
$awal = microtime(true);  
$arrA =  
array(1,2,3,4,5,6,7,8,9);  
for( $i=0; i< count($arrA);  
$i++){  
    echo count($arrA);  
}  
$akhir = microtime(true);  
$waktu = $akhir - $awal;  
echo "Membutuhkan $waktu  
detik\n";
```

#### Lebih cepat

```
$awal = microtime(true);  
$arrA =  
array(1,2,3,4,5,6,7,8,9);  
$len = count($arrA);  
for( $i=0; i< $len; $i++){  
    echo $len;  
}  
$akhir = microtime(true);  
$waktu = $akhir - $awal;  
echo "Membutuhkan $waktu  
detik\n";
```

- **Menggunakan function isset()**

Gunakan `isset()` jika memungkinkan untuk memeriksa lebih besar dari 0.  
Hindari menggunakan `count()`, `strlen()`, `sizeof()`.

#### Lebih lambat

```
if(count($returnValue) >  
0){  
    // lakukan sesuatu  
}
```

#### Lebih cepat

```
if(isset($returnValue)){  
    // lakukan sesuatu  
}
```

- **Menggunakan Swith Case dibanding If**

Menggunakan pernyataan case lebih efisien dibandingkan struktur if/else untuk mengerjakan pekerjaan yang sama.

#### Lebih lambat

```
if($a == 0){  
    // lakukan tugas a  
}else if($a == 1){  
    // lakukan tugas b  
}else{  
    // lakukan tugas c  
}
```

#### Lebih cepat

```
Switch ($a){  
    case 0:  
        // lakukan tugas  
a  
    case 1:  
        // lakukan tugas  
b  
    default:  
        // lakukan tugas c  
}
```

- **Menggunakan Switch Case**

When using for loop, make sure the maximum number of executions instead of calculating maximum execution times in each loop. Use foreach if you can;

#### Lebih lambat

```
if($a == 0){  
    // lakukan tugas a  
}else if($a == 1){  
    // lakukan tugas b  
}else{  
    // lakukan tugas c  
}
```

Lebih cepat

