A. JavaScript

1) Struktur Data

Variabel `list_tugas`

javascript
Copy
```
let list_tugas = [""];
```

Variabel ini adalah array yang digunakan untuk menyimpan daftar tugas. Pada awalnya, array diinisialisasi dengan elemen kosong, tetapi sebaiknya dimulai dengan array kosong (`let list_tugas = [];`) untuk menghindari masalah saat merender tugas.

2) Elemen Output

Pengambilan Elemen Output

javascript
Copy
```
const output_element = document.querySelector("#output");
```

Baris ini menggunakan `document.querySelector` untuk memilih elemen div dengan ID `output`, di mana daftar tugas akan ditampilkan. Elemen ini akan diperbarui setiap kali pengguna menambahkan atau menghapus tugas.

3) Fungsi Render Tugas

Fungsi `renderTasks`

javascript
Copy
```
function renderTasks() {
  output_element.innerHTML = "";
  list_tugas.forEach((value, index) => {
    const p_elm = document.createElement("p");
    p_elm.textContent = value;

    p_elm.addEventListener("click", () => {
      list_tugas.splice(index, 1);
      renderTasks();
    });

    output_element.appendChild(p_elm);
  });
}
```

Fungsi ini bertanggung jawab untuk menampilkan daftar tugas di dalam elemen output. Berikut adalah langkah-langkah yang dilakukan oleh fungsi ini:

1. **Mengosongkan Elemen Output**: `output_element.innerHTML = "";` memastikan bahwa setiap kali fungsi ini dipanggil, konten sebelumnya dihapus agar tidak terjadi pengulangan saat merender tugas baru.
2. **Mengiterasi Daftar Tugas**: Menggunakan `forEach`, setiap tugas dalam `list_tugas` diambil dan diproses. Setiap tugas diubah menjadi elemen paragraf (`<p>`) dan diisi dengan teks tugas.
3. **Event Listener untuk Menghapus Tugas**: Setiap elemen paragraf memiliki event listener yang mendengarkan klik. Saat pengguna mengklik tugas, tugas tersebut akan dihapus dari daftar menggunakan metode `splice`, dan fungsi `renderTasks` dipanggil kembali untuk memperbarui tampilan.
4. **Menambahkan Elemen ke Output**: Setiap elemen paragraf ditambahkan ke dalam elemen output.

4) Penanganan Formulir

Event Listener untuk Formulir

javascript
Copy

```javascript
const formulir = document.querySelector("form");

formulir.addEventListener("submit", (e) => {
  e.preventDefault();

  const formData = new FormData(e.target);
  const obj = Object.fromEntries(formData);

  if (obj.tugas) {
    list_tugas.push(obj.tugas);
    renderTasks();
  }

  e.target.reset();
});
```

Bagian ini menangani pengiriman formulir untuk menambahkan tugas baru ke dalam daftar. Berikut adalah penjelasannya:

1. **Menangkap Elemen Formulir**: Dengan `document.querySelector("form")`, kode ini menangkap elemen formulir yang ada di dalam HTML.
2. **Mencegah Pengiriman Default**: `e.preventDefault();` digunakan untuk mencegah perilaku default dari formulir, yang biasanya akan menyebabkan halaman memuat ulang.
3. **Mengambil Data dari Formulir**: `new FormData(e.target)` digunakan untuk mengambil data dari formulir yang disubmit. Selanjutnya, data tersebut diubah menjadi objek menggunakan `Object.fromEntries`.
4. **Menambahkan Tugas**: Jika input tugas tidak kosong (`if (obj.tugas)`), tugas akan ditambahkan ke dalam `list_tugas`, dan fungsi `renderTasks` dipanggil untuk memperbarui tampilan.
5. **Reset Formulir**: `e.target.reset();` digunakan untuk mengosongkan input setelah tugas ditambahkan.

B. HTML

1) Struktur Kode HTML

Kode HTML yang disajikan terdiri dari beberapa elemen penting yang berfungsi untuk membangun antarmuka pengguna. Berikut adalah komponen utama dari kode tersebut:

a. Deklarasi DOCTYPE dan Elemen HTML

Pada baris pertama, terdapat deklarasi DOCTYPE yang menunjukkan bahwa dokumen ini adalah dokumen HTML5. Ini diikuti dengan tag `<html>` yang menandai awal dari dokumen HTML dan menetapkan bahasa yang digunakan sebagai bahasa Inggris.

html
Copy

```
<!DOCTYPE html>
<html lang="en">
```

RefreshNew tab
Share

Console

Close console

b. Bagian Kepala (Head)

Bagian `<head>` berisi informasi meta tentang dokumen, termasuk pengaturan karakter dan pengaturan tampilan responsif untuk perangkat seluler. Ini juga mencakup judul halaman yang muncul di tab browser.

html
Copy

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
</head>
```

RefreshNew tab
Share

Console

Close console

c. Bagian Tubuh (Body)

Bagian `<body>` adalah tempat konten yang ditampilkan kepada pengguna berada. Di sini terdapat elemen-elemen berikut:

1. **Formulir untuk Input Tugas**
    - Formulir ini terdiri dari elemen input yang memungkinkan pengguna untuk memasukkan tugas dan tombol untuk mengirimkan data tersebut.

html

Copy

```
<form action="">
  <input type="text" name="tugas" placeholder="masukan tugas" />
  <button type="submit">tambah</button>
</form>
```

RefreshNew tab

Share

Console

Close console

2. **Div Output**
   - Elemen `<div>` dengan ID `output` berfungsi sebagai tempat untuk menampilkan daftar tugas yang telah ditambahkan oleh pengguna.

html

Copy

```
<div id="output"></div>
```

RefreshNew tab

Share

Console

Close console

d. Penyertaan Skrip JavaScript

Di akhir dokumen, terdapat dua tag `<script>` yang menyertakan file JavaScript eksternal (`aplikasi.js` dan `data.js`). Skrip ini bertanggung jawab untuk menangani logika aplikasi, seperti pengolahan data input dan pembaruan tampilan output.

html
Copy

```
<script src="aplikasi.js"></script>
<script src="data.js"></script>
```

RefreshNew tab
Share

Console

Close console

2) Fungsi Utama Aplikasi

Aplikasi ini dirancang untuk memenuhi fungsi dasar dalam manajemen tugas:

- **Input Tugas**: Pengguna dapat mengetikkan tugas yang ingin dikerjakan pada kolom input.
- **Menampilkan Tugas**: Setelah tugas ditambahkan, aplikasi akan menampilkan daftar tugas pada elemen output.
- **Interaktivitas**: Dengan menggunakan JavaScript, aplikasi dapat memperbarui tampilan secara dinamis tanpa perlu memuat ulang halaman.