

ToK Diklat  
Pengembangan Aplikasi Berbasis Web  
1 – 12 Juli 2019

YHT  
Juli 2019

# Outline

- Konsep Dasar Jaringan Komputer
- Konsep Dasar Web
- Pengenalan HTML
- Pengenalan PHP
- Pengenalan MySQL
- Pengenalan Yii Framework

# Konsep Dasar Jaringan Komputer

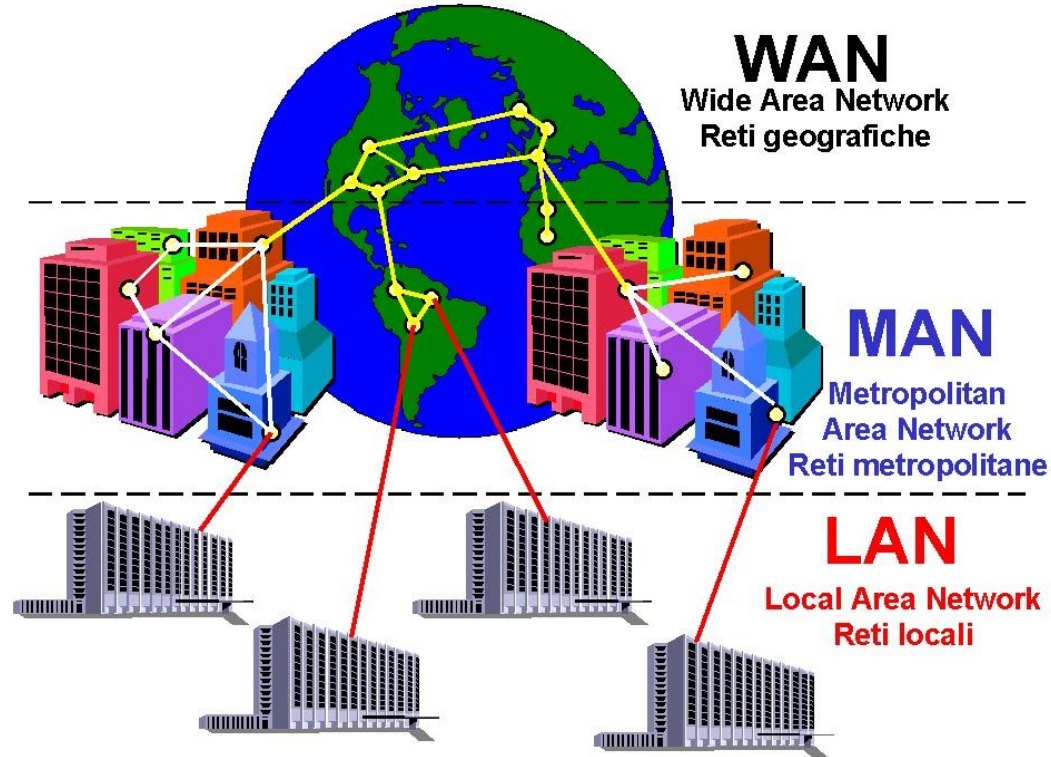
# Konsep Dasar Jaringan #1

Jaringan komputer adalah sekelompok komputer otonom yang saling berhubungan antara satu dengan yang lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program-program, dan menggunakan bersama perangkat keras.

# Contoh

- Sharing resource (data, printer, dll)
- Efisiensi waktu
- Efisiensi tempat
- Efisiensi dana

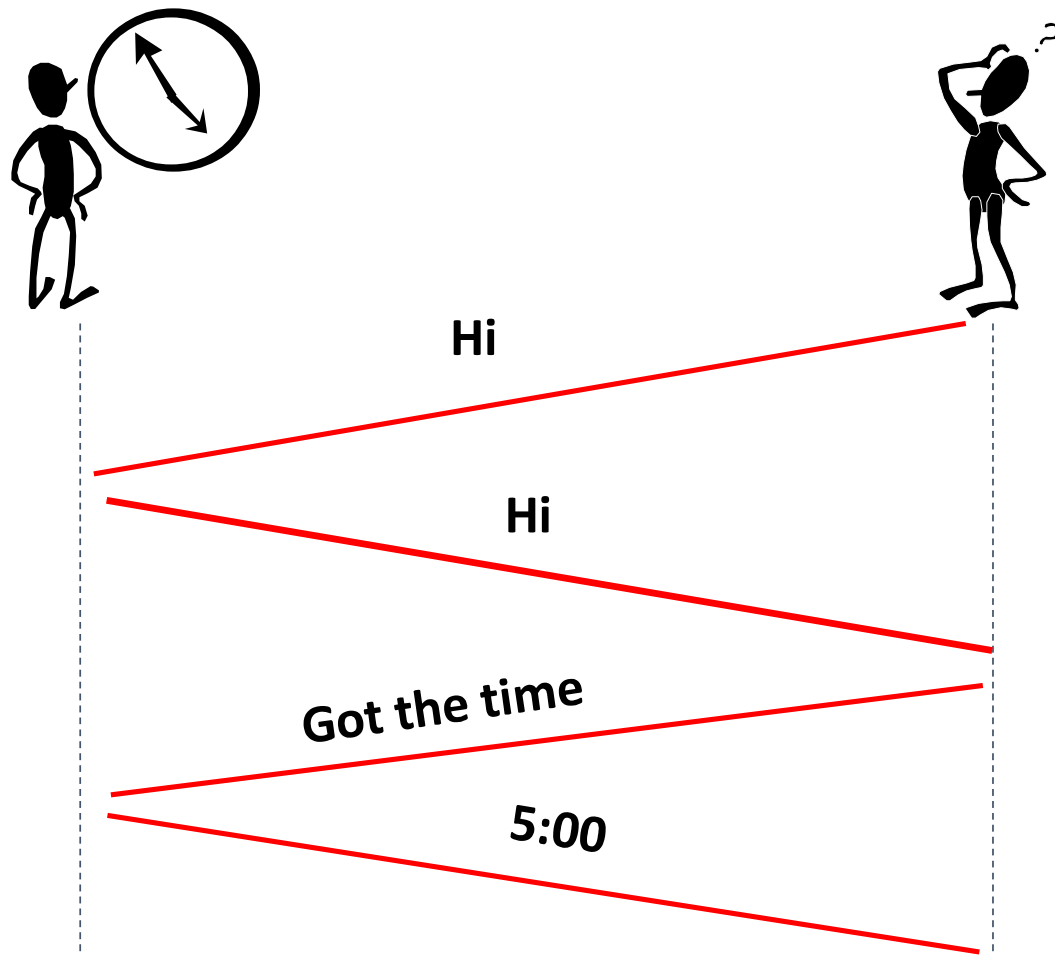
# Tipe Jaringan



# Protokol

- Protokol adalah sebuah aturan yang mendefinisikan beberapa fungsi yang ada dalam sebuah jaringan komputer, misalnya mengirim pesan, data, informasi dan fungsi lain yang harus dipenuhi oleh si pengirim (transmitter) dan si penerima (reciever) agar komunikasi dapat berlangsung dengan benar.

