

LATIHAN PRAKTIKUM PENGEMBANGAN APLIKASI WEB

BAB : FORM
NAMA : DANI ADRIAN
NIM : 225150201111009
ASISTEN : - IBAR HUTTAQI SULTHON
 - MUHAMMAD AMMAR
TANGGAL PRAKTIKUM : 28/03/2024

LATIHAN 1

A. Soal

Metode GET dan POST

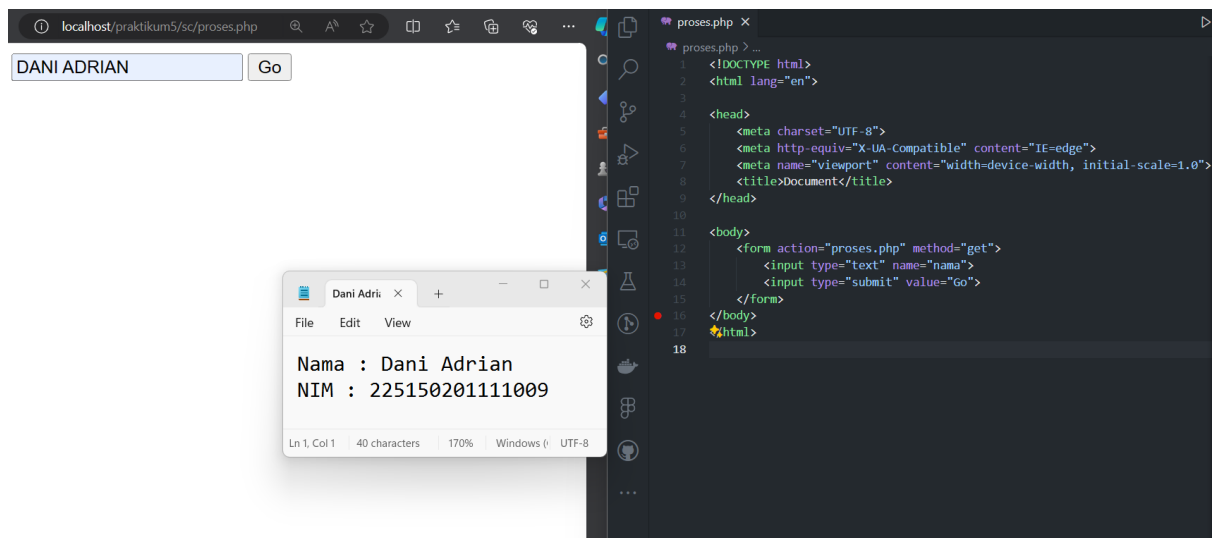
Tulis kode berikut

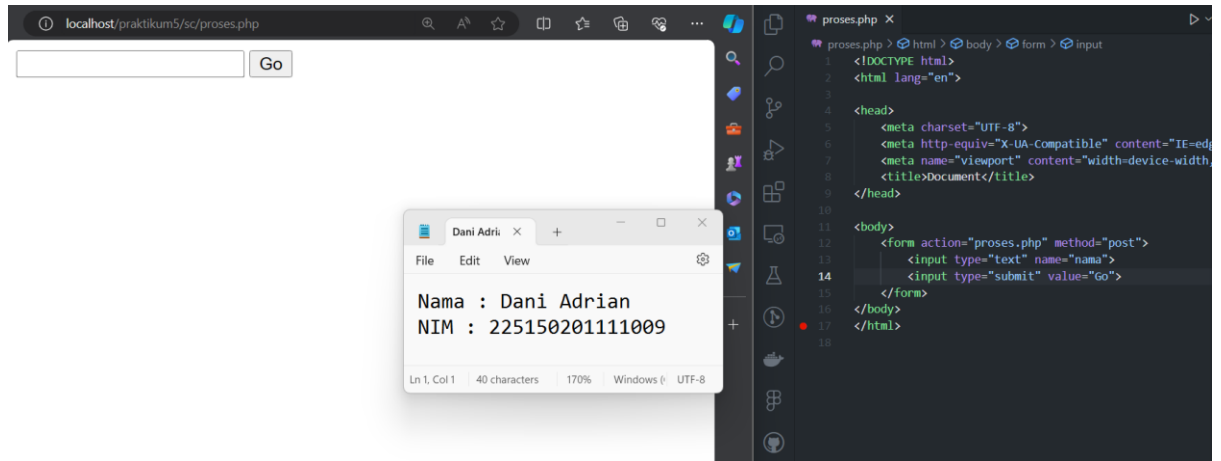
1.	<body>
2.	<form action="proses.php" method="post">
3.	<input type="text" name="nama">
4.	<input type="submit" value="Go">
5.	</form>
6.	</body>

