**Pertemuan 4 : Java Script Part 2**

**Tugas 4 : PopUp**

|  |  |
| --- | --- |
| **Nama / Nim** | **: Indratma Rakha Putra (2218129)** |
| **Kelas** | **: F** |
| **Pengajar** | **:** **AHMAD RIZKY RAMADANI** **(2218103)** |

* 1. **Tujuan**

1. Mampu memahami dasar – dasar Javascript.
2. Mampu memahami penggunaan Javascript.
3. Mampu memahami DOM pada Javascript.
   1. **Alat dan Bahan**
4. Laptop/pc
5. Modul Praktikum Web 2024
6. TextEditor (Visual Studio Code / Sublime Text)
   1. **Langkah-langkah**
7. PopUp

Source code html :

|  |
| --- |
| <div id="scrollable-popup" class="popup">  <div class="popup-content">  <h2>Daftar Item</h2>  <ul class="scrollable-list">  <!-- Tambahkan item di sini -->  <li><input type="checkbox" class="item-checkbox" value="Item 1"> Item 1</li>  <li><input type="checkbox" class="item-checkbox" value="Balonku"> Balonku</li>  <li><input type="checkbox" class="item-checkbox" value="Pelangi"> Pelangi</li>  <li><input type="checkbox" class="item-checkbox" value="Puncak Gunung"> Puncak Gunung</li>  <li><input type="checkbox" class="item-checkbox" value="Kasih Ibu"> Kasih Ibu</li>  <li><input type="checkbox" class="item-checkbox" value="Gundul Pacul"> Gundul Pacul</li>  <li><input type="checkbox" class="item-checkbox" value="Ampar-Ampar Pisang"> Ampar-Ampar Pisang</li>  <li><input type="checkbox" class="item-checkbox" value="Indonesia Raya"> Indonesia Raya</li>  <li><input type="checkbox" class="item-checkbox" value="Unknow"> Unknow</li>  <li><input type="checkbox" class="item-checkbox" value="Unknow"> Unknow</li>  <!-- Anda dapat menambahkan lebih banyak item -->  </ul>  <button class="btn">Play</button>  <button type="button" class="popup-close" onclick="closePopup()">Tutup</button>  </div>  </div> |

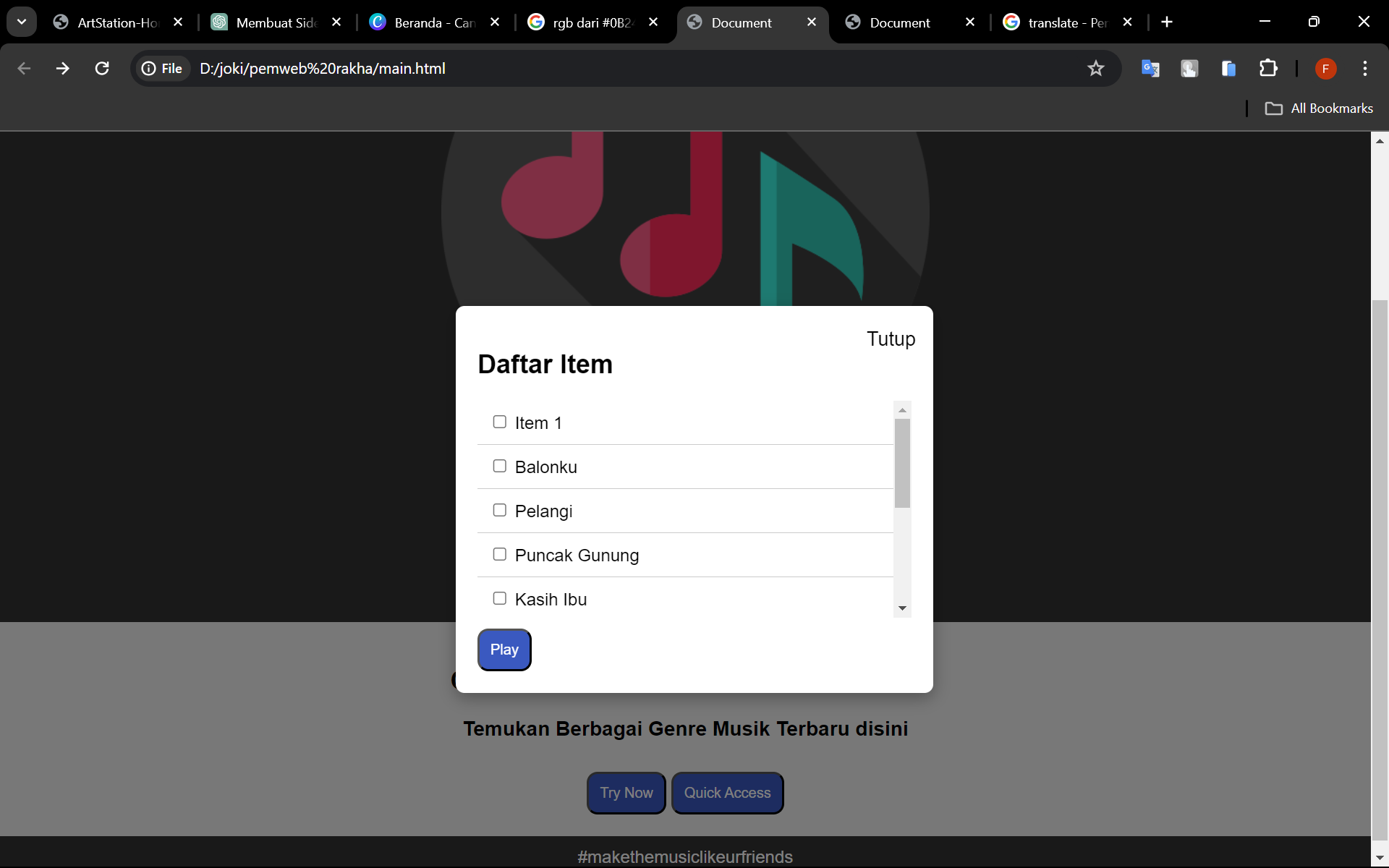
Source code css :