# Protokol: Analogi





# Konsep Dasar Web

# Aplikasi Berbasis Web

---

- Berarsitektur client-server
  - software web browser di sisi client
  - software web server di sisi server
- Menggunakan protokol HTTP dalam komunikasi antara client dan server
- Mempunyai fungsi untuk mengambil/menjalankan isi file dokumen web di server dan menampilkannya di sisi client

# Aplikasi Berbasis Web

---

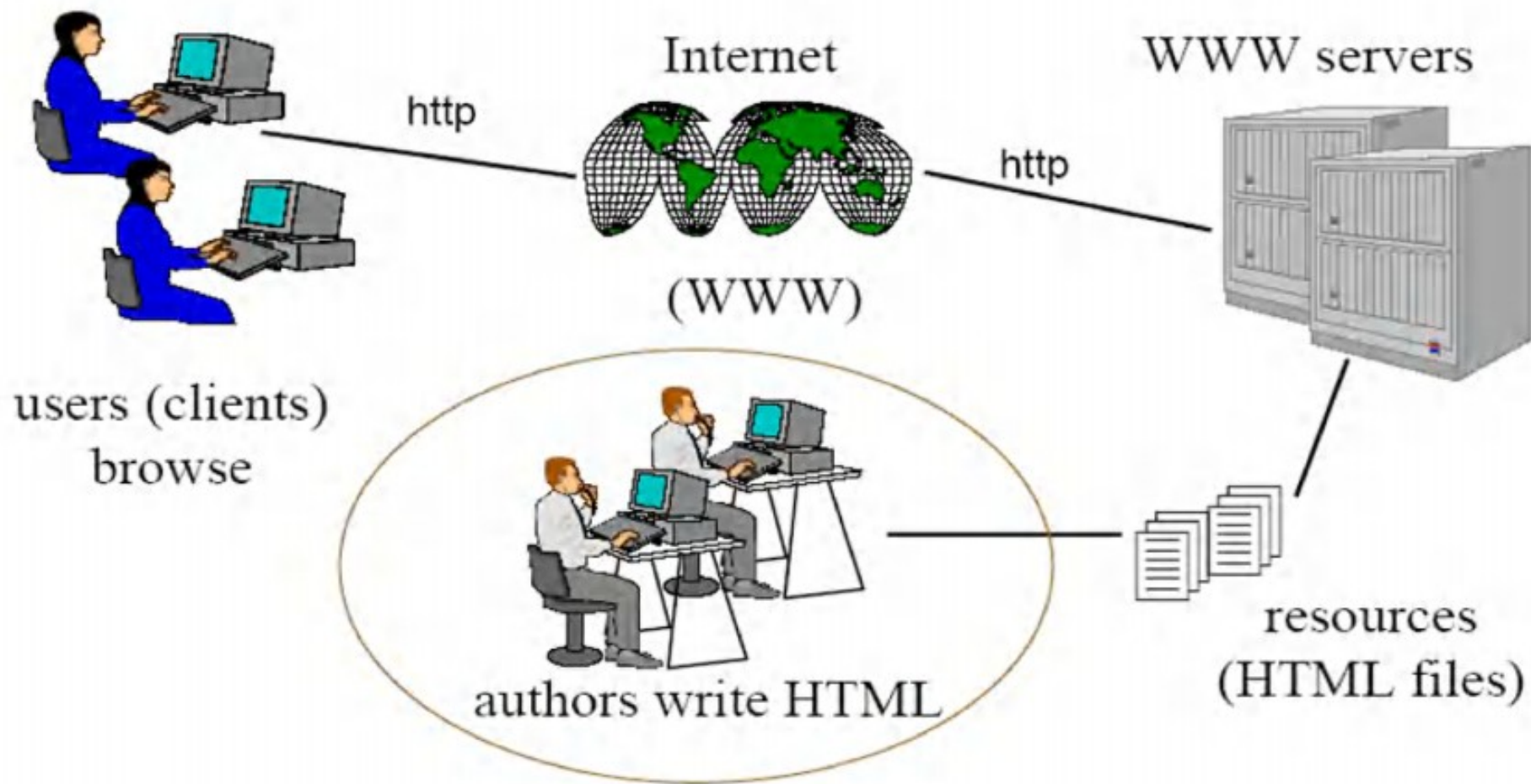
## Keunggulan

- Dapat diakses kapan pun dan dari mana pun selama ada internet
- Dapat diakses hanya dengan menggunakan web browser (umumnya sudah tersedia di PC, PDA, dan handphone terbaru),
- tidak perlu menginstall aplikasi client khusus

## Kekurangan

- Antarmuka yang dapat dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen web
- keterbatasan kemampuan web browser untuk menampilkannya
- Terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat

# Cara kerja Web





Database Layer

Application Layer



Database Connectivity



Server  
Operating  
System



HTTP  
Services





# Web Statis vs. Web Dinamis

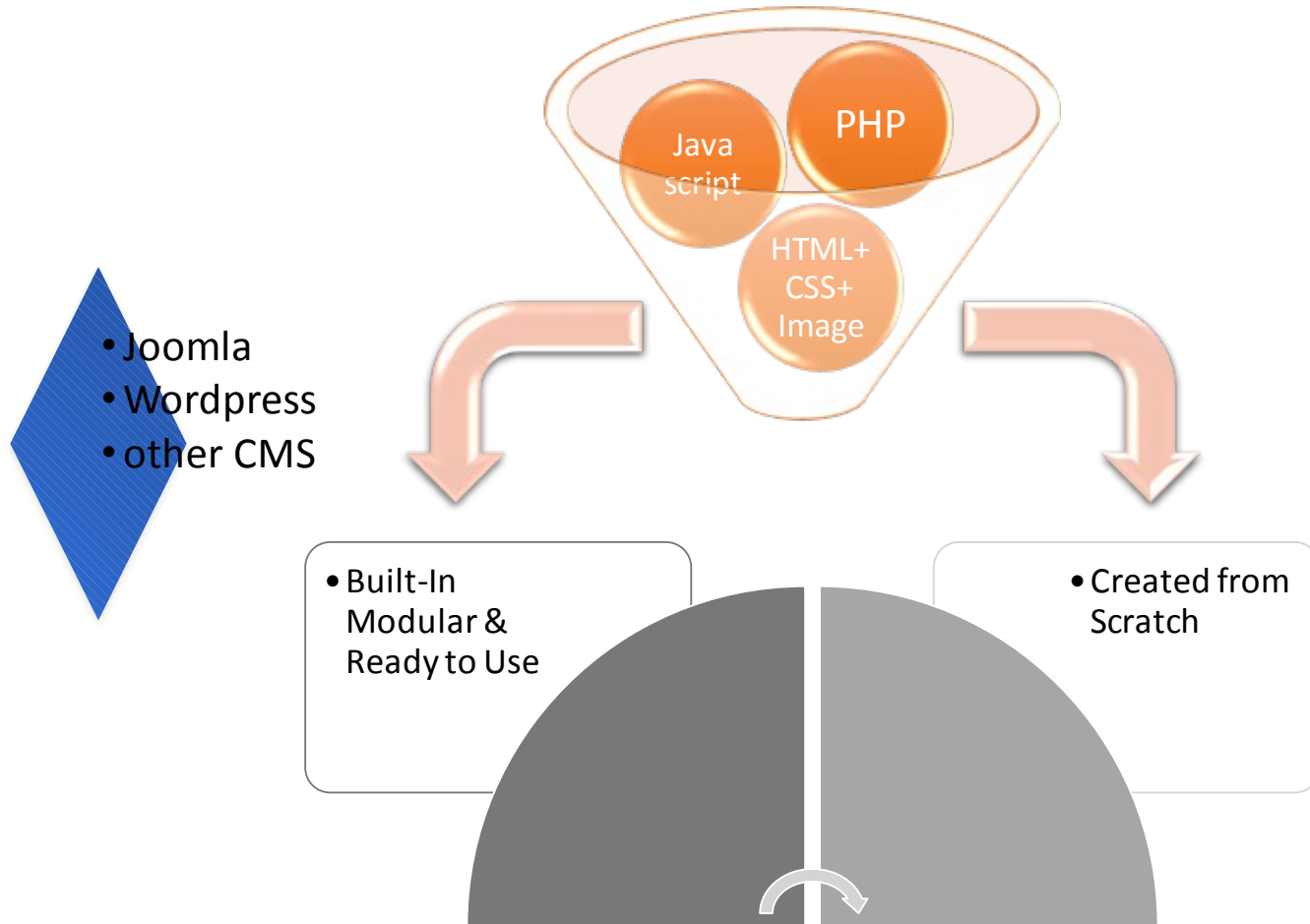
---

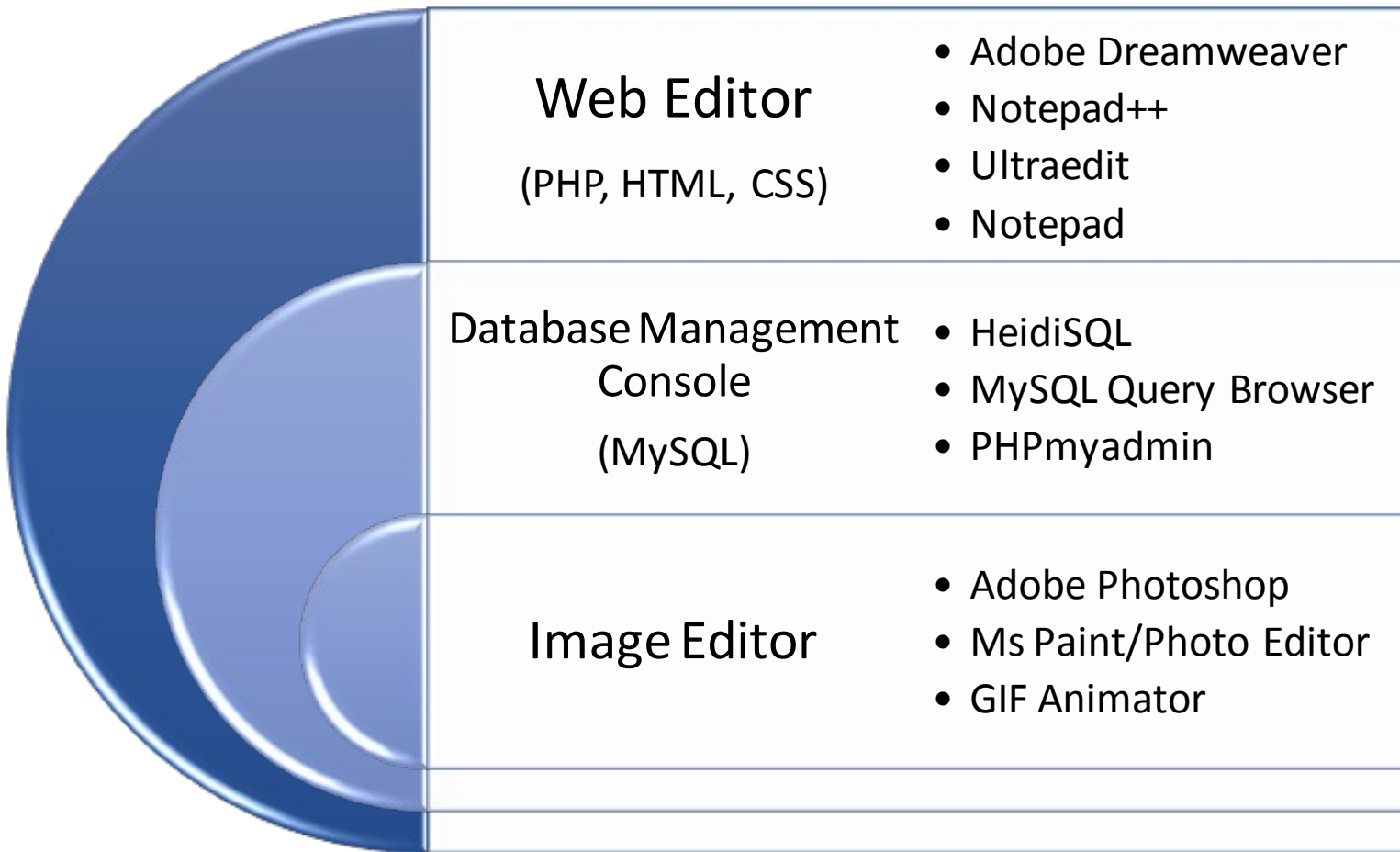
## Web Statis

- Sebagian besar halaman web statis
- Isi (teks / link / gambar) yang sama setiap kali diakses
- HyperText Markup Language (HTML) digunakan untuk menentukan teks / format gambar
- contoh: dokumen online, kebanyakan homepage

