

Nama : Gizza Auralia Pradnya.G

KLS : SIK\_A3

MATKUL : PBO

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Manajemen Perikanan</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="style.css">
</head>
```

<!DOCTYPE html>: Menunjukkan bahwa dokumen ini adalah dokumen HTML5.

<html lang="en">: Elemen <html> adalah akar dari dokumen HTML, dan atribut lang menunjukkan bahasa yang digunakan dalam dokumen (dalam hal ini, bahasa Inggris).

<meta charset="UTF-8">: Menentukan encoding karakter untuk halaman, sehingga teks ditampilkan dengan benar.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Mengatur tampilan untuk perangkat mobile agar responsif.

<title>Manajemen Perikanan</title>: Judul halaman yang muncul di tab browser.

<link>: Mengimpor stylesheet dari Bootstrap dan file CSS lokal (style.css), yang digunakan untuk gaya visual.

```
<body>
  <div class="container mt-5">
    <h1 class="text-center mb-4">Manajemen Data Ikan</h1>
```

<body>: Bagian ini berisi konten yang akan ditampilkan di halaman.

<div class="container mt-5">: Menggunakan kelas Bootstrap container untuk mengatur lebar konten, dan mt-5 untuk memberikan margin atas.

<h1>: Menampilkan judul halaman dengan kelas text-center untuk memusatkan teks dan mb-4 untuk margin bawah.

```
<!-- Form Tambah Ikan -->
  <div class="card mb-4">
    <div class="card-body">
      <h4 class="card-title">Tambah Ikan</h4>
      <form id="fishForm">
        <div class="mb-3">
          <label for="fishName" class="form-label">Nama
Ikan</label>
          <input type="text" class="form-control"
id="fishName" placeholder="Masukkan nama ikan" required>
        </div>
```

```

        <div class="mb-3">
            <label for="fishType" class="form-label">Jenis
Ikan</label>
            <input type="text" class="form-control"
id="fishType" placeholder="Masukkan jenis ikan" required>
        </div>
        <div class="mb-3">
            <label for="fishWeight" class="form-label">Berat
Ikan (kg)</label>
            <input type="number" class="form-control"
id="fishWeight" placeholder="Masukkan berat ikan" required>
        </div>
        <button type="submit" class="btn btn-primary">Tambah
Ikan</button>
    </form>
</div>
</div>

```

<div class="card mb-4">: Membuat card Bootstrap untuk menampung form, dengan margin bawah.

Form: Terdapat tiga input (nama ikan, jenis ikan, dan berat ikan) dengan label dan input yang dibungkus dalam elemen div untuk tata letak yang baik.

<button type="submit" class="btn btn-primary">: Tombol untuk mengirimkan form. Kelas btn dan btn-primary dari Bootstrap memberikan gaya yang sesuai.

```

<!-- Daftar Ikan -->
    <div class="card mb-4">
        <div class="card-body">
            <h4 class="card-title">Daftar Ikan</h4>
            <ul id="fishList" class="list-group">
                <!-- Data ikan akan ditampilkan di sini -->
            </ul>
        </div>
    </div>
</div>

```

<div class="card mb-4">: Card lain untuk menampilkan daftar ikan.

<ul id="fishList" class="list-group">: Daftar yang akan diisi dengan data ikan. ID fishList memungkinkan manipulasi dengan JavaScript.

```

<!-- Tabel Ikan -->
    <div class="card">
        <div class="card-body">
            <h4 class="card-title">Daftar Ikan dalam Tabel</h4>
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th>No</th>
                        <th>Nama Ikan</th>
                        <th>Jenis Ikan</th>
                        <th>Berat (kg)</th>
                        <th>Aksi</th>
                    </tr>
                </thead>
            </table>
        </div>
    </div>

```

```

        <tbody id="fishTableBody">
            <!-- Data ikan akan diisi di sini menggunakan
JavaScript -->
        </tbody>
    </table>
</div>
</div>
</div>

```

Tabel: Menyediakan struktur yang teratur untuk menampilkan data ikan dalam bentuk tabel dengan kolom untuk nomor, nama, jenis, berat, dan aksi (edit/hapus).

<tbody id="fishTableBody">: Bagian tabel yang akan diisi dengan data ikan menggunakan JavaScript.

```

<script src="java.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

<script src="java.js"></script>: Mengimpor file JavaScript yang berisi logika untuk menambah, mengedit, dan menghapus data ikan.

<script src="https://cdn.jsdelivr.net/npm

```

let fishList = [];
let currentIndex = null;

```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```

// Fungsi untuk render data ikan ke tabel
function renderFish() {
    const fishTableBody = document.getElementById('fishTableBody');
    const fishListElement = document.getElementById('fishList');

    fishTableBody.innerHTML = ''; // Menghapus isi tabel sebelumnya
    fishListElement.innerHTML = ''; // Menghapus isi daftar ikan sebelumnya

    fishList.forEach((fish, index) => {
        // Render ke dalam tabel
        const row = document.createElement('tr'); // Membuat baris baru
        row.innerHTML = `
            <td>${index + 1}</td>
            <td>${fish.name}</td>
            <td>${fish.type}</td>
            <td>${fish.weight}</td>
            <td>
                <button class="btn btn-warning btn-sm"
onclick="editFish(${index})">Edit</button>

```