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

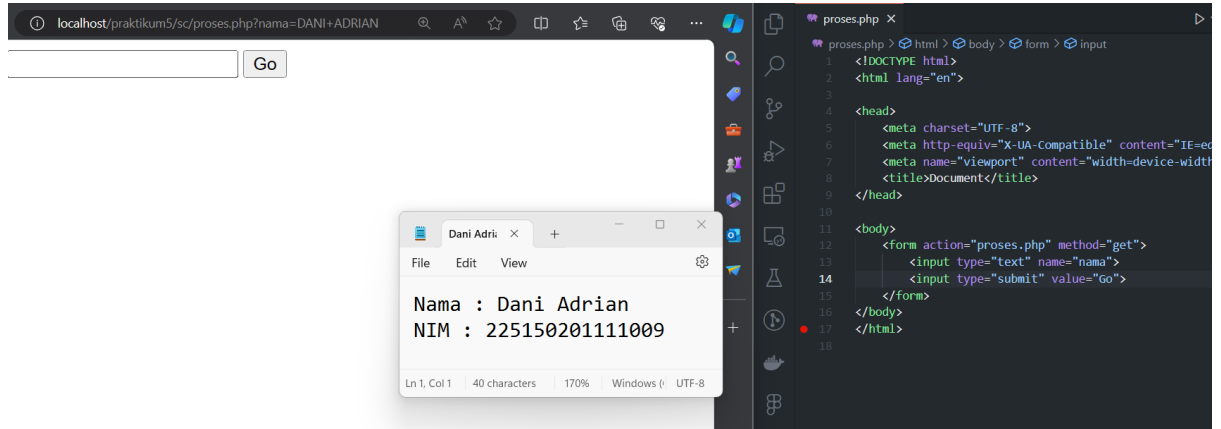
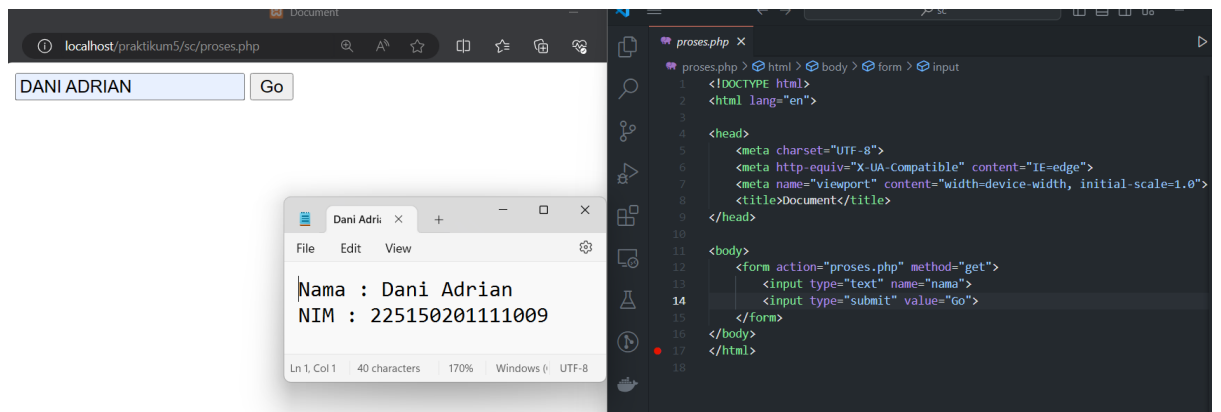
B. Screenshoot

Sebelum dirubah GET





Sesudah dirubah GET



C. Syntax

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>

D. Penjelasan

Perbedaan antara metode POST dan GET terletak pada cara data dikirimkan. Metode POST mengirimkan data melalui body permintaan HTTP, sementara metode GET

mengirimkan data melalui URL. Metode POST lebih aman karena data tidak ditampilkan secara langsung di URL, sedangkan metode GET lebih sesuai digunakan untuk permintaan yang bersifat idempoten dan ketika Anda ingin data ditampilkan di URL.