## Web Dinamis

- Sebagai Web dinamis mengarah ke layanan online/ e-commerce
- halaman Webnya juga harus menyediakan konten dinamis
- halaman harus update, berubah-ubah (misalnya: berputar banner, artikel ganti)
- harus mampu bereaksi terhadap tindakan info pengguna, permintaan dan proses, pemesanan jasa, dll
- Contoh: [www.amazon.com](http://www.amazon.com), [www.thehungersite.com](http://www.thehungersite.com)





Web  
Server  
(Apache,  
XAMPP,  
IIS)

PHP &  
MySQL  
Engine



# Pengenalan HTML

# Contoh

(kode/01\_basic.html)

## HTML Basic

```
<html>

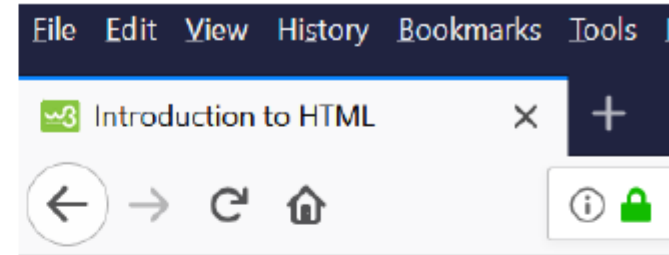
<head>
<title>Introduction to HTML</title>
</head>

<body>
The content of the document.....

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>

<p><b>Tip:</b> This is a paragraph.</p>
</body>

</html>
```

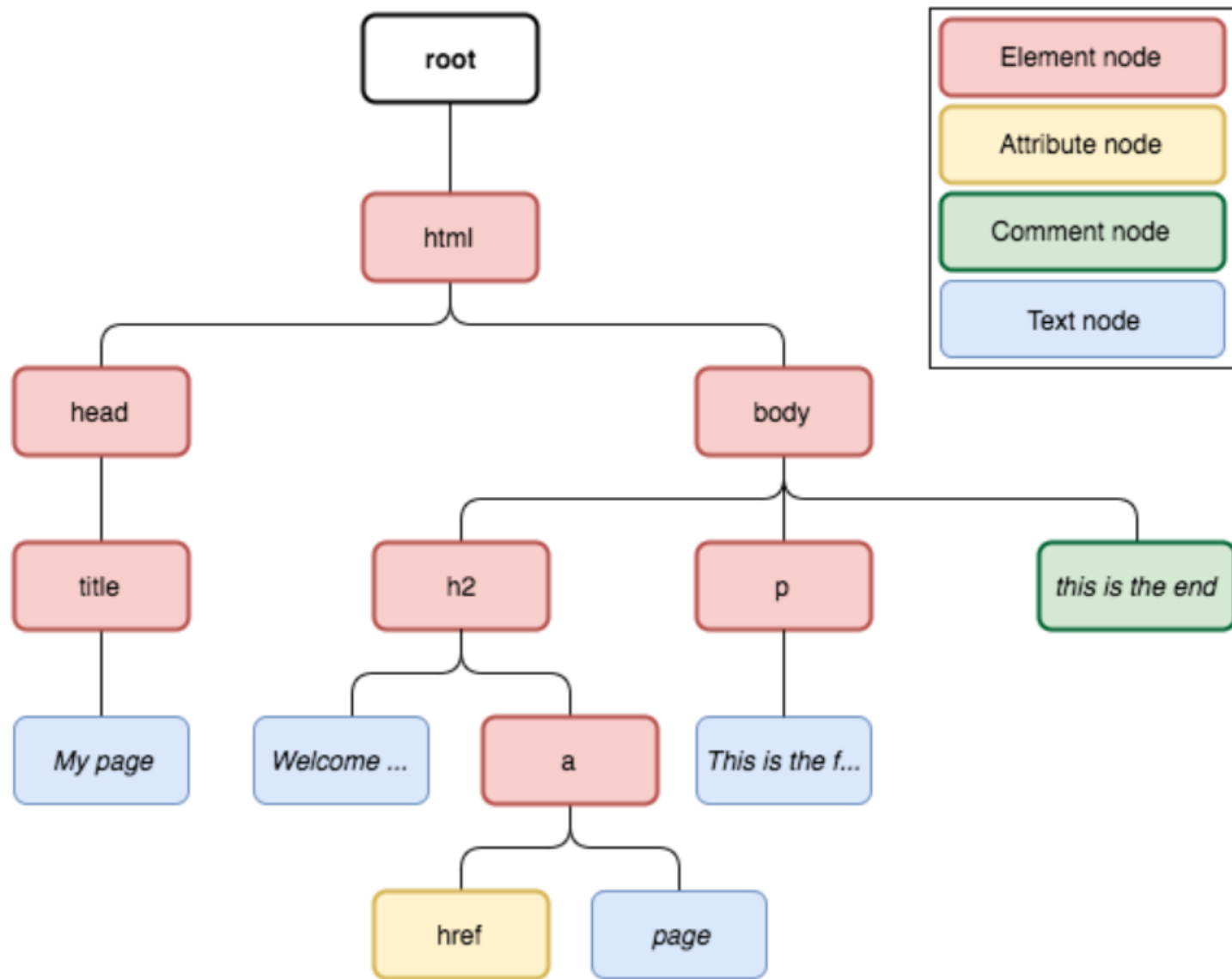


The content of the document.....

