

## **MATERI MODULE PEMOGRAMAN BERORIENTASI OBJEK PHP**

|              |                                 |
|--------------|---------------------------------|
| Kelas        | : 12 RPL                        |
| Mapel        | : PEMOGRAMAN BERORIENTASI OBJEK |
| Disusun oleh | : Hasudungan Sitorus S, Kom     |

|                                |   |
|--------------------------------|---|
| <b>PROLOG</b>                  | •penjelasan singkat, penggunaan PBO dan kelebihan dalam penggunaan                    |
| <b>CLASS-OBJECT</b>            | •penjelasan singkat, penggunaan tentang class-objek dan disertai dengan base code     |
| <b>ACCESS MODIFIERS</b>        | •penjelasan singkat, penggunaan tentang static properti dan disertai dengan base code |
| <b>INHERITANCE / PEWARISAN</b> | •penjelasan singkat, penggunaan tentang static properti dan disertai dengan base code |
| <b>CONSTANTS</b>               | •penjelasan singkat, penggunaan tentang constant dan disertai dengan base code        |
| <b>ABSTRACT CLASS</b>          | •penjelasan singkat, penggunaan tentang abstract class dan disertai dengan base code  |
| <b>INTERFACE</b>               | •penjelasan singkat, penggunaan tentang interface dan disertai dengan base code       |
| <b>TRAIT</b>                   | •penjelasan singkat, penggunaan tentang trait dan disertai dengan base code           |
| <b>STATIC METHOD</b>           | •penjelasan singkat, penggunaan tentang static method dan disertai dengan base code   |
| <b>STATIC PROPERTHIES</b>      | •penjelasan singkat, penggunaan tentang static properti dan disertai dengan base code |

## **AUTOLOADING**

- penjelasan singkat, penggunaan tentang static properti dan disertai dengan base code

## **DEPEDENCY INJECTION**

- penjelasan singkat, penggunaan tentang static properti dan disertai dengan base code

## **STUDI KASUS PENGGUNAAN KONSEP PBO(CRUD)**

- penjelasan singkat, penggunaan tentang static properti dan disertai dengan base code