LATIHAN 2

A. Soal

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.

B. Screenshoot

C. Syntax

D. Penjelasan

1. Form untuk login

Metode yang cocok: POST.

Alasan: Informasi login, seperti nama pengguna dan kata sandi, adalah sensitif dan harus dilindungi. Metode POST menyembunyikan informasi ini dari URL dan mengirimkannya melalui `body` permintaan HTTP, sehingga lebih aman daripada GET.

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

Metode yang cocok: POST.

Alasan: Gambar desain poster mungkin memiliki ukuran besar dan tidak cocok untuk disertakan dalam URL. Metode POST memungkinkan pengguna untuk mengunggah file gambar dengan aman melalui `body` permintaan HTTP.

3. Form untuk mencari suatu artikel di situs berita:

Metode yang cocok: GET.

Alasan: Permintaan pencarian adalah permintaan idempoten yang tidak mengubah status atau data di server. Pengguna mungkin ingin menyalin dan menyimpan URL hasil pencarian untuk referensi di masa depan. Dengan

menggunakan metode GET, hasil pencarian dapat disertakan dalam URL sehingga mudah untuk dibagikan atau disimpan.

4. Form untuk search engine:

Metode yang cocok: GET.

Alasan: Search engine seperti Google menggunakan metode GET untuk permintaan pencarian. Pengguna ingin melihat hasil pencarian dengan cepat dan kemudahan dalam pembagian hasil pencarian adalah faktor yang penting. Dengan menggunakan metode GET, query pencarian dapat disertakan dalam URL sehingga pengguna dapat dengan mudah membagikan atau menyimpan hasil pencarian yang relevan.