# This is heading 1

## This is heading 2

**Tip:** This is a paragraph.



# Tag lain

- Tabel (<table> ... </table>)
- Daftar (<ul> ... </ul>; <ol> ... </ol>)
- Tautan (<a href="http://url.to"> ...</a>)
- Blok (<div> ... </div>)
- Span (<span> ... </span>)

# Instalasi XAMPP

Folder **Kode/** silakan disalin ke **c:\xampp\htdocs**



# Pengenalan PHP

# PHP: Pengenalan

## **Defnisi**

PHP: Hypertext Preprocessor – dibuat oleh Rasmus Lerdorf (1995), merupakan bahasa pemrograman web server side yang bersifat open source (dikembangkan secara terbuka oleh berbagai programmer di dunia web), terintegrasi dengan HTML dan dapat menggunakan seluruh fasilitasnya, bersifat dinamis, dan seluruh perintah dieksekusi pada web server.

## **Mengapa PHP?**

- Sederhana
- Mudah digunakan dan Resource Referensi yang besar
- Open source, dikembangkan oleh banyak pihak
- Webserver friendly dan Database friendly
- Java collaboratøn

# Penulisan

<?php

...

[general]

\$variable = \$[variable/constant/function] [operator] ; (end with semicolon)

...

[bracketing]

[reserve words] { [performed operation] }

...

?>



# Variabel

## Variabel Tradisional

## Objek

### Numeric (all-signed)

- Integer (bilangan bulat): 0,1,2,3,...,12323,.. 2147483647(max)
- Floating point (bilangan real): 0.000003, 1543.234638, 786e23,... 1.8e308(max)

### AlphaNumeric/String/Text

- \$text = "Hello PHP, this is string type data"

### Array

- \$arr[1] = 9;
- \$arr = array("foo" => "bar", 12 => true);

```
<?php
class foo
{
    function do_foo()
    {
        echo "Doing foo.";
    }
}
```

```
$bar = new foo;
$bar->do_foo();
?>
```

# Pencabangan

## Pencabangan Terstruktur

```
switch ($i) {  
  case 0:  
    echo "i equals 0";  
    break;  
  case 1:  
    echo "i equals 1";  
    break;  
  case 2:  
    echo "i equals 2";  
    break;  
}
```

## Pencabangan Tidak Terstruktur

```
if ($a > $b) {  
  echo "a is bigger than b";  
} elseif ($a == $b) {  
  echo "a is equal to b";  
} else {  
  echo "a is smaller than b";  
}
```

# Pengulangan

## Pengulangan Definitif

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}  
  
$arr = array("one", "two", "three");  
reset($arr);  
  
foreach ($arr as $key => $value) {  
    echo "Key: $key; Value: $value<br />\n";  
}  
  
while (list($key, $value) = each($arr)) {  
    echo "Key: $key; Value: $value<br />\n";  
}
```

## Pengulangan Tidak Berhingga

```
$i = 0;  
do {  
    echo $i;  
} while ($i > 0);  
  
$i = 1;  
while ($i <= 10) {  
    echo $i++; /* the printed value would be  
               $i before the increment  
               (post-increment) */  
}
```

# Fungsi

## Fungsi Dasar (Built-In)

Daftar fungsi dasar PHP secara keseluruhan dapat dilihat pada halaman web berikut ini

<http://php.net/quickref.php>

## User Defined Function

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Example function. \n";
    return $retval;
}

/** calling a function **/
$a = foo($arg_1, $arg_2, /* ..., */ $arg_n)

?>
```

# PHP & HTML

(Kode/02\_html.php)

```
<html>
  <head>
    <title>Example #1 TDavid's Very First PHP Script ever!</title>
  </head>
  <body>
    <?php
      echo "Hari ini adalah " . Date("1 F d, Y");
    ?>
  </body>
</html>
```



# Pengenalan MySQL

# MySQL

MySQL adalah sistem manajemen basis data (RDBMS) yang menggunakan bahasa SQL, bersifat open source dan gratis dibawah lisensi GNU GPL, dikembangkan oleh MySQL AB dan mendukung konektivitas ke berbagai aplikasi, baik web maupun client application.

## **Mengapa MySQL?**

- Stabil dan dapat digunakan pada hampir semua operating system
- Mudah digunakan dan Resource Referensi yang besar
- Open source, dikembangkan oleh banyak pihak dan dukungan komunitas
- Kecepatan yang cukup baik untuk kategori open source DBMS
- Level Sekuriti/Keamanan Data yang cukup baik

# SQL

## SQL Statement

Pernyataan query yang terstruktur, yang digunakan untuk memanipulasi sekumpulan data

## Jenis SQL Statement dan Penjelasannya

- Data Definition
  - [CREATE/ALTER/DROP] [database/table]
- Data Manipulation
  - [SELECT/INSERT/UPDATE/DELETE/TRUNCATE/...] [datarow] [FROM] [database/table]
  - SubQuery [SELECT IN SELECT]
  - TRANSACTIONAL COMMAND
- Engine Specific Command
  - PIVOT, LIMIT

A SQL statement takes the general form:

```
SELECT field_1
FROM table_1
WHERE criterion_1
;
```



# SQL: SELECT

SQL clause	What it does	Required
SELECT	Lists the fields that contain data of interest.	Yes
FROM	Lists the tables that contain the fields listed in the SELECT clause.	Yes
WHERE	Specifies field criteria that must be met by each record to be included in the results.	No
ORDER BY	Specifies how to sort the results.	No
GROUP BY	In a SQL statement that contains aggregate functions, lists fields that are not summarized in the SELECT clause.	Only if there are such fields
HAVING	In a SQL statement that contains aggregate functions, specifies conditions that apply to fields that are summarized in the SELECT statement.	No