```
Switch ($a){  
    case 0:  
        // lakukan tugas  
a  
    case 1:  
        // lakukan tugas  
b  
    default:  
        // lakukan tugas c  
}
```

### Mengimplementasikan Kemudahan Interaksi

**Desain User Interface (UI)** fokus untuk mengantisipasi apa yang mungkin dilakukan pengguna dan memastikan bahwa antarmuka memiliki elemen-elemen yang mudah diakses, difahami dan digunakan untuk memfasilitasi aktifitas pengguna.

Dalam membangun aplikasi yang memiliki kinerja tinggi, kita perlu memperhatikan juga kemudahan aplikasi ketika digunakan oleh pengguna.

Aplikasi yang dibuat harus dipastikan sesuai dengan panduan desain Antarmuka Pengguna (User Interface).

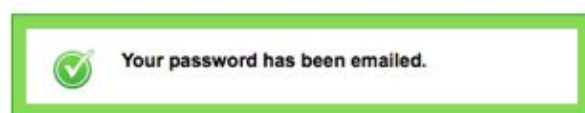
#### 10 Panduan desain antarmuka pengguna Nielsen dan Molich's

##### 1. Visibilitas status sistem.

Pengguna harus selalu diberitahu tentang status proses yang terjadi. Pemberitahuan harus mudah dimengerti, mudah terlihat di layar, dan ditampilkan dalam waktu yang wajar.

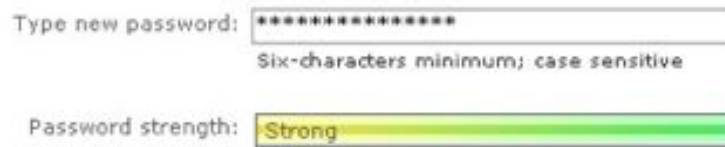


Indikator menunjukkan proses yang terjadi



Theresa Neil sign in

Pesan umpan balik ditampilkan ketika aksi telah dilakukan

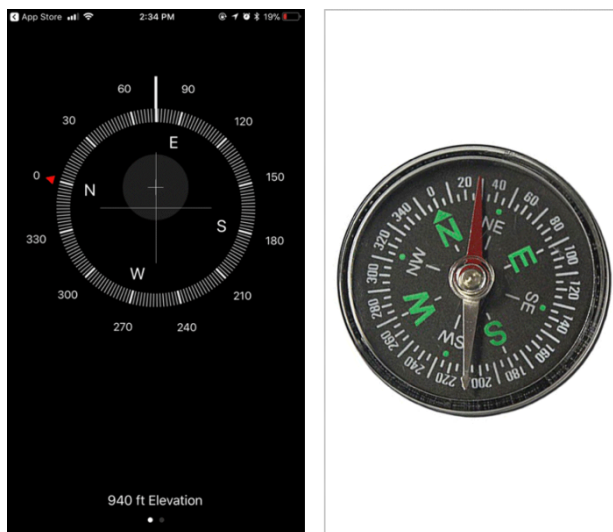


Kekuatan sandi ditunjukkan sesuai password yang dimasukkan

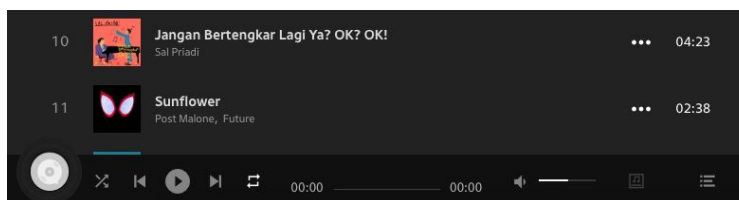
## 2. Aplikasi sesuai dengan dunia-nyata

Sistem harus menggunakan bahasa, kata-kata, frasa dan konsep yang familiar dengan pengguna.

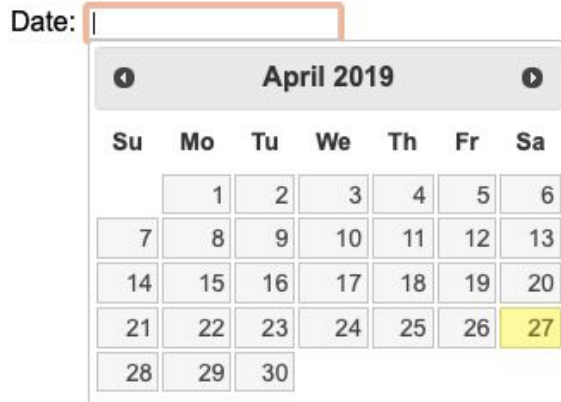
Menyajikan informasi dalam urutan logis dan mendukung ekspektasi pengguna yang berasal dari pengalaman mereka di dunia nyata akan membuat sistem lebih mudah digunakan.



Elemen antarmuka pengguna di aplikasi kompas, serupa dengan kompas sebenarnya, akan mempermudah pengguna untuk menggunakan



Penggunaan elemen ikon pemutar musik yang standar di perangkat musik akan mempermudah pengguna



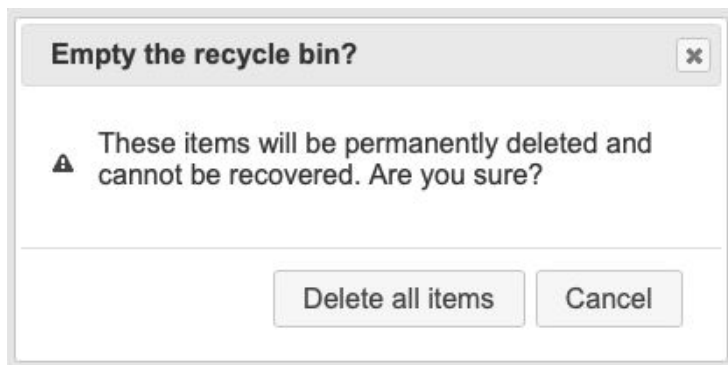
Penggunaan widget kalender dengan tampilan yang sama dengan kalender cetak

### 3. Kontrol dan Kebebasan Pengguna

Tawarkan ruang digital kepada pengguna agar memungkinkan langkah mundur, termasuk membatalkan dan mengulangi tindakan sebelumnya.



Fasilitas pencarian mudah dibuka, memasukkan kata kunci & filter, eksekusi atau membatalkan.



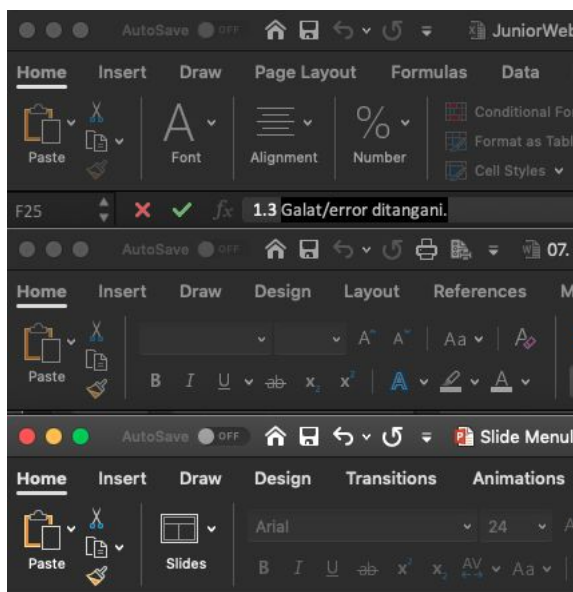
Memberi peringatan kepada pengguna untuk aktifitas yang beresiko. Apakah mau dilanjutkan

#### 4. Konsistensi dan Standar

Elemen grafis dan terminologi dipertahankan di seluruh platform yang sama. Misalnya, ikon yang mewakili satu kategori atau konsep tidak boleh mewakili konsep yang berbeda ketika digunakan pada layar yang berbeda.



Gmail menyusun organisasi folder yang sama dengan yang digunakan dalam aplikasi email klien (mis Outlook): Inbox, Draft, Sent Mail



MicrosoftOffice memiliki susunan menu yang serupa untuk aplikasi Word, Excell dan PowerPoint

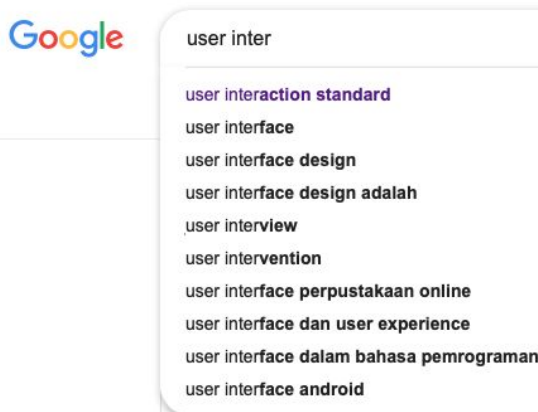
#### 5. Pencegahan Kesalahan

- Sistem dirancang agar potensi kesalahan dijaga seminimal mungkin.
- Pengguna tidak suka dipanggil untuk mendeteksi dan memperbaiki masalah, yang kadang-kadang berada di luar tingkat keahlian mereka.
- Menghilangkan atau menandai tindakan yang dapat menyebabkan kesalahan.





Buat tindakan utama menjadi menonjol dengan area klik yang lebih besar.  