LATIHAN 3

A. Soal

Form

Tulis kode dibawah ini

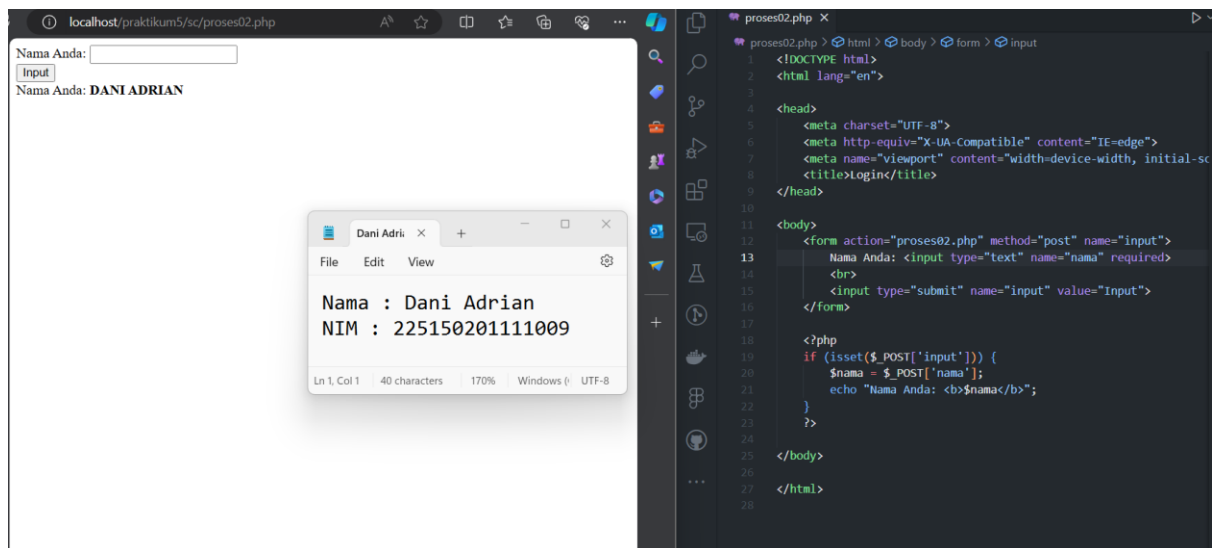
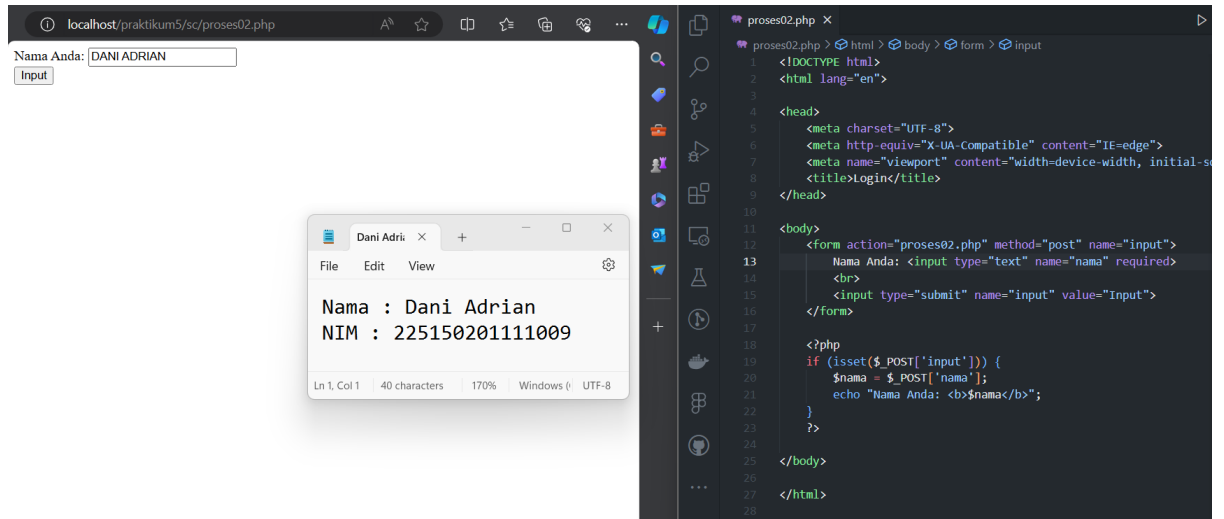
1.	<form action="proses02.php" method="post" name="
2.	input">
3.	Nama Anda: <input type="text" name="nama"
4.	required>
5.	
	<input type="submit" name="input"
	value="Input">
	</form>

Kemudian tulis kode dibawah ini.

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

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

B. Screenshoot



C. Syntax

proses02.php	
1.	<!DOCTYPE html>
2.	<html lang="en">
3.	
4.	<head>
5.	<meta charset="UTF-8">
6.	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7.	<meta name="viewport" content="width=device-
	width, initial-scale=1.0">
8.	<title>Login</title>
9.	</head>
10.	
11.	<body>
12.	<form action="proses02.php" method="post"
	name="input">
13.	Nama Anda: <input type="text" name="nama"
	required>
14.	
15.	<input type="submit" name="input"
	value="Input">

```

16.         </form>
17.
18.         <?php
19.         if (isset($_POST['input'])) {
20.             $nama = $_POST['nama'];
21.             echo "Nama Anda: <b>$nama</b>";
22.         }
23.         ?>
24.
25.     </body>
26.
27. </html>

```

D. Penjelasan

- Ketika tombol input ditekan, browser akan mengirimkan data dari input ke berkas proses02.php menggunakan metode POST.
- Di dalam proses02.php, kode PHP akan memeriksa apakah ada data yang dikirimkan melalui formulir dengan menggunakan `isset($_POST['input'])`.
- Jika ada data yang dikirimkan, maka nilai dari input dengan nama "nama" akan disimpan dalam variabel `$nama`.
- Kemudian, nilai dari variabel `$nama` akan ditampilkan dalam sebuah pesan dengan format "Nama Anda: [nilai nama]".