# MySQL: Client

- PHPMyAdmin
- Toad for MySQL

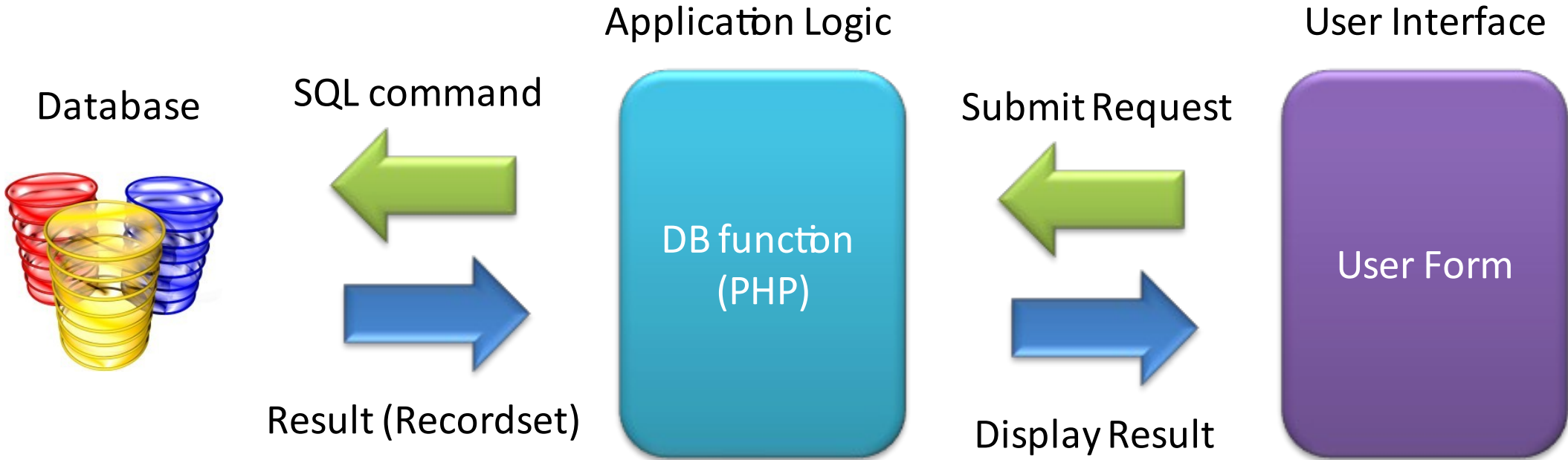
# DB Designer

- Power Designer

# PHPMyAdmin

- Buka: <http://localhost/phpmyadmin>
- Masukkan login

# PHP → MySQL



# Contoh: Fungsi Koneksi

(Kode/03\_mysql.php [3:20])

```
function connectdb() {  
    $host = "localhost";  
    $user = "root";  
    $pass = "12345";  
    $dbse = "webdb";  
    break;  
  
    $link = mysql_connect($host, $user, $pass);  
    if ($link) {  
        if (mysql_select_db($dbse)) {  
            return $link;  
        } else {  
            die ("Cannot select database!");  
        }  
    } else {  
        die ("Cannot connect to database service!");  
    }  
}
```

# Contoh: Fungsi *Query*

(Kode/03\_mysql.php [22:33])

```
function runsqltext($sSQL)
{
    $a = connectdb();
    if ($a) {
        $result = mysql_query($sSQL);
        if ($result) {
            return $result;
        } else {
            die ("Cannot execute Query!<br><strong>Query Line</strong> :  

                <font color='red'>\".$sSQL.\"</font>");
        }
    }
    mysql_close($a);
}
```

# Contoh: *Run Query*

(Kode/03\_mysql.php [35:36])

```
$sql = "SELECT a.TradeDate, a.FundId, b.DescFund, SUM(a.Unit) AS Units,  
          SUM(a.CapitalValue) AS NAV, SUM(a.CapitalValue)/SUM(Unit) AS NAVPERUNIT  
FROM mi_holding a INNER JOIN ufund b ON a.FundId = b.FundId  
WHERE a.tradedate = '$cDate' AND a.FundID = '$id'  
GROUP BY a.TradeDate, a.FundId, b.DescFund  
ORDER BY SecType, SecCode";  
  
$res = runsqltext($sql);
```



# Contoh: Tampilan

## (Kode/03\_mysql.php [38:43])

```
echo "<table border=0 width=100%>";
while ($row = mysql_fetch_array($res)) {
    echo "<tr>
        <td width=15%>Fund Code</td>
        <td width=35%>$row[1]</td>
        <td align='right' width=30%>NAV</td>
        <td align='right' width=20%>".number_format($row[4],2,".",",")."</td>
    <tr>";
    echo "<tr>
        <td>Descriptions</td>
        <td>$row[2]</td>
        <td align='right'>NAV per Unit</td>
        <td align='right'>".number_format($row[5],2,".",",")."</td>
    <tr>";
    echo "<tr>
        <td>Date</td>
        <td>".date("d-M-Y",strtotime($row[0]))."</td>
        <td align='right'>Unit</td>
        <td align='right'>".number_format($row[3],2,".",",")."</td>
    <tr>";
}
echo "</table>";
```

# Pengenalan Yii Framework

# Yii Framework

Merupakan salah satu framework (kerangka kerja) dalam pemrograman web dengan PHP yang menggunakan konsep MVC.

# Konsep OOP

- Class adalah *blueprint* dari suatu **object**.
- Class digunakan hanya untuk membuat kerangka dasar.
- Pada bahasa pemrograman, class merupakan sekelompok kode yang dituliskan untuk mendefinisikan properti dan metod yang ada pada sebuah object.

# Contoh: OOP

(Kode/04\_OOP.php [2:6])

file\_class.php

```
1  <?php
2  class Bdkmalang
3  {
4      function tampilHallo() //metode
5      {
6          echo "Hallo $this->nama";
7      }
8  }
9  ?>
```

# Contoh: OOP

(Kode/04\_OOP.php [8:12])

file\_object.php

```
1  <?php
2  require_once("file_class.php"); //file yang memuat class
3  $obj = new Bdkmalang(); //proses instansiasi object
4  $obj->nama = "Supriyono"."<br>"; //mendefinisikan nilai properti
5  echo $obj->nama; // memanggil properti
6  $obj->tampilHallo(); // menjalankan metode
7  ?>
```

# Constructor / Destruktor

(Kode/05\_OOP\_Init.php)

```
1 <?php
2 // buat class Bdkmalang
3 class Bdkmalang {
4
5     private $instruktur = "Tim IT Bdk Malang";
6     private $nohp = "081333289554";
7
8     public function __construct(){
9         echo "Ini berasal dari Constructor Bdkmalang";
10    }
11
12    public function jadwal_pelatihan(){
13        return "Jadwal Pelatihan $this->instruktur WA $this->nohp";
14    }
15
16    public function __destruct(){
17        echo "Ini berasal dari Destructor Bdkmalang";
18    }
19 }
20
21 // buat objek dari class Bdkmalang (instansiasi)
22 $jadwal_pelatihan= new Bdkmalang();
23
24 echo "<br />";
25 echo $jadwal_pelatihan->jadwal_pelatihan();
26 echo "<br />";
27 ?>
```

# Encapsulasi

- Encapsulation adalah mekanisme “membungkus” sebuah data pada sebuah object.
- Dalam istilah lain yang seringkali disebutkan adalah “Information Hiding”.
- Pada PHP terdapat 3 modifier yang dapat diimplementasikan untuk melakukan “pembungkusan” data yaitu **private**, **protected** dan **public**.
- Modifier tersebut digunakan untuk mendefinisikan tingkat visibilitas sebuah data (properti) atau fungsi (metode) yang ada di dalam class.



