

NAMA : Yanuarin Salwa Afranita  
NIM : 215150600111018  
KELAS : PWTI - A  
BAB : 7 - Form  
ASISTEN : Iqbal Biondy

## 1. Langkah-Langkah Praktikum

### a. Metode GET dan POST

Tulis kode berikut.

```
1 <body>
2   <form action="proses.php" method="post">
3     <input type="text" name="nama">
4     <input type="submit" value="Go">
5   </form>
6 </body>
```

Jalankan kode tersebut di browser lalu ganti metodenya dengan GET. Lihat perbedaannya pada URL di browser. Beri kesimpulan.

### b. Menentukan Metode

Untuk beberapa kasus berikut, tentukan metode apakah yang cocok untuk digunakan dan beri alasan.

1. Form untuk login.
2. Form untuk mengirim gambar desain poster ke website percetakan.
3. Form untuk mencari suatu artikel di situs berita.
4. Form untuk search engine.

### c. Form

Tulis kode di bawah ini.

```
1 <form action="proses02.php" method="post" name="input">
2   Nama Anda: <input type="text" name="nama" required>
3   <br>
4   <input type="submit" name="input" value="Input">
5 </form>
```

Kemudian tulis kode di bawah ini.

```
1 <?php
2     if (isset($_POST['input'])) {
3         $nama = $_POST['nama'];
4         echo "Nama Anda: <b>$nama</b>";
5     }
```

1. Jalankan kode di atas kemudian isikan data input dan tekan tombol input. Apa yang terjadi?
2. Ulangi dengan mengosongkan data input dan amati apa yang terjadi.
3. Jelaskan alur pengiriman data dari form yang ada di kode tersebut!

#### d. Required

Ubah kembali kode pada Latihan 3 dengan menghapus tulisan required pada baris 2.

Jalankan kode dan kosongi data nama kemudian tekan tombol Input. Apa yang terjadi?

Jelaskan apa fungsi dari kode required.

#### e. Upload

Tulis kode berikut.

```
1 <form enctype="multipart/form-data" action="upload.php" method="post">
2     Choose a file to upload:
3     <input name="uploadedfile" type="file" /> <br>
4     <input type="submit" value="Upload File" />
5 </form>
```

Tulis kode berikut.

```
1 <?php
2     $target_path = "uploads/";
3     $target_path = $target_path . basename($_FILES['uploadedfile']['name']);
4
5     if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'],
6         $target_path)) {
7         echo "The file ". basename( $_FILES['uploadedfile']['name']). " has
8         been uploaded";
9     } else {
10        echo "There was an error uploading the file, please try again!";
11    }
```

Di manakah posisi file setelah proses upload berhasil?

Apakah hasil output dari kode tersebut setelah dijalankan?

## 2. Pembahasan Tugas

## a. Metode GET dan POST

### 2. 1 Source code

#### Method GET

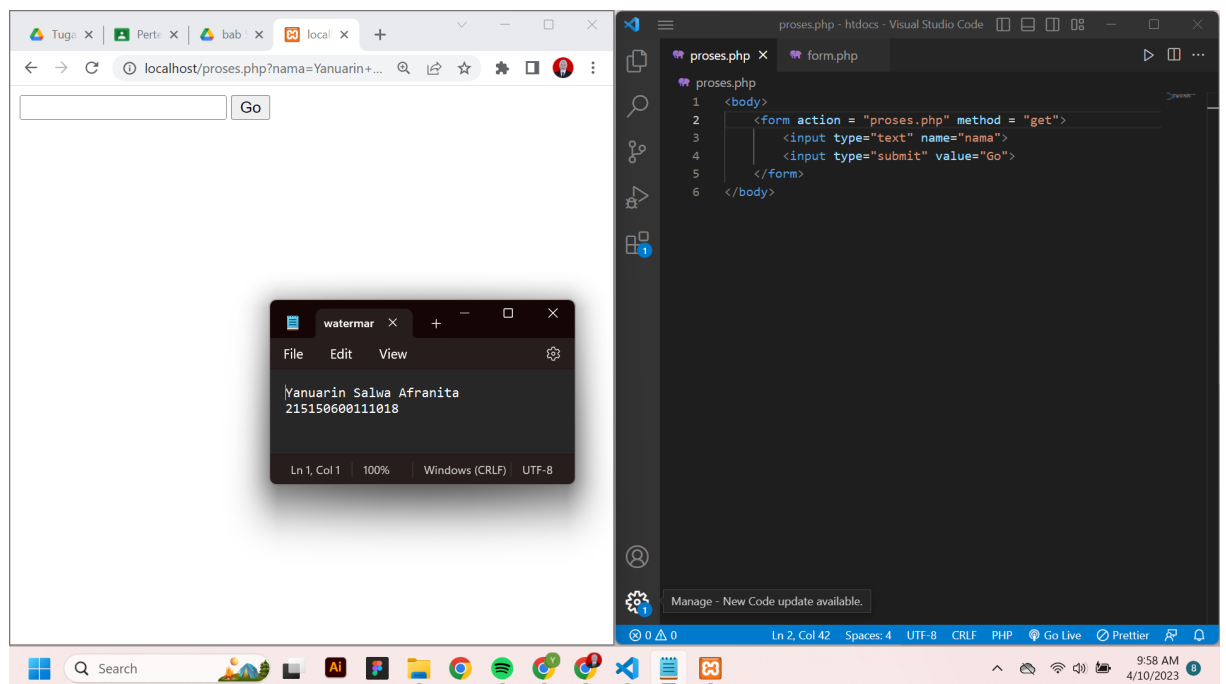
```
<body>
  <form action = "proses.php" method = "get">
    <input type="text" name="nama">
    <input type="submit" value="Go">
  </form>
</body>
```