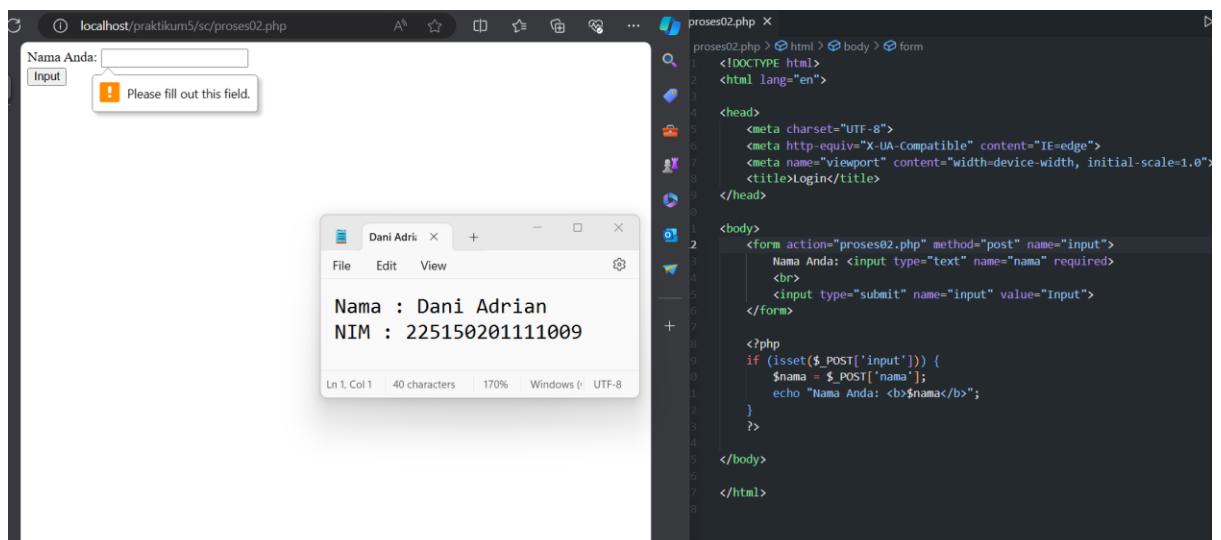
LATIHAN 4

A. Soal

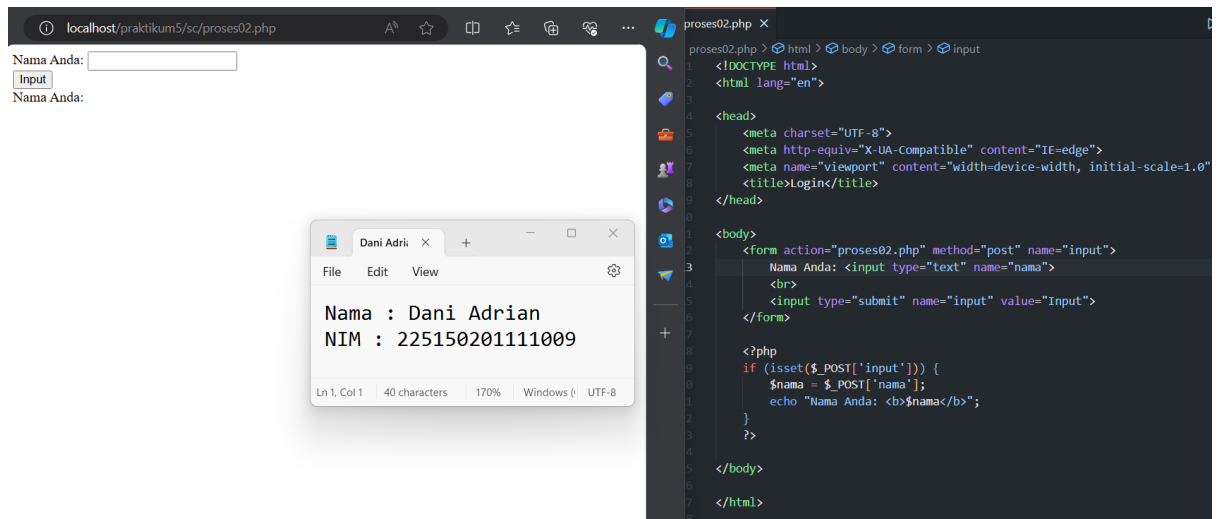
Ubah kembali kode pada Latihan 3 dengan menghapus tulisan `required` pada baris 2. Jalankan kode dan kosongi data nama kemudian tekan tombol Input. Apa yang terjadi? Jelaskan apa fungsi dari kode `required`.