## **#PROLOG**

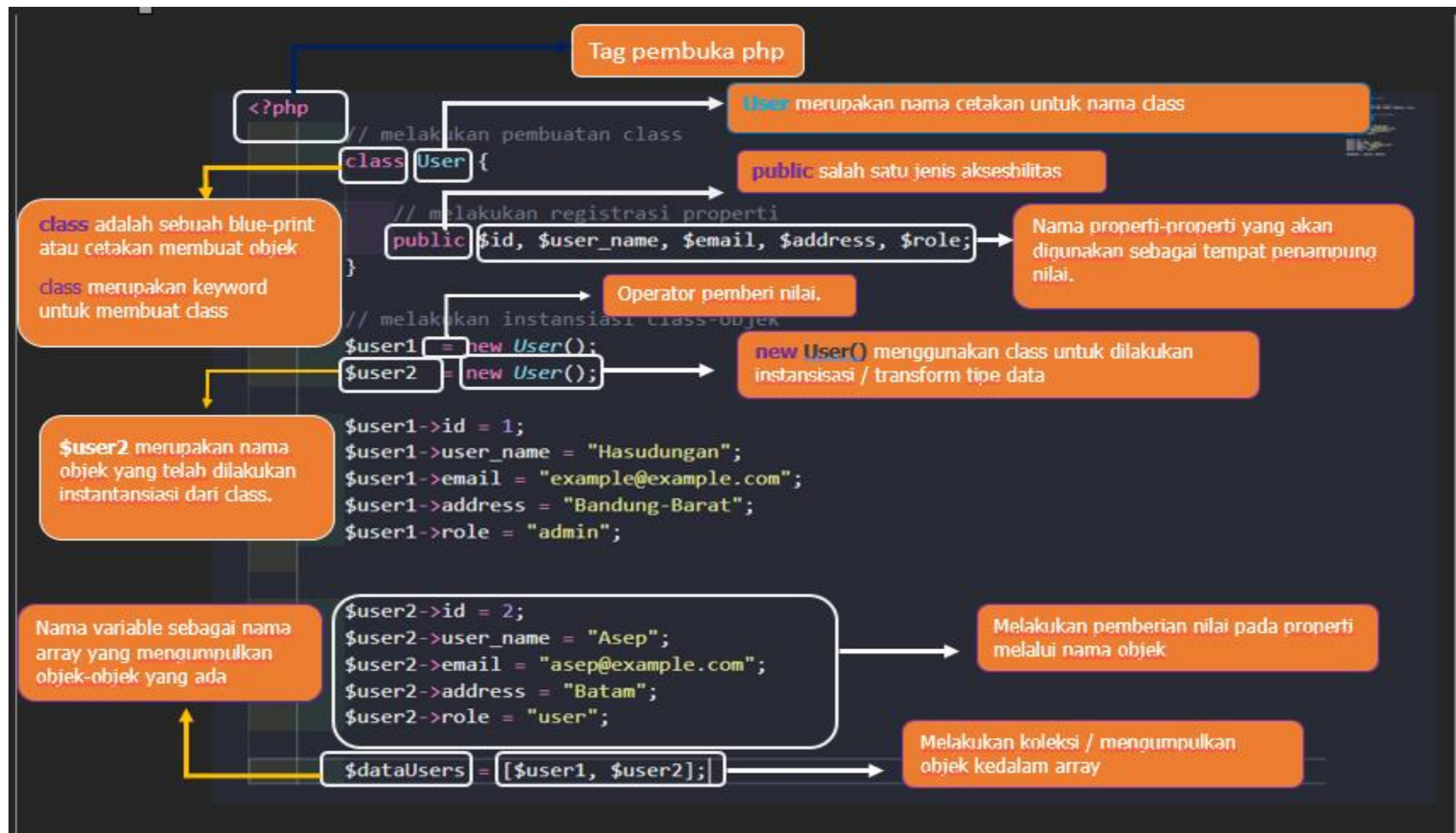
Pemrograman Berorientasi Objek atau Object Oriented Programming (OOP) adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Pemahaman ini mengacu pada teknis dari pemrograman OOP, yakni menyusun, mengolah objek-objek berisi data dan operasi yang dibutuhkan melalui Class dan Method yang dibutuhkan dalam suatu perangkat lunak. Referensi: <https://serupa.id/konsep-pemrograman-berorientasi-objek-pbo-oop/>

Dengan kata lain pemograman berorientasi objek merupakan cara berpikir programmer menyelesaikan / memecahkan masalah berorientasi dasar class,objek-objek serta attribute pendukung lainnya yang digunakan.