Batalan dan tindakan sekunder hanya ditampilkan sebagai tautan



Rekomendasi otomatis untuk mencegah kesalahan masukan



Fokus otomatis pada input mencegah sumber frustrasi

## 6. Mengenali daripada mengingat

- Minimalkan beban memori pengguna. Buat objek, tindakan, dan opsi terlihat. Pengguna tidak harus mengingat informasi dari satu bagian dialog ke bagian lainnya. Petunjuk penggunaan sistem harus terlihat atau mudah diambil kapan pun diperlukan.

Fira Sans AaBbCcDdEeFfGgHhIiJjKkL

Z Y M m Fira Sans Mozilla 32 Styles DOWNLOAD OTF

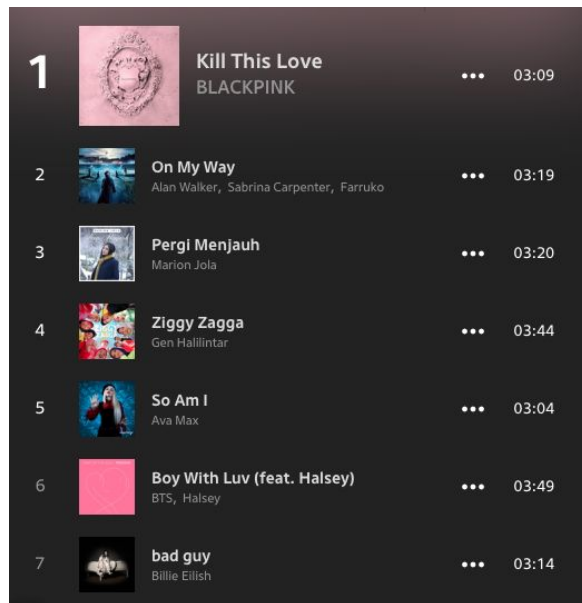
Corbert AaBbCcDdEeFfGgHhIiJj

Z Y M m Corbert The Northern Block 2 Styles DOWNLOAD OTF (OFFSITE)

Alex Brush AaBbCcDdEeFfGg

Z Y M m Alex Brush TypeSETit 1 Style DOWNLOAD OTF

Pratinjau font yang dapat dipilih, bukan hanya nama font.



Pratinjau gambar lebih mudah dikenali oleh user.

## 7. Fleksibilitas dan efisiensi penggunaan

- Interaksi yang lebih sedikit yang memungkinkan navigasi lebih cepat.
- Sehingga sistem dapat melayani pengguna yang tidak berpengalaman maupun yang berpengalaman.
- Izinkan pengguna untuk menyesuaikan tindakan yang sering dilakukan.
- Dapat menggunakan singkatan, tombol fungsi, perintah tersembunyi dan fasilitas makro

New

Edit

Delete

|                          | First name ▲ | Last name ▴ ▾ | Position ▴ ▾                  | Office ▴ ▾    |
|--------------------------|--------------|---------------|-------------------------------|---------------|
| <input type="checkbox"/> | Airi         | Satou         | Accountant                    | Tokyo         |
| <input type="checkbox"/> | Angelica     | Ramos         | Chief Executive Officer (CEO) | London        |
| <input type="checkbox"/> | Ashton       | Cox           | Junior Technical Author       | San Francisco |
| <input type="checkbox"/> | Bradley      | Greer         | Software Engineer             | London        |

Memperbaharui data secara langsung di tabel tanpa perlu melalui aksi lain

## Common Shortcuts

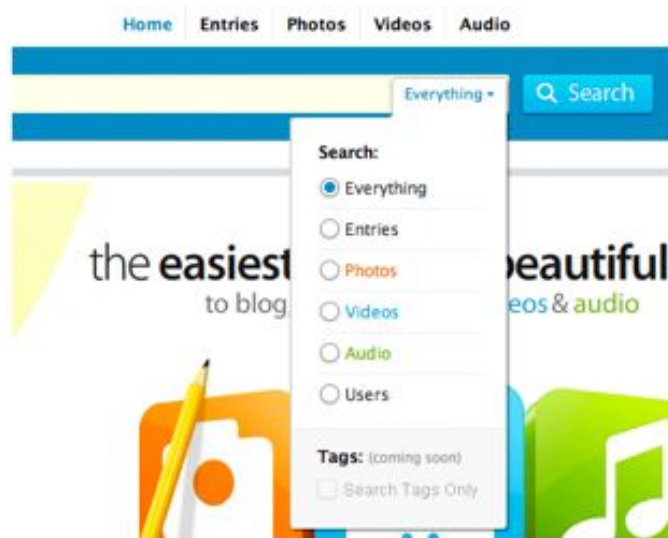
|                         |        |
|-------------------------|--------|
| Add Action              | Return |
| New Window              | ⌘N     |
| Synchronize with Server | ⌘S     |
| Clean Up                | ⌘K     |
| Planning Mode           | ⌘1     |
| Context Mode            | ⌘2     |
| Inbox                   | ⌘1     |
| Quick Entry             | ⌘Space |

*Quick Entry's shortcut can be customized in Preferences*

Penggunaan shortcut untuk menjalankan aksi

### 8. Desain estetis dan minimalis

- Interaksi/infor dengan pengguna dijaga agar tetap minimalis dengan tetap memperhatikan keindahan dan kegunaan
- Tampilan harus dikurangi menjadi hanya komponen yang diperlukan untuk tugas saat ini, sehingga memberikan cara yang jelas dan tidak ambigu untuk menavigasi ke konten lain.



Menu pencarian Kontain mencontohkan empat prinsip desain visual:

**Kontras:** teks tebal digunakan untuk dua label dalam pencarian

**Pengulangan:** teks oranye, biru, dan hijau cocok dengan jenis media

**Alignment:** alignment kiri yang kuat dari teks, rata bawah drop down

**Kedekatan:** aturan cahaya digunakan untuk memisahkan tag dari opsi lain

Show  entries Search:

| NAME           | POSITION                      | OFFICE        | AGE | START DATE | SALARY      |
|----------------|-------------------------------|---------------|-----|------------|-------------|
| Airi Satou     | Accountant                    | Tokyo         | 33  | 2008/11/28 | \$162,700   |
| Angelica Ramos | Chief Executive Officer (CEO) | London        | 47  | 2009/10/09 | \$1,200,000 |
| Ashton Cox     | Junior Technical Author       | San Francisco | 66  | 2009/01/12 | \$86,000    |
| Bradley Greer  | Software Engineer             | London        | 41  | 2012/10/13 | \$132,000   |

Padding dan spacing yang cukup menjadikan tabel ini mudah dibaca. Baris header dan konten menggunakan warna yang agak berbeda agar mudah dibedakan

## 9. Bantu pengguna mengenali, mendiagnosis, dan memulihkan dari kesalahan

### Or start a new account

Choose a username (no spaces)

bert

⚠ bert is already taken. Please choose a different username.

Choose a password

\*\*\*

⚠ Passwords must be at least 6 characters and can only contain letters and numbers.

Retype password

Email address (must be real!)

not an email

⚠ The email provided does not appear to be valid

☒ Send me occasional Digg updates.

Memberikan umpan balik langsung dengan instruksi spesifik



Menggunakan gambar yang lucu, tetapi memberikan alternatif yang layak (daftar artikel dan tautan blog) dan tindakan (laporkan)

## 10. Bantuan dan Dokumentasi

- Meskipun lebih baik jika sistem dapat digunakan tanpa dokumentasi, mungkin perlu memberikan bantuan dan dokumentasi.
- Setiap informasi seperti itu harus mudah dicari, difokuskan pada aktifitas pengguna, daftar langkah konkret yang harus dilakukan, dan tidak terlalu besar.



Bantuan kontekstual (ini adalah contoh bantuan dalam modul 'Kolase') kiat di Picnik jelas dan mudah dinavigasi

[Tooltips](#) can be attached to any element. When you hover the element with your mouse, the title attribute is displayed in a little box next to the element, just like a native tooltip.

But as it's not a native tooltip, it can be styled. Any themes built with [ThemeRoller](#) will also style tooltips accordingly.

Toolt  
addi ThemeRoller: jQuery UI's theme builder / some  
application  
Your age:

Hover the field to see the tooltip.

Bantuan "*tooltip*" dapat digunakan untuk membuat bantuan sederhana untuk elemen tertentu.

## 2. Menggunakan ukuran performansi dalam menuliskan kode sumber

## 3. Menggunakan tipe data dan control program

### Variabel

- Suatu lokasi penyimpanan (di dalam memori) yang berisikan nilai atau informasi
- Kode program yang digunakan untuk menampung nilai tertentu
- Nilai dari variable dapat diisi dengan informasi yang diinginkan dan dapat dirubah nilainya pada saat kode program sedang berjalan.

**Aturan :**

- Diawali dengan karakter dolar ( \$ ) dan diikuti dengan nama pengenalan
- Nama pengenalan dimulai dengan huruf atau garis bawah (\_), tidak boleh diawali dengan angka.
- Tidak boleh mengandung spasi dan terdiri dari minimal satu karakter
- Bersifat case sensitive (huruf besar dan kecil dibedakan)
- Tidak diperlukan deklarasi type variable, tetapi type variable akan mengikuti type nilai yg diberikan.
- Setiap variabel terbentuk dalam tipe data variant (dapat menampung jenis data apapun).

**Contoh penulisan Variabel**