B. Screenshoot

Sebelum



Sesudah



C. Syntax

proses02.php	
1.	<!DOCTYPE html>
2.	<html lang="en">
3.	
4.	<head>
5.	<meta charset="UTF-8">
6.	<meta http-equiv="X-UA-Compatible"
7.	content="IE=edge">
8.	<meta name="viewport" content="width=device-
9.	width, initial-scale=1.0">
10.	<title>Login</title>
11.	</head>
12.	<body>
13.	<form action="proses02.php" method="post"
14.	name="input">
15.	Nama Anda: <input type="text" name="nama">
16.	
17.	<input type="submit" name="input"
18.	value="Input">
19.	</form>
20.	<?php
21.	if (isset(\$_POST['input'])) {
22.	\$nama = \$_POST['nama'];
23.	echo "Nama Anda: \$nama";
24.	}
25.	?>
26.	</body>
27.	</html>

D. Penjelasan

Setelah kode `required` dihapus, dilakukan percobaan untuk mengkosongkan field atau kolom kemudian diinput. Yang terjadi adalah halaman akan menampilkan text "Nama Anda : " saja dan tidak diikuti dengan data apapun.

Atribut `required` adalah atribut HTML yang memaksa pengguna untuk mengisi input sebelum formulir dapat dikirim. Dengan demikian, pengguna tidak dapat mengirimkan formulir jika input yang wajib diisi belum diisi. Ini membantu untuk memastikan data yang diperlukan oleh formulir untuk diproses oleh server tersedia dan tidak kosong.

LATIHAN 5

A. Soal

Tulis kode berikut.