# Contoh Encapsulasi #1

(Kode/06\_OOP\_Enkapsulasi.php [1:12])

```
1 <?php
2 class Timbdk
3 {
4     private $_nama;
5     private $_umur;
6
7     function __construct($nama, $umur = 0)
8     {
9         if (!is_int($umur))
10        {
11            throw new Exception("Parameter Nilai UMUR Bukan Integer");
12        }
13        $this->_umur = $umur;
14        $this->_nama = $nama;
15    }
```

# Contoh Encapsulasi #2

(Kode/06\_OOP\_Encapsulasi.php [13:32])

```
16
17 public function set_umur($umur)
18 {
19     if (!is_int($umur))
20     {
21         throw new Exception("Parameter Nilai UMUR Bukan Integer");
22     }
23     $this->_umur = $umur;
24 }
25
26 public function tahunPensiun()
27 {
28     return 70 - $this->_umur;
29 }
30 }
31
```

# Contoh Encapsulasi #3

(Kode/06\_OOP\_Encapsulasi.php [34:43])

```
32 $Timbdk = new Timbdk("Supriyono");  
33 $Timbdk->set_umur(30);  
34  
35 echo $Timbdk->tahunPensiun();  
36 ?>
```

# Pembahasan

- Properti `_nama` dan `_umur` mempunyai modifier `private`, dimana tidak akan bisa diakses langsung.
- Properti tersebut akan bisa diakses jika kita mendefinisikan sebuah metode untuk mengakses properti tersebut yang memiliki modifier `public`.

# Inheritance (Pewarisan)

(Kode/07 OOP Inheritans.php)

```
1 <?php
2 // buat class induk: pegawai
3 class pegawai {
4
5     public $jabatan;
6     public $golongan;
7     public $gaji;
8
9     public function daftar_pegawai() {
10         return "menjadi pegawai teladan";
11     }
12 }
```

```
14 // turunkan class pegawai ke staf_ahli
```

```
15 class staf_ahli extends pegawai {
```

```
17     public function lihat_detail() {
```

```
18         return "jabatan: $this->jabatan, golongan: $this->golongan, gaji: $this->gaji";
```

```
19     }
```

```
20 }
```

```
21
22 // buat objek dari class staf_ahli (instansiasi)
23 $staf_ahli_baru = new staf_ahli();
24
25 // isi property objek
26 $staf_ahli_baru->jabatan = "Direktur IT";
27 $staf_ahli_baru->golongan = "IV A";
28 $staf_ahli_baru->gaji = "20.000.000";
29
30 //panggil method objek
31 echo $staf_ahli_baru->daftar_pegawai();
32 echo "<br />";
33 echo $staf_ahli_baru->lihat_detail();
34 ?>
```

# Abstract Class

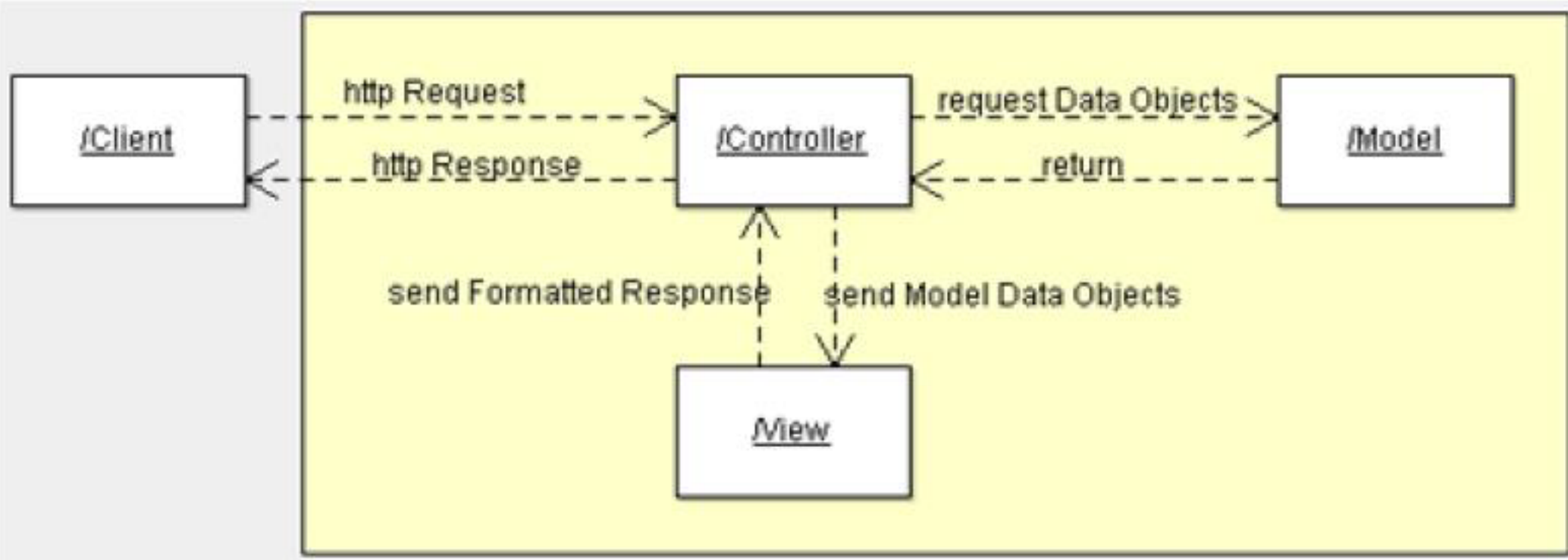
(Kode/08\_OOP\_Abstrak.php)

```
1 <?php
2 // buat abstract class
3 abstract class pegawai{
4     // buat abstract method
5     abstract public function lihat_detail_peg();
6 }
7
8 class gaji_peg extends pegawai{
9
10     // implementasi abstract method
11     public function lihat_detail_peg(){
12         return "Lihat gaji peg...";
13     }
14     // method 'biasa'
15     public function gaji_peg_detail(){
16         return "Detail gaji peg...";
17     }
18 }
19 // buat objek dari class gaji_peg
20 $gaji_peg_baru = new gaji_peg();
21 echo $gaji_peg_baru->lihat_detail_peg();
22 // Lihat Spec gaji_peg...
23
24 echo "<br />";
25 echo $gaji_peg_baru->gaji_peg_detail();
26 // Lihat gaji_peg...
27 ?>
```