```
<?php
$i;
$nama;
$Umur;
$_lokasi_memori;
$ANGKA_MAKSIMUM;
?>
```

**Contoh penulisan yang salah**

```
<?php
$4ever; //variabel tidak
        boleh diawali dengan
        angka
$_salah satu; //varibel
        tidak boleh mengandung
        spasi
$nama*^; //variabel tidak
        boleh mengandung karakter
        khusus: * dan ^
?>
```

**Tipe Data**

- Variable merupakan 'tempat' dari data
- Data yang dinput kedalam variable memiliki tipe tertentu (angka, desimal, text)
- Tipe Data di PHP : integer, floating, string, Boolean, array dan object.

### **Tipe Data Integer**

- Tipe data berupa bilangan bulat
- Misalnya untuk menyimpan data jumlah stock, umur, tinggi badan, nomor sepatu.
- Dapat bernilai positif (+) maupun negative (-)
- Dapat dilakukan operasi matematis (tambah, kurang, kali, bagi).

### **Tipe Data Integer**

- Contoh

```
<?php
    $umur=21;
    $harga=15000;
    $rugi=-500000;

    echo $umur; //21
    echo "<br />";
    echo $harga; //15000
    echo "<br />";
    echo $rugi; //-500000
?>
```

Contoh dalam operasi matematik