|  |
| --- |
| /\* Popup \*/  .popup {  display: none; /\* Tersembunyi secara default \*/  position: fixed;  top: 0;  left: 0;  width: 100vw;  height: 100vh;  background-color: rgba(0, 0, 0, 0.5); /\* Latar belakang dengan transparansi \*/  justify-content: center;  align-items: center;  z-index: 1000;  }  /\* Popup konten \*/  .popup-content {  background-color: white;  padding: 20px;  border-radius: 8px;  width: 400px;  position: relative;  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);  }  /\* Scrollable list \*/  .scrollable-list {  max-height: 200px; /\* Batasi tinggi untuk membuat scroll \*/  overflow-y: auto; /\* Aktifkan scrollbar vertikal \*/  padding: 0;  list-style-type: none;  margin: 0;  }  .scrollable-list li {  padding: 10px;  border-bottom: 1px solid #ccc;  }  .popup-close {  position: absolute;  top: 10px;  right: 10px;  border: none;  background: none;  font-size: 18px;  cursor: pointer;  }  button {  margin-top: 10px;  } |

Source code javascript :

|  |
| --- |
| // Mendapatkan elemen tombol untuk membuka popup  const openPopupBtn = document.getElementById("open-popup-btn");  // Mendapatkan elemen popup  const scrollablePopup = document.getElementById("scrollable-popup");  // Mendapatkan tombol "Play"  const playButton = document.querySelector('.popup-content .btn');  // Fungsi untuk membuka popup  function openPopup() {  scrollablePopup.style.display = 'flex';  }  // Fungsi untuk menutup popup  function closePopup() {  scrollablePopup.style.display = 'none';  }  // Fungsi asynchronous untuk menangani tindakan ketika tombol "Play" diklik  // Fungsi asynchronous untuk menangani tindakan ketika tombol "Play" diklik  async function handlePlayButtonClick() {  const checkboxes = document.querySelectorAll('.item-checkbox');  // Kumpulkan nilai-nilai yang dipilih  const selectedValues = [];  checkboxes.forEach(checkbox => {  if (checkbox.checked) {  selectedValues.push(checkbox.value);  }  });  try {  // Simulasikan tindakan yang dilakukan oleh tombol "Play" menggunakan promise  await playMedia();  // Tampilkan pesan sukses  alert(`Berhasil Memutar: ${selectedValues.join(', ')}`);  // Setelah pesan alert ditampilkan, tutup pop-up  closePopup();  // Opsi: Anda juga bisa melakukan refresh halaman setelah alert, jika ingin  // location.reload();  } catch (error) {  // Tangani kesalahan jika ada  alert(`Terjadi kesalahan: ${error.message}`);  }  }  // Menambahkan event listener ke tombol "Play"  playButton.addEventListener("click", handlePlayButtonClick);  // Menambahkan event listener ke tombol untuk membuka popup  openPopupBtn.addEventListener("click", openPopup);  // Menambahkan event listener ke tombol tutup untuk menutup popup  const closeButton = document.querySelector(".popup-close");  closeButton.addEventListener("click", closePopup); |

Hasil Tampilan :



Gambar 4.1 Tampilan PopUp

Analisa :