#### Method POST

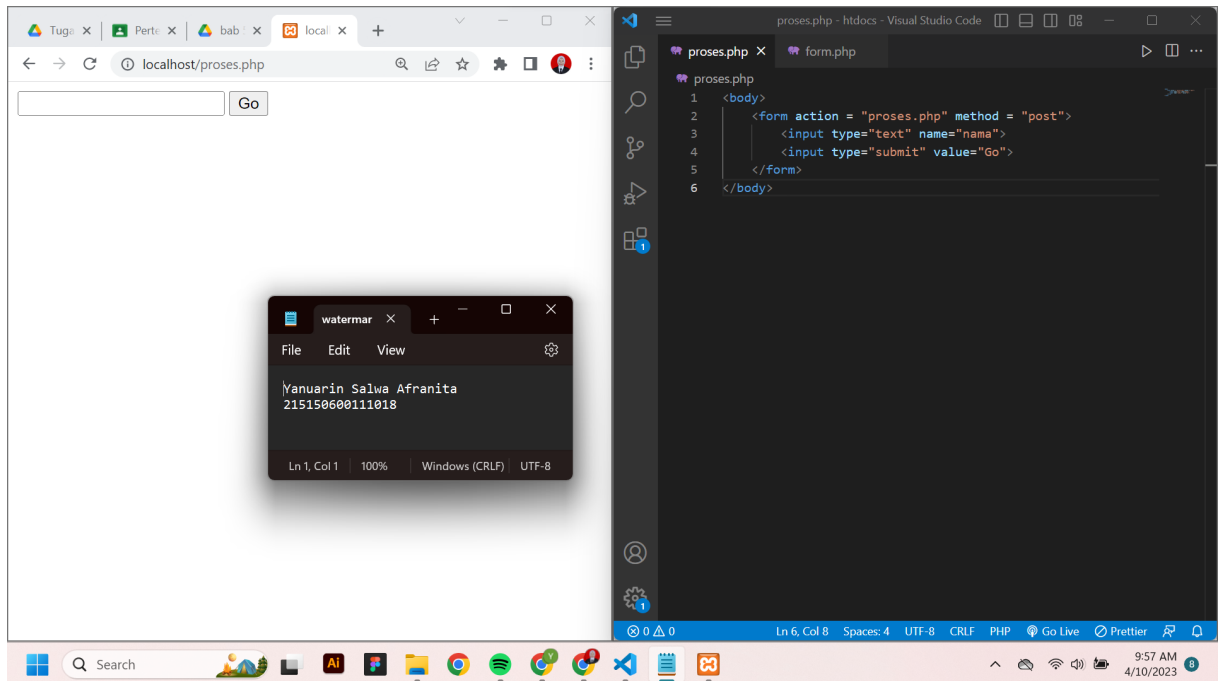
```
<body>
  <form action = "proses.php" method = "post">
    <input type="text" name="nama">
    <input type="submit" value="Go">
  </form>
</body>
```

### 2. 2 Screenshot

#### Metode GET



#### Metode POST



### Pembahasan:

Perbedaan antara penggunaan method POST dan GET adalah URL yang ditampilkan. Saat menggunakan method GET tampilan URL akan berubah sesuai dengan inputan yang kita inginkan. Namun, ketika menggunakan method POST, hasil dari inputan kita tidak akan ditunjukkan dalam URL sehingga lebih aman.