```

<?php
    $a=14;
    $b=16;
    $c= $a + $b;
    echo $c; // 30

    $d=$a * $b;
    echo $d; // 224

```

### Type Data Float

- Tipe data floating point/real number berupa bilangan decimal (pecahan)
- Digunakan untuk variable yang akan berisi angka pecahan seperti nilai IPK, hasil pembagian, atau hasil komputasi numeric yang angkanya tidak bias ditampung oleh tipe data integer
- Contoh angka float: 0.9 , 2.80 , 3.14 , 0.314E1

Contoh:

```

<?php
    $angka_float1= 0.78;
    $angka_float2= 14.99;
    $angka_scientific1=0.314E1;
    $angka_scientific2=0.3365E-3
;

    echo $angka_float1; // 0.78
    echo "<br />";
    echo $angka_float2; //14.99
    echo "<br />";

    echo      $angka_scientific1;
//3.14
    echo "<br />";
        echo  $angka_scientific2;
//0.0003365
?>

```

- Contoh Dalam Operasi matematis

```
<?php
$a=10.66;
$b=12.4;
$c= $a + $b;
echo $c; // 23.06

$d=$a / $b;
echo $d; //
0.85967741935484
?>
```

### **Tipe Data String**

- Tipe data yang berisi text, kalimat, atau kumpulan karakter
- Penulisannya diapit oleh single quoted/petik satu (') atau double quoted/petik ganda (")
- Contoh:  
'a' , 'saya sedang belajar PHP' , 'emailku19@gmail.com'  
"a" , "saya sedang belajar PHP" , emailku19@gmail.com

```

<?php
$string1='Ini adalah string
sederhana';
$string2='Ini adalah string
yang bisa memiliki beberapa
baris';
$string3='Dia berkata: "I\'ll
be back"';
$string4="Dia berkata: \"I'll
be back\"";
$string5="Variabel akan
otomatis ditampilkan: $string1
dan $string3";

echo $string1; echo "<br>";
echo $string2; echo "<br>";
echo $string3; echo "<br>";

```

Contoh:

```

echo $string1; echo "<br>" ini adalah string sederhana
echo $string2; echo "<br>" ini adalah string yang bisa memiliki beberapa baris
echo $string3; echo "<br>" dia berkata: "I'll be back"
echo $string4; echo "<br>" Dia berkata: "I'll be back"
echo $string5; echo "<br>" Variabel akan otomatis ditampilkan: ini adalah string sederhana dan dia berkata: "I'll be back"

```

## Type data Boolean

- hanya memiliki 2 nilai : true dan false
- Digunakan dalam operasi logika

Contoh:

```

<?php
    $benar=true;
    $salah=false;

    echo "benar = $benar, salah =
    $salah";
    // hasil output: benar = 1,
    salah =
    $x = FALSE; // false
    $x = ""; // string kosong
    dianggap false
    $x = " "; // string dengan
    karakter adalah true;
    $x = 0; // false
    $x = 1; // true
?>

```

**B. Ketrampilan yang diperlukan dalam menulis kode dengan prinsip sesuai guidelines dan best practices**

1. Menerapkan coding- guidelines dan best practices dalam penulisan program (kode sumber)
2. Menggunakan ukuran performansi dalam menuliskan kode sumber
3. Menggunakan tipe data dan control program

**C. Sikap yang diperlukan dalam menulis kode dengan prinsip sesuai guidelines dan best practices**

1. Disiplin
2. Teliti
3. Objektif
4. Bertanggung jawab

## 1. Kuis 9 Web Developer

### Link Referensi Modul Pertama

1. Video Pembelajaran
2. E-book
3. Link Youtube/Website rujukan

### Link Pertanyaan Modul Pertama

<https://app.sli.do/> (bisa menggunakan aplikasi ini)

### Bahan Tayang

Bisa berupa Link/ Screen Capture Slide pelatihan

### Link room Pelatihan dan Jadwal live sesi bersama instruktur

Zoom, Blue Jeans, Meets

### Penilaian

Komposisi penilaian Kuis 9 Web Developer: Nilai 10 (Range 0 -10)

### Target Penyelesaian Modul Pertama

1hari/sampai 6JP



DIGITAL  
TALENT  
SCHOLARSHIP



[digitalent.kominfo.go.id](https://digitalent.kominfo.go.id)

 [DTS\\_kominfo](#)

 [digitalent.kominfo](#)

 [digitalent.kominfo](#)

   [Digital Talent Scholarship](#)



Badan Penelitian dan Pengembangan Sumber Daya Manusia  
Kementerian Komunikasi dan Informatika  
Republik Indonesia