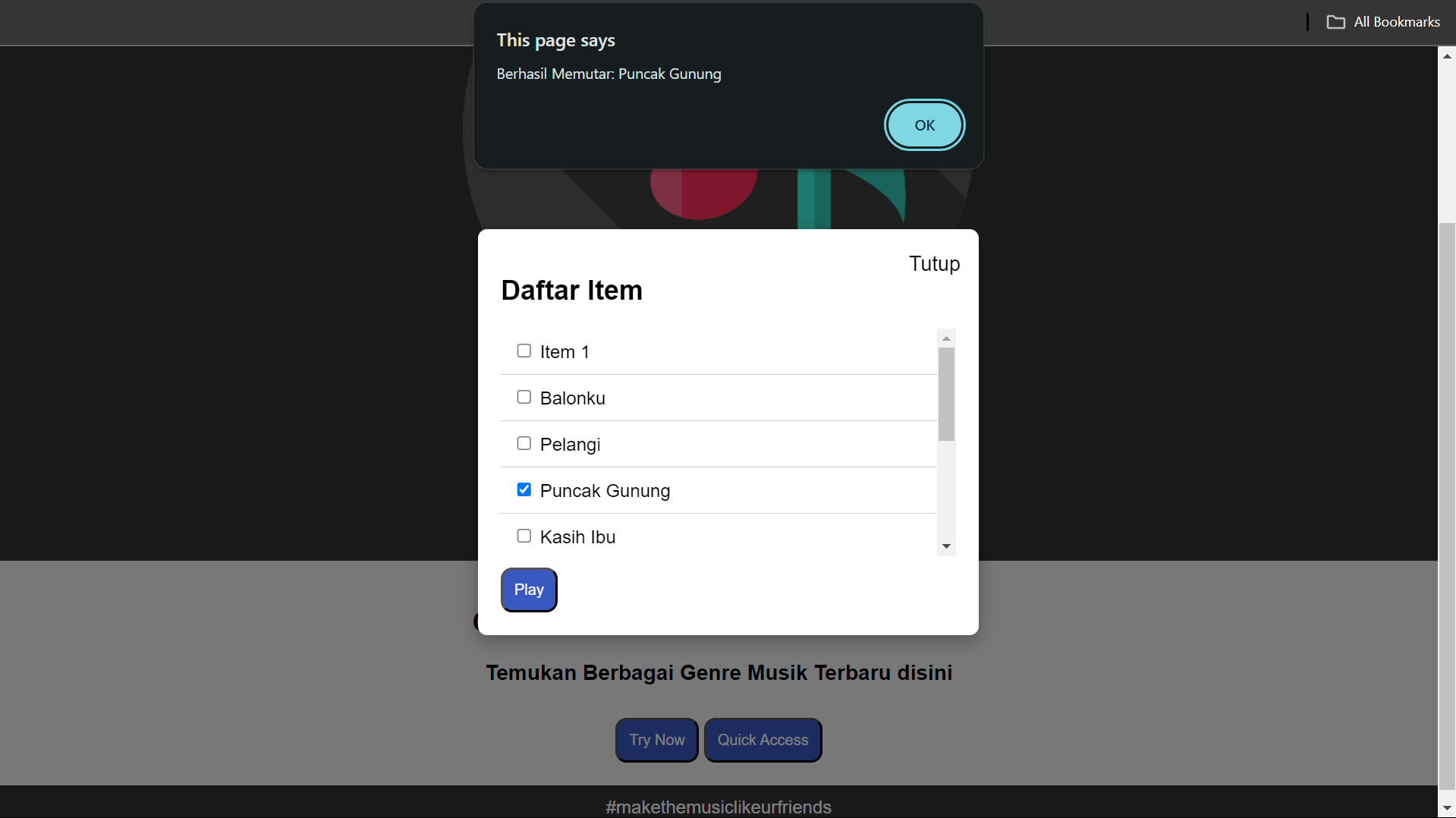
Kode di atas mengatur interaksi dengan pop-up dan tombol "Play" pada halaman web. Ini mencakup fungsi untuk membuka dan menutup pop-up berdasarkan elemen yang diidentifikasi dengan ID tertentu. Fungsi handlePlayButtonClick adalah fungsi asynchronous yang menangani tindakan saat tombol "Play" diklik. Fungsi ini mengumpulkan nilai-nilai dari checkbox yang dipilih, kemudian mensimulasikan tindakan play media menggunakan promise (playMedia()), menampilkan alert dengan daftar nilai yang dipilih, dan menutup pop-up setelah itu. Jika terjadi kesalahan, pesan kesalahan akan ditampilkan di alert. Event listener juga ditambahkan ke tombol "Play", tombol buka pop-up, dan tombol tutup pop-up untuk menangani klik pada elemen-elemen tersebut dengan fungsi terkait. Kode ini menggabungkan asynchronous, promise, dan DOM manipulation untuk mengatur interaksi pada halaman web dengan pop-up dan tombol.

1. Asynchronous dan Promise

Source code :

|  |
| --- |
| // Fungsi promise untuk mensimulasikan tindakan play media atau tugas tertentu  function playMedia() {  return new Promise((resolve, reject) => {  // Simulasikan waktu tunggu (misalnya, 2 detik) untuk menjalankan tindakan  setTimeout(() => {  // Misalnya, simulasi tindakan berhasil  const isSuccess = true; // Ubah ini sesuai kondisi yang diinginkan  if (isSuccess) {  resolve("Media berhasil diputar!"); // Simulasi hasil sukses  } else {  reject(new Error("Kesalahan saat memutar media.")); // Simulasi kesalahan  }  }, 2000);  });  } |

Hasil Tampilan :



Gambar 2.4 Tampilan Asynchronous dan Promise

Analisa :

Fungsi playMedia adalah fungsi promise yang mensimulasikan tindakan play media atau tugas tertentu dengan mengatur waktu tunggu selama 2 detik menggunakan setTimeout. Setelah waktu tunggu selesai