### b. Menentukan Metode

#### Pembahasan:

#### 1. Form untuk login

Menggunakan **method POST**. Karena data yang digunakan untuk login biasanya berupa password yang sifatnya rahasia dan membutuhkan keamanan. Sehingga method POST lebih sesuai digunakan pada form login agar mengurangi resiko bocornya security dari password apabila menggunakan method GET karena bisa dilihat pada URL-nya.

#### 2. Form untuk mengirim gambar desain poster ke website percetakan.

Menggunakan **method POST**. Karena biasanya file gambar memiliki ukuran yang besar. Sehingga, method POST lebih sesuai digunakan karena method GET memiliki keterbatasan ukuran.

#### 3. Form untuk mencari suatu artikel di situs berita.

Menggunakan **method GET**. karena dengan menggunakan method GET inputan user akan terlihat di dalam URL dan memudahkan user untuk membaca URL dan menyimpannya apabila akan membuka URL itu kembali.

4. Form untuk search engine.

Menggunakan **method GET**. Agar user dapat melihat inputannya di dalam URL yang mana akan mempermudah user dalam memahami URL yang ia kunjungi.

c. Form

2. 1 Source Code

Tulis Kode 1

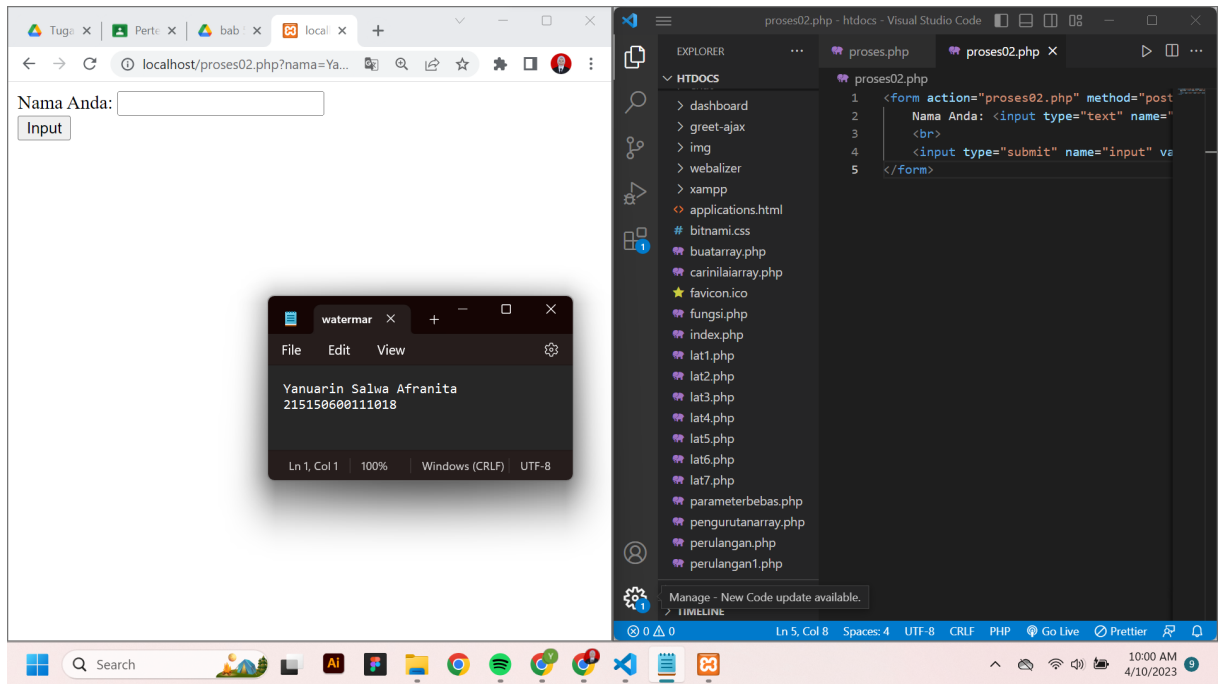
```
<form action="proses02.php" method="post" name="input">
    Nama Anda: <input type="text" name="nama" required>
    <br>
    <input type="submit" name="input" value="Input">
</form>
```