## **#CLASS-OBJEK**

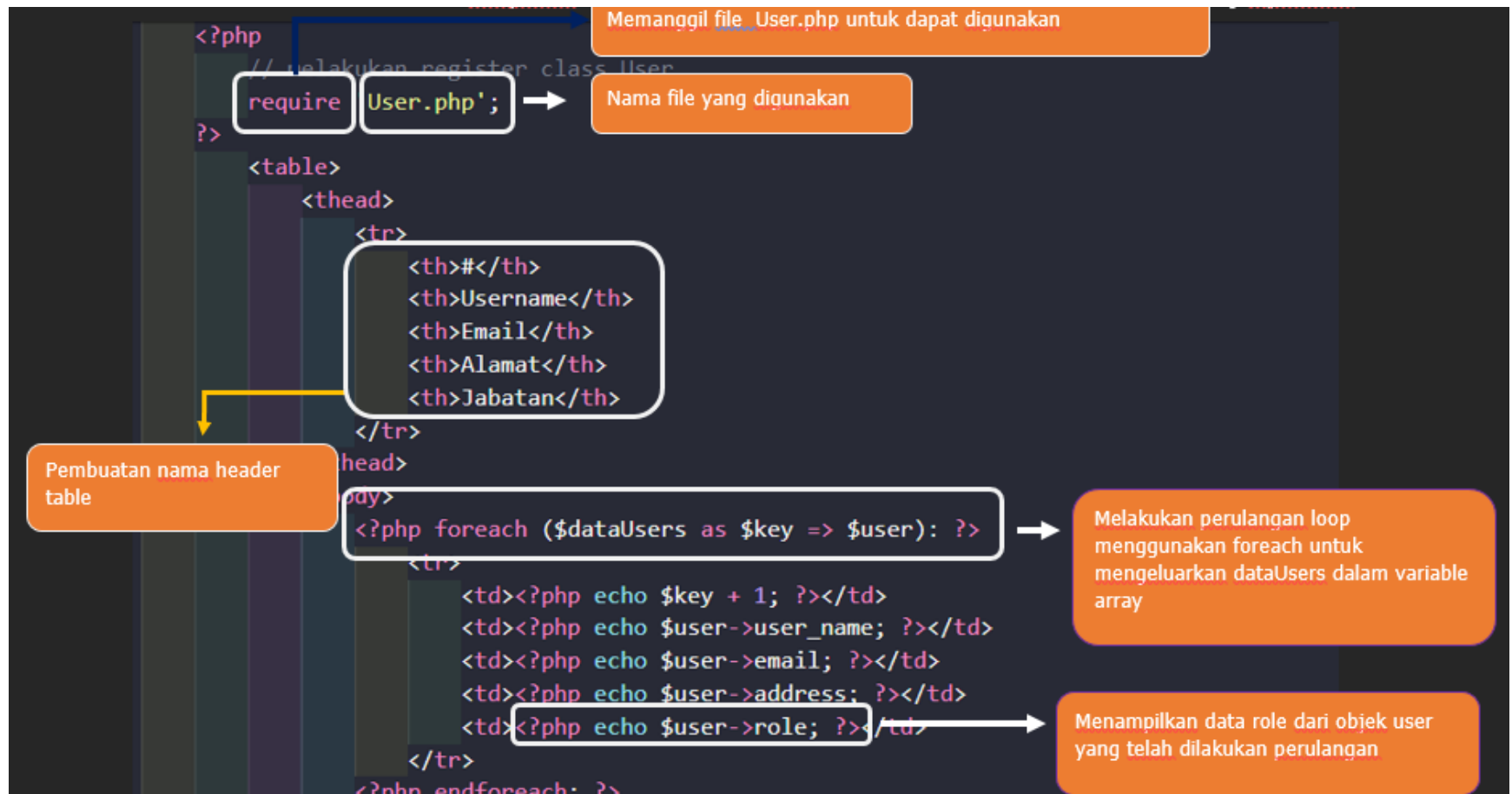
Class adalah template (pola) untuk berbagai obyek dengan fitur serupa. Class mewujudkan semua kumpulan/set fitur tertentu dari objek. Referensi: <https://staffnew.uny.ac.id/upload/132206816/pendidikan/bambangshm-pbo-matericlassobject.pdf#:~:text=Class%20adalah%20template%20%28pola%29%20untuk%20berbagai%20obyek%20dengan,Class%20mewujudkan%20semua%20kumpulan%2Fset%20fitur%20tertentu%20dari%20objek.>

Contoh base-code dan penggunaannya.