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

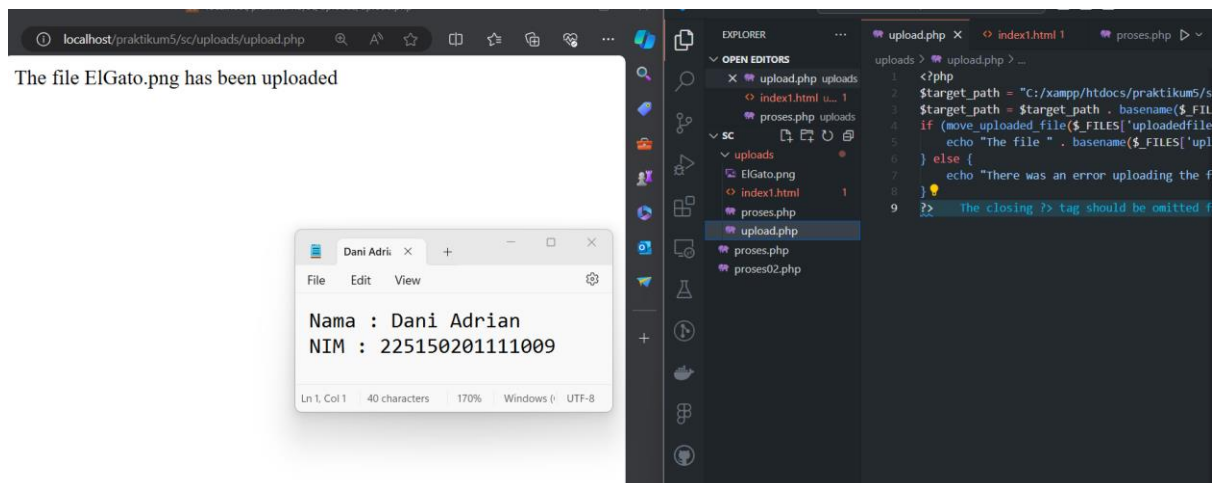
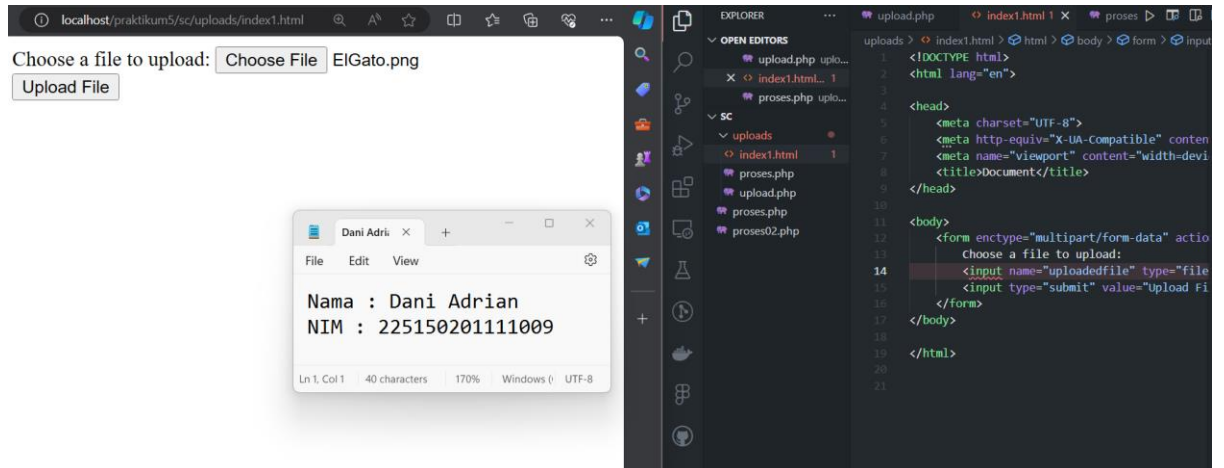
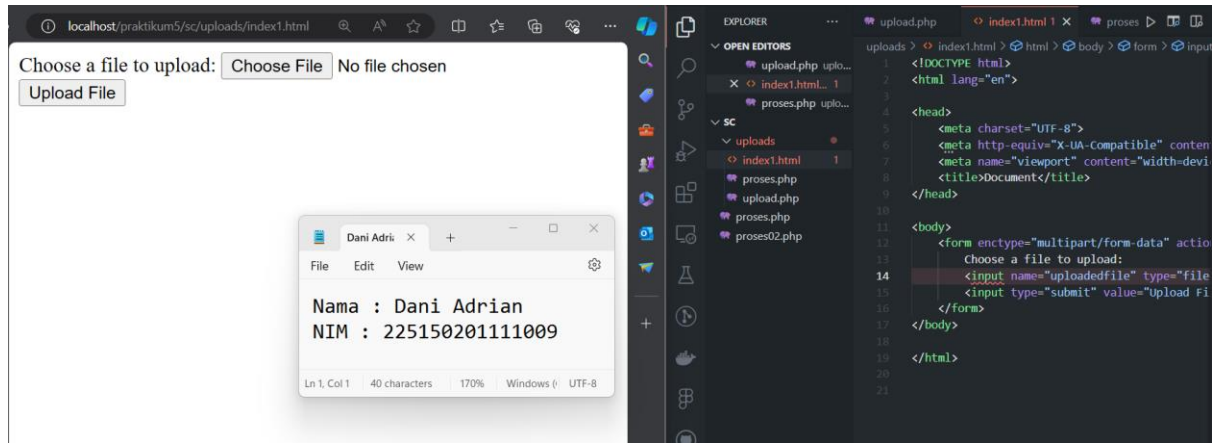
Tulis kode berikut.