Tulis Kode 2

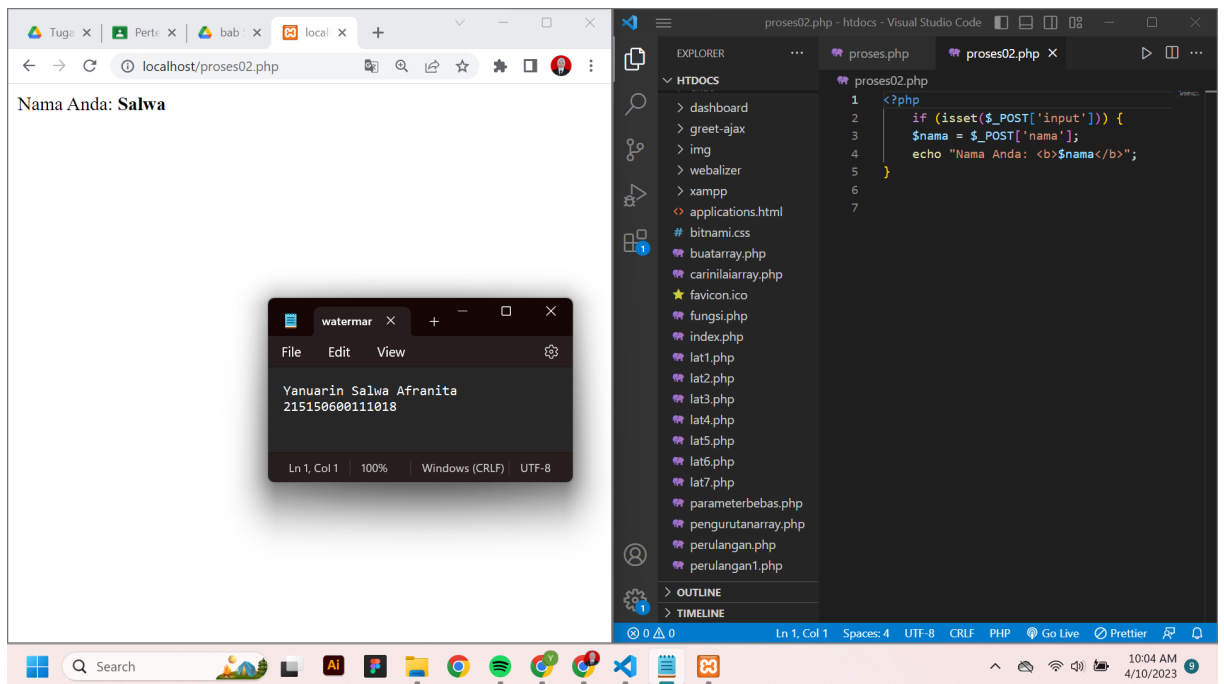
```
<?php
    if (isset($_POST['input'])) {
        $nama = $_POST['nama'];
        echo "Nama Anda: <b>$nama</b>";
    }
```