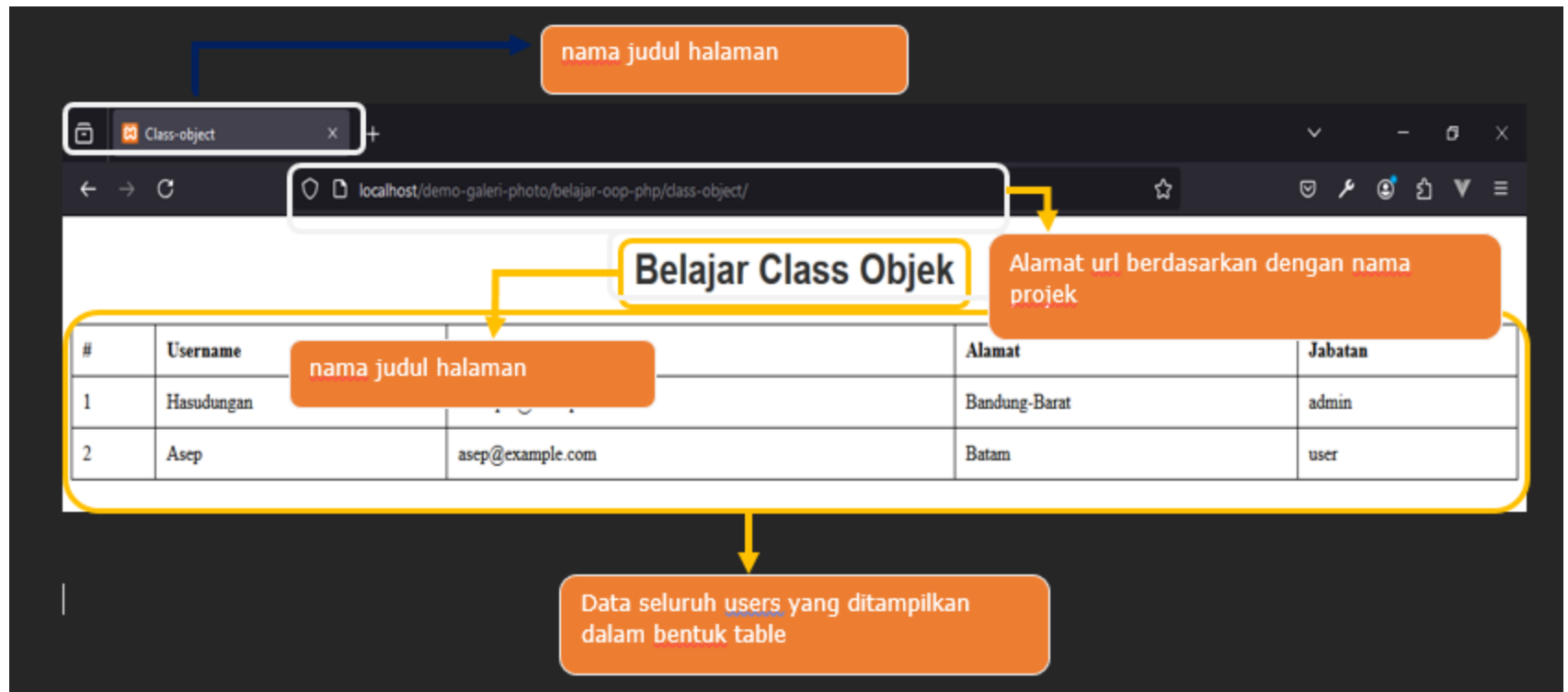


Gambar 1.1 class User.php

Dalam latihan dilakukan, dianjurkan untuk memisahkan antara class dengan front-end serta css yang digunakan. Berikut gambar dari base-code.