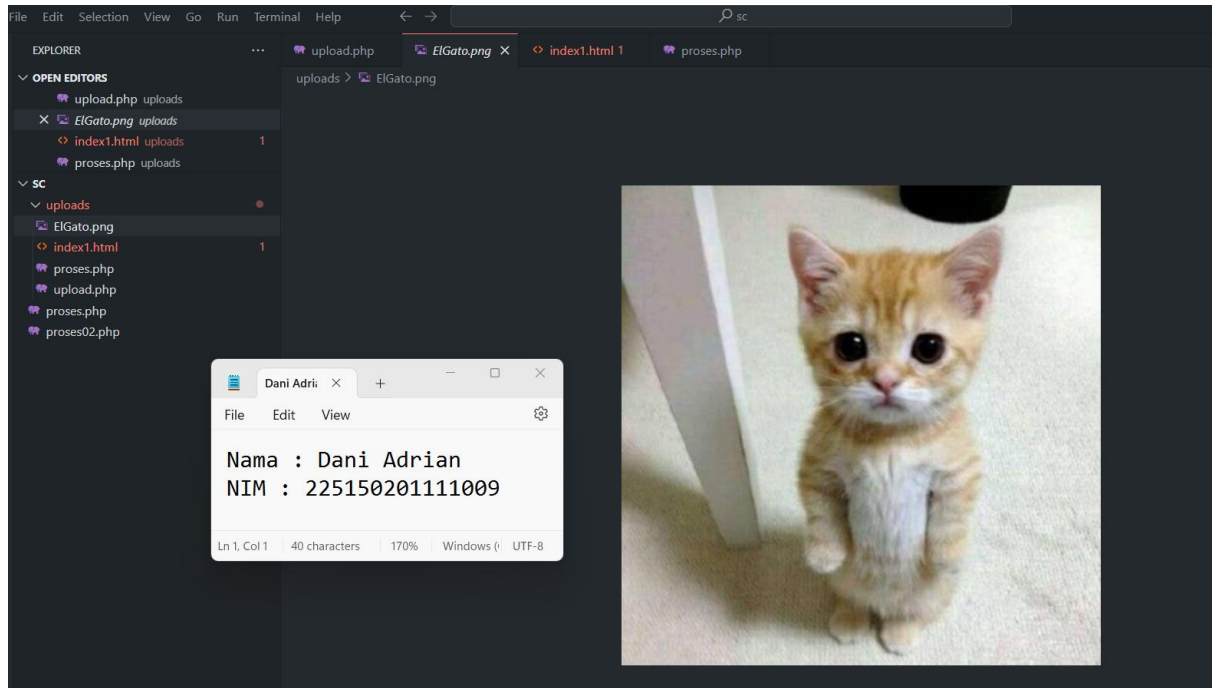
1.	<code><?php</code>
2.	<code>\$target_path = "uploads/";</code>
3.	<code>\$target_path = \$target_path .</code>
	<code>basename(\$_FILES['uploadedfile']['name']);</code>
4.	<code>if (move_uploaded_file(</code>
5.	<code>\$_FILES['uploadedfile']['tmp_name'],</code>
6.	<code>\$target_path</code>
7.	<code>))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

Di manakah posisi file setelah proses upload berhasil?

Apakah hasil output dari kode tersebut setelah dijalankan?

B. Screenshoot





C. Syntax

index1.html	
1.	<!DOCTYPE html>
2.	<html lang="en">
3.	
4.	<head>
5.	<meta charset="UTF-8">
6.	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7.	<meta name="viewport" content="width=device-
	width, initial-scale=1.0">
8.	<title>Document</title>
9.	</head>
10.	
11.	<body>
12.	<form enctype="multipart/form-data"
	action="upload.php" method="post">
13.	Choose a file to upload:
14.	<input name="uploadedfile" type="file" />
15.	<input type="submit" value="Upload File" />
16.	</form>
17.	</body>
18.	
19.	</html>

proses.php	
1.	<!DOCTYPE html>
2.	<html lang="en">
3.	
4.	<head>
5.	<meta charset="UTF-8">

6.	<meta http-equiv="X-UA-Compatible"
7.	content="IE=edge">
8.	<meta name="viewport" content="width=device-
9.	width, initial-scale=1.0">
10.	<title>Document</title>
11.	</head>
12.	<body>
13.	<form action="proses.php" method="post">
14.	<input type="text" name="nama">
15.	<input type="submit" value="Go">
16.	</form>
17.	</body>
	</html>

upload.php	
1.	<?php
2.	\$target_path =
3.	"C:/xampp/htdocs/praktikum5/sc/uploads/";
4.	\$target_path = \$target_path .
5.	basename(\$_FILES['uploadedfile']['name']);
6.	if
7.	(move_uploaded_file(\$_FILES['uploadedfile']['tmp_nam
8.	e', \$target_path)) {
9.	echo "The file " .
	basename(\$_FILES['uploadedfile']['name']) . " has
	been uploaded";
	} else {
	echo "There was an error uploading the file,
	please try again!";
	}
	?>

D. Penjelasan

Posisi File Setelah Upload Berhasil:

Setelah proses upload berhasil, file akan disimpan di direktori C:\xampp\htdocs\praktikum5\sc\uploads, sesuai dengan nilai variabel \$target_path yang telah ditentukan dalam script upload.php.

Hasil Output:

Proses upload berhasil, hasil output adalah:

The file ElGato.png has been uploaded