2. 2 Screenshot

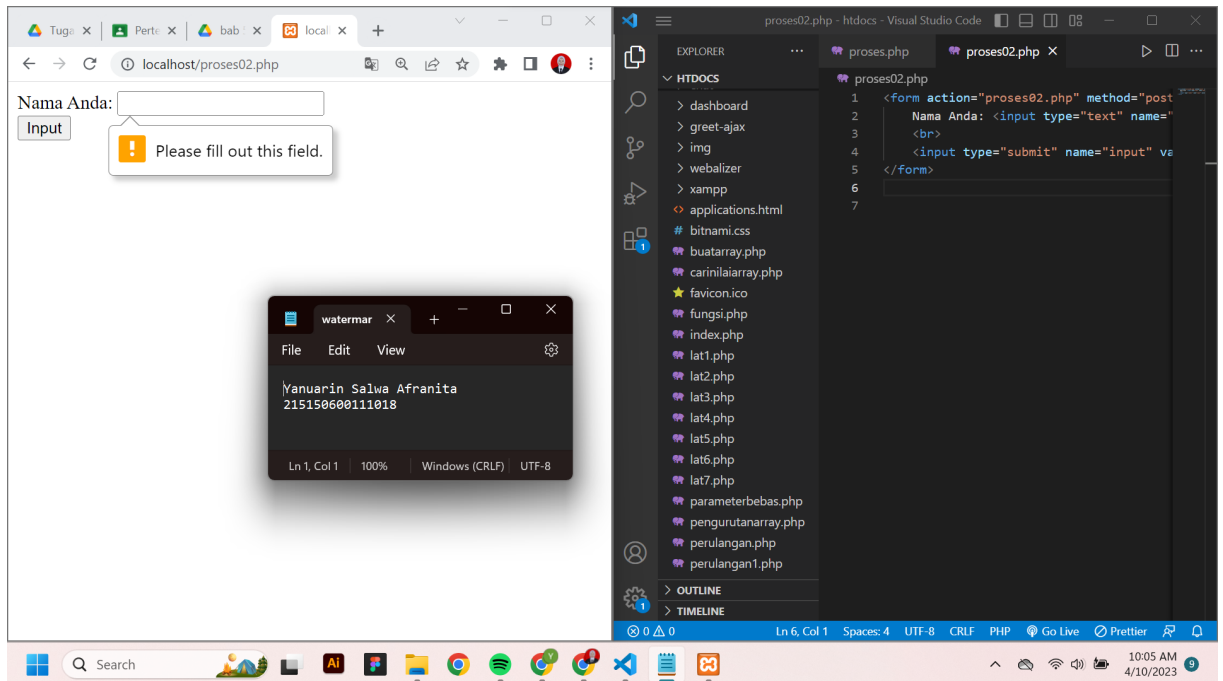
Tulis kode 1



## Tulis kode 2



## Input kosong



### Pembahasan:

1. Jalankan kode di atas kemudian isikan data input dan tekan tombol input. Apa yang terjadi?

Saat kita menginputkan sebuah data lalu menekan tombol input, maka kita akan diarahkan ke laman proses02.php dimana pada laman tersebut akan menampilkan 'Nama anda : ' sesuai dengan inputan yang dimasukkan sebelumnya.

2. Ulangi dengan mengosongkan data input dan amati apa yang terjadi.

Saat mengosongkan field text dan menekan tombol input, maka akan muncul warning atau pop up peringatan dimana field text tersebut harus diisi terlebih dahulu. Sehingga, laman tidak akan dialihkan ke laman proses02.php sebelum user mengisi field text pada laman form tersebut.

3. Jelaskan alur pengiriman data dari form yang ada di kode tersebut!

Alur dari pengiriman datanya adalah pada laman form kita memberikan atribut agar form dapat beralih untuk menjalankan laman proses02.php dimana kita menggunakan method POST. Apabila kita telah mengisi data dan menekan tombol input, maka form akan menjalankan laman proses02.php. Sedangkan, pada laman proses02.php akan dilakukan pengecekan apakah variabel \$\_POST yang memiliki index 'input' telah berjalan? Jika iya, maka data akan disimpan

dari variabel \$\_POST kepada variabel \$nama, lalu akan dicetak ‘Nama anda : \$nama’.