**Gambar 1.2** index.php



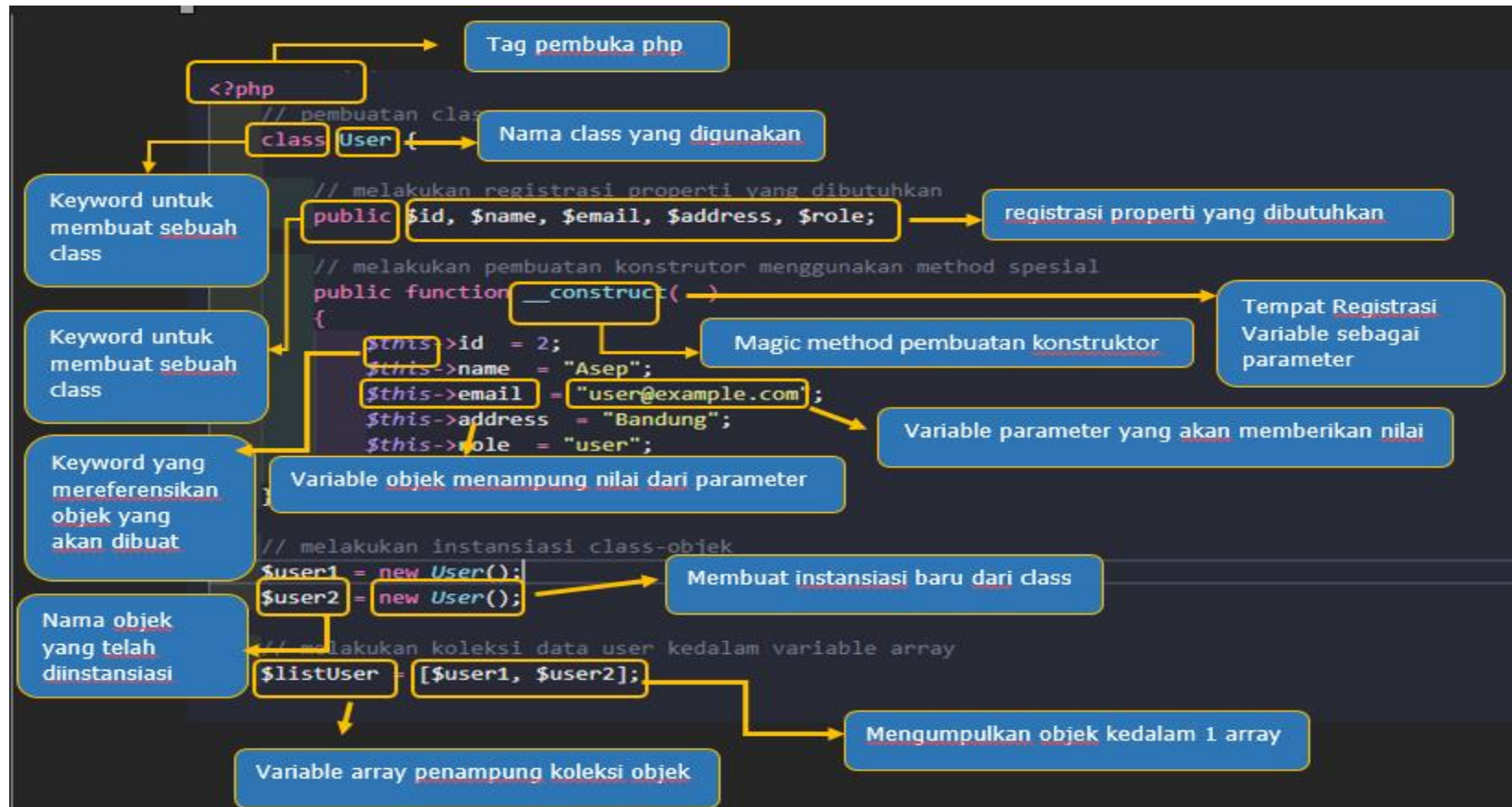
**Gambar1.3** Tampilan Halaman index.php

## **#Constructor**

Constructor adalah merupakan konsep pembuatan konstruksi yang memungkinkan melakukan inisialisasi sebuah property yang ketika dilakukan pembuatan objek. Dalam pembuatan Konstruktur yaitu dengan menggunakan fungsi

special dari php yang memudahkan inisialisasi yang diinginkan. Berikut potongan kode, penjelasan serta beberapa jenis construct yang dapat dibangun.