# Abstrak Class dan Abstract Method

- **Abstract Class** adalah sebuah class yang tidak bisa di-*instansiasi* (tidak bisa dibuat menjadi objek) dan berperan sebagai '*kerangka dasar*' bagi class turunannya. Di dalam *abstract class* umumnya akan memiliki *abstract method*.
- **Abstract Method** adalah sebuah '*method dasar*' yang harus diimplementasikan ulang di dalam class anak (*child class*). *Abstract method* ditulis tanpa isi dari *method*, melainkan hanya '**signature**'-nya saja. **Signature** dari sebuah *method* adalah bagian method yang terdiri dari nama method dan parameternya (jika ada).

# Konsep MVC





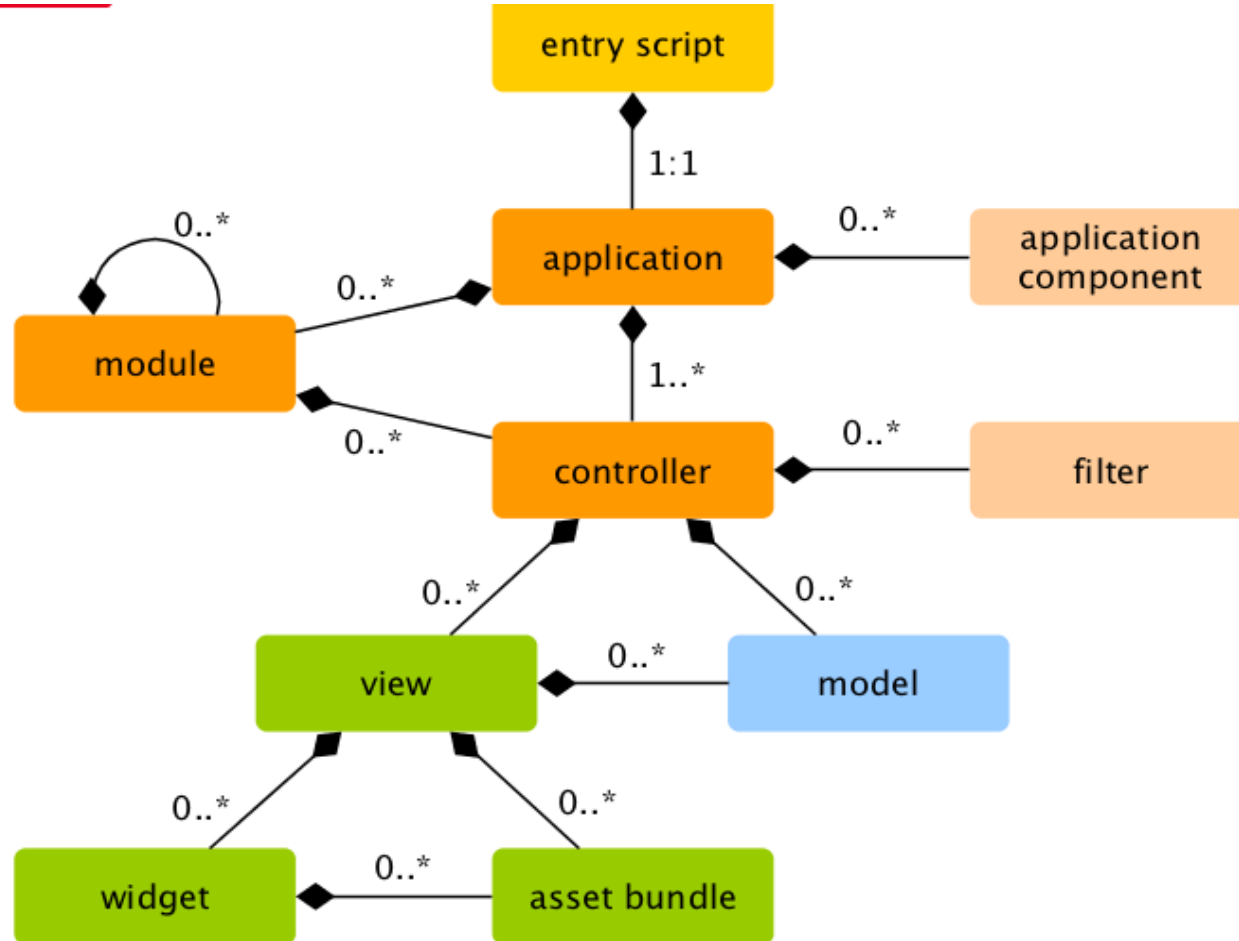
# Pemasangan Yii

- Download dari [yiiframework.com](http://yiiframework.com)
  - yii2-app-basic \*
  - yii2-app-advanced
- *Extract* berkas ke direktori yang dapat diakses via web.
  - C:\xampp\htdocs
- Test
  - <http://localhost/yii2/frontend/web>

# Folder

- config/
- controllers/
- models/
- views/

# MVC



# Yii: Hello

(Kode/09\_Yii\_Hello.txt)

- Edit app\controllers\SiteController.php
- Buat app\views\site\hello.php
- Buka:
  - <http://localhost/yii2/frontend/web/index.php?r=hello>

# CRUD: Gii

(Kode/10\_CRUD.txt)

- Pembuatan MVC
  - Create
  - Read
  - Update
  - Delete
- Modul
  - <http://localhost/yii2/frontend/web/index.php?r=gii>

# Contoh IDE Proyek

- DB Registrasi Kelas IHT
- DB Surat Masuk
- DB Disposisi
- DB Surat Keluar

# Terima Kasih

YHT  
Juli 2019