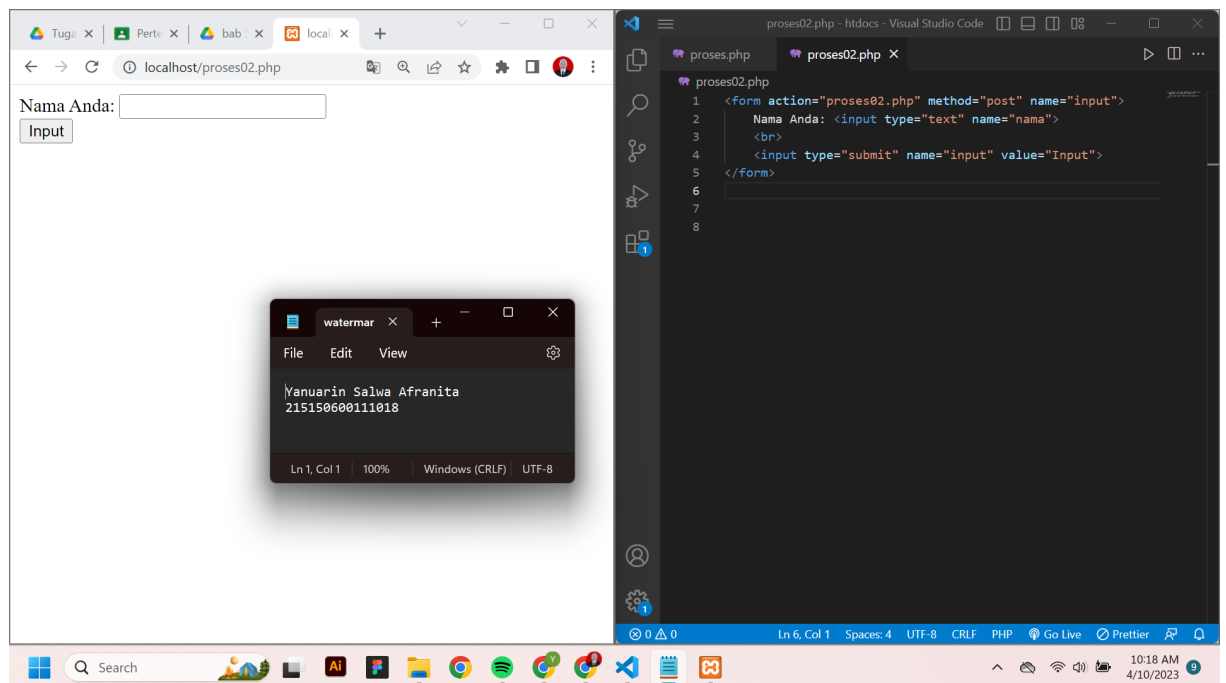
#### d. Required

##### 2. 1 Source Code



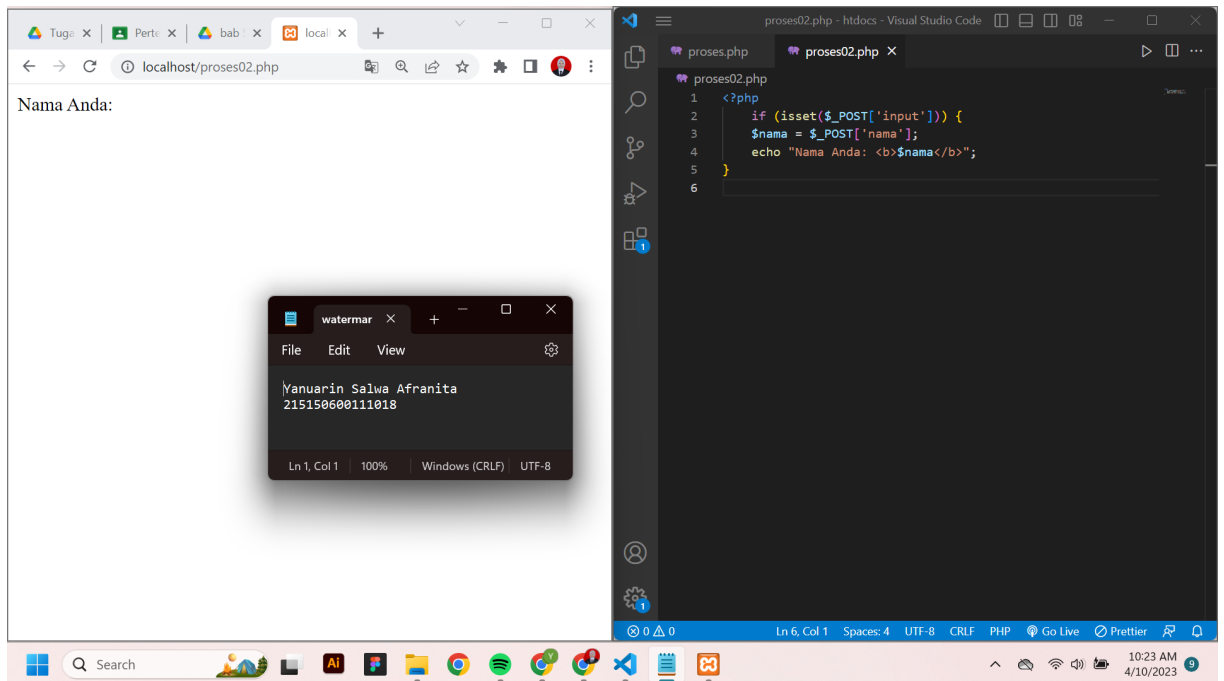
##### 2. 2 Screenshot

Tanpa “required”