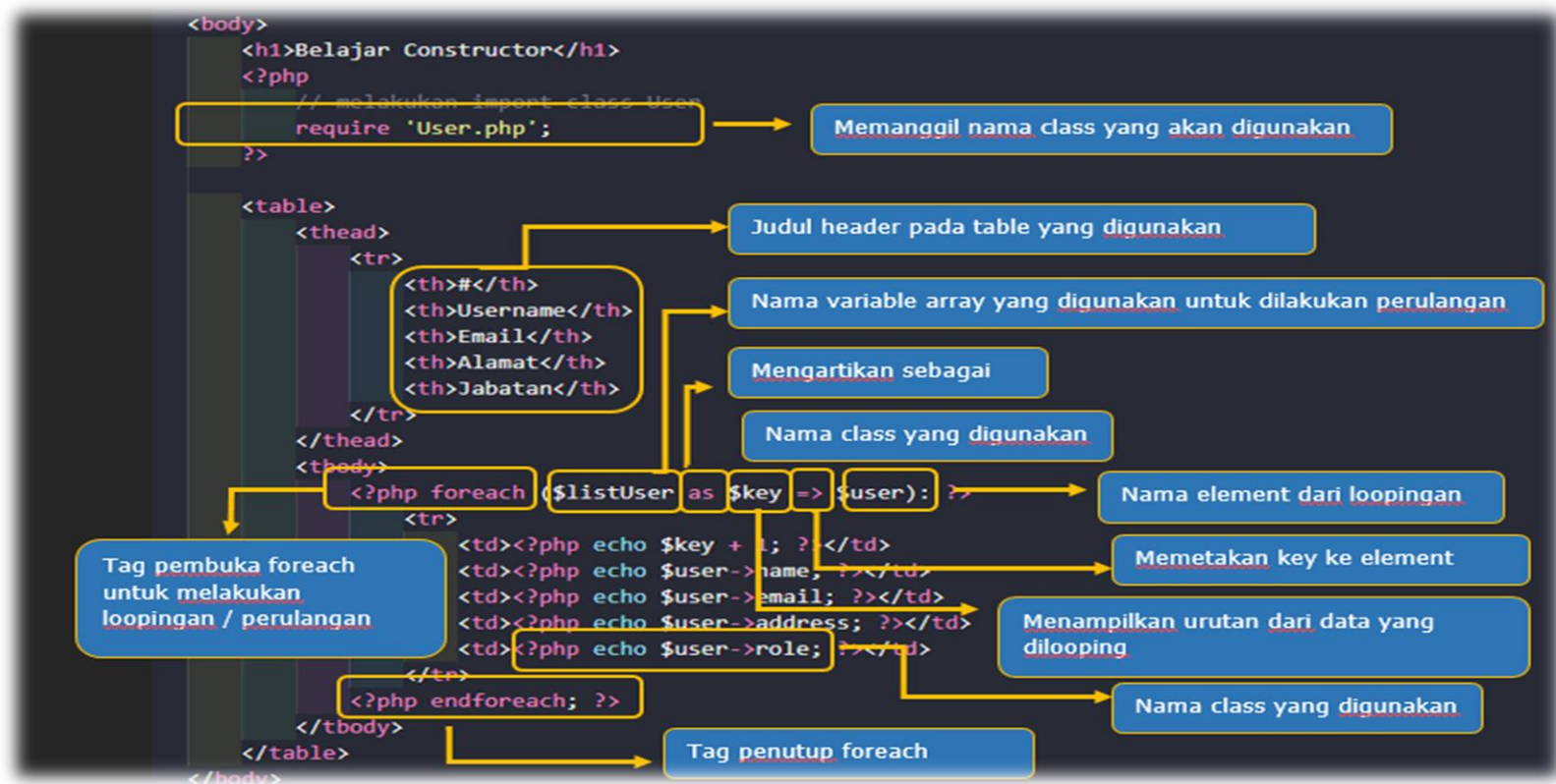
### #Constructor dengan nilai default tanpa memberi variable parameter



Gambar 2.1 Constructor dengan parameter



**Instansiasi** adalah proses pembuatan / transformasi dari class menjadi sebuah objek. Dalam contoh base-code diatas, kita melakukan konstruktor dengan nilai default namun tidak memiliki variable parameter didalam method construct yang kita gunakan. Setelah itu dilanjutkan dengan melakukan instansiasi class objek serta kita sudah memiliki 2 objek yang dikumpulkan dalam 1 variable array, berikut base-code untuk menampilkan data user