```

        <button class="btn btn-danger btn-sm"
onclick="deleteFish(${index})">Delete</button>
      </td>
    `;
    fishTableBody.appendChild(row); // Menambahkan baris ke tabel

    // Render ke dalam daftar
    const listItem = document.createElement('li');
    listItem.className = 'list-group-item';
    listItem.innerText = `${fish.name} (${fish.type}, ${fish.weight}
kg)`;
    fishListElement.appendChild(listItem); // Menambahkan item ke daftar
  });
}

```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```

• / Fungsi untuk menambah ikan
• function addFish(name, type, weight) {
•   fishList.push({ name, type, weight }); // Menambah ikan baru ke
array
•   renderFish(); // Memperbarui tampilan tabel dan daftar
• }

```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```

• // Fungsi untuk mengedit ikan
• function editFish(index) {
•     currentIndex = index; // Menyimpan index ikan yang akan diedit
•     const fish = fishList[index];
•

```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```

•
• // Fungsi untuk mengedit ikan
• function editFish(index) {
•     currentIndex = index; // Menyimpan index ikan yang akan diedit
•     const fish = fishList[index];
•

```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```

• // Mengisi form dengan data ikan yang dipilih
• document.getElementById('fishName').value = fish.name;
• document.getElementById('fishType').value = fish.type;
• document.getElementById('fishWeight').value = fish.weight;
• }

```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```
• // Fungsi untuk mengupdate ikan
• function updateFish(name, type, weight) {
•   fishList[currentIndex] = { name, type, weight }; // Mengupdate
    ikan yang telah diedit
•   currentIndex = null; // Reset index saat selesai
•   renderFish(); // Memperbarui tampilan tabel dan daftar
• }
•
```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai currentIndex.

Mereset form setelah pengiriman data.

```
• // Fungsi untuk menghapus ikan
• function deleteFish(index) {
•   fishList.splice(index, 1); // Menghapus ikan dari array
•   renderFish(); // Memperbarui tampilan tabel dan daftar
• }
•
```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

e.preventDefault(): Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai `currentIndex`.

Mereset form setelah pengiriman data.

```
• // Event listener untuk form tambah dan update ikan
• document.getElementById('fishForm').addEventListener('submit',
  function (e) {
•     e.preventDefault(); // Mencegah form dikirim secara default
•
•     const name = document.getElementById('fishName').value;
•     const type = document.getElementById('fishType').value;
•     const weight = document.getElementById('fishWeight').value;
•
•     // Menentukan apakah kita akan menambah atau mengupdate ikan
•     if (currentIndex === null) {
•         addFish(name, type, weight); // Tambah ikan baru
•     } else {
•         updateFish(name, type, weight); // Update ikan yang ada
•     }
•
•     // Reset form
•     document.getElementById('fishForm').reset();
• });
```

Event Listener: Mendaftarkan fungsi yang akan dijalankan ketika form dikirim.

`e.preventDefault()`: Mencegah pengiriman form secara default, yang bisa menyebabkan halaman reload.

Mengambil nilai dari input form (nama, jenis, berat ikan).

Menggunakan kondisi untuk menentukan apakah akan menambahkan ikan baru atau memperbarui ikan yang ada berdasarkan nilai `currentIndex`.

Mereset form setelah pengiriman data.

```
body {
    background-color: #588dcd; /* Ganti dengan warna latar belakang yang
Anda inginkan */
    color: #343a40; /* Warna teks default */
}
```

`body`: Aturan untuk elemen `<body>` dari halaman web.

background-color: #588dcd;; Mengatur warna latar belakang dari halaman menjadi biru (hex #588dcd).

color: #343a40;; Menentukan warna teks default (gelap) untuk semua elemen di dalam <body> yang tidak memiliki warna teks yang diatur secara eksplisit.

```
.card {  
  border-radius: 10px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
  background-color: rgba(255, 255, 255, 0.9); /* Warna latar belakang  
putih dengan transparansi */  
}
```

.card: Aturan untuk elemen dengan kelas card, sering digunakan untuk menampilkan informasi dalam bentuk kotak.

border-radius: 10px;; Mengatur sudut kotak agar melengkung dengan jari-jari 10 piksel, memberikan tampilan yang lebih lembut.

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);: Menambahkan bayangan halus di sekitar elemen card untuk memberikan efek kedalaman. Nilai 0 0 10px menunjukkan bahwa bayangan tidak terdistribusi secara horizontal atau vertikal, tetapi memiliki kekaburan sebesar 10 piksel dengan warna hitam (RGBA dengan alpha 0.1 memberikan transparansi).

background-color: rgba(255, 255, 255, 0.9);: Mengatur warna latar belakang card menjadi putih dengan transparansi 90% (10% transparan), sehingga terlihat lebih lembut di atas warna latar belakang biru.

```
h1 {  
  color: #ffffff; /* Warna teks judul menjadi putih untuk kontras yang  
lebih baik */  
}
```

h1: Aturan untuk elemen heading pertama (<h1>).

color: #ffffff;; Mengatur warna teks judul menjadi putih. Ini memberikan kontras yang baik dengan latar belakang biru, membuat teks lebih mudah dibaca.

```
.btn {  
  border-radius: 5px;  
}
```

.btn: Aturan untuk elemen dengan kelas btn, biasanya digunakan untuk tombol.

border-radius: 5px;; Mengatur sudut tombol agar melengkung dengan jari-jari 5 piksel, memberikan tampilan yang lebih halus dan modern.