Input kosong





### **Pembahasan:**

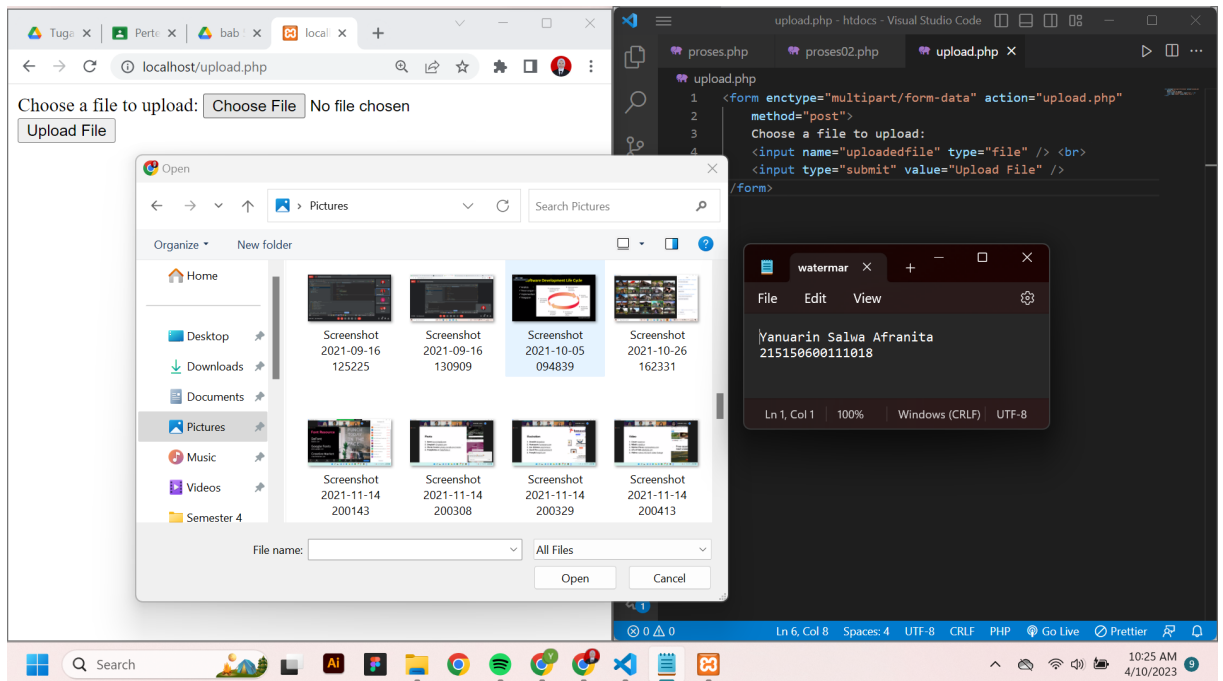
Dengan menghapus required pada baris 2 hal yang berubah ialah tidak ada pop up warning atau peringatan ketika kita menekan tombol input tanpa memasukkan data apapun. Sehingga, data akan tetap dijalankan dan menuju ke proses02.php walaupun output yang ditampilkan adalah data kosong.

## **e. Upload**

### **2. 1 Source Code**



### **2. 2 Screenshot**



### Pembahasan:

1. Setelah berhasil diupload, posisi file berada pada direktori C:\xampp\htdocs\praktikum\uploads. Karena pada code yang dijalankan, kita menuliskan variabel `$target__path` yang akan mengarahkan file-file yang diupload menuju direktori folder uploads pada lokasi file yang sama dengan upload.php.
2. Hasil dari kode tersebut setelah dijalankan yaitu akan menampilkan form yang berfungsi untuk mengupload file. Ketika menekan tombol choose file akan otomatis diarahkan ke penyimpanan pada device untuk memilih file apa yang akan diupload. Setelah itu, apabila kita menekan tombol 'Upload File' maka form akan menjalankan laman upload.php. Apabila file berhasil diupload, maka akan menampilkan "The file [file name] has been uploaded". Sedangkan, apabila gagal mengupload akan menampilkan "There was an error uploading the file, please try again!".