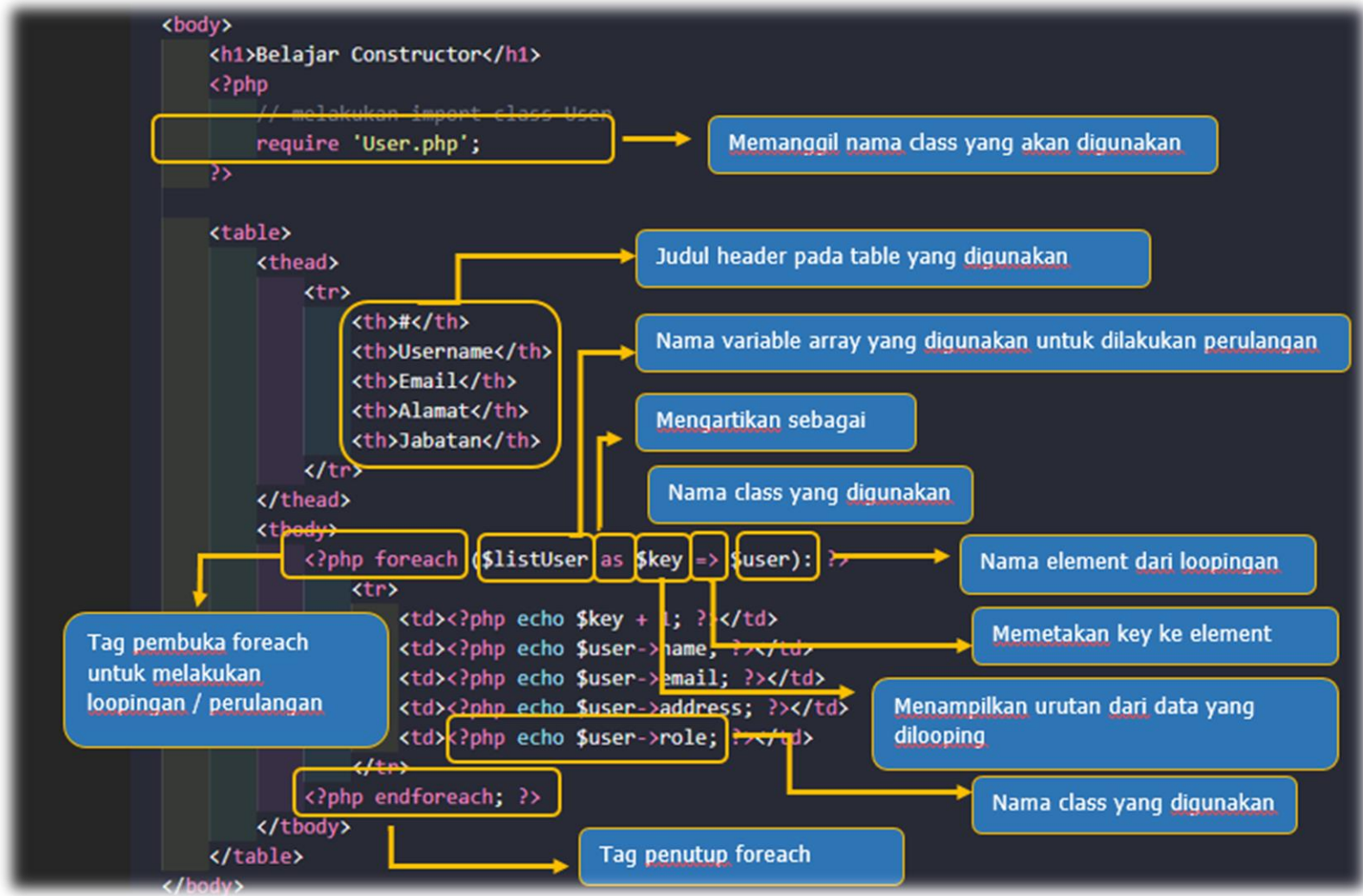
**Gambar 2.2** index.php sebagai code menampilkan ke dalam tampilan halaman

The screenshot shows a web browser window with the following elements and annotations:

- Browser Tab:** Labeled "Belajar Konstruktor".
- Address Bar:** Shows the URL `localhost/demo-galeri-photo/belajar-oop-php/constructor/`. An annotation "Alamat url berdasarkan nama proyek" points to the `project` part of the URL.
- Page Title:** "Belajar Constructor".
- Table:** A table with 5 columns: #, Username, Email, Alamat, and Jabatan. It contains 2 rows of data.
- Annotations:** Several blue boxes with yellow arrows pointing to specific parts of the page:
  - "Tag penutup foreach" points to the closing tag of the table loop.
  - "Tag penutup foreach" points to the closing tag of the page content.
  - "Belajar Constructor" points to the page title.
  - "Data user yang ditampilkan. Berdasarkan data user yang ditampilkan ada 2 objek, namun ditanya yang ditampilkan sama dikarenakan nilai default yang diberikan nilai didalam construct yang telah dibuat" points to the table data.

| # | Username | Email            | Alamat  | Jabatan |
|---|----------|------------------|---------|---------|
| 1 | Asep     | user@example.com | Bandung | user    |
| 2 | Asep     | user@example.com | Bandung | user    |

Berdasarkan data user yang ditampilkan, data user ada 2 atau objek yang telah diinstansiasi ada 2 namun data yang ditampilkan sama dikarenakan nilai yang dalam construct telah diberi nilai default.



Gambar 2.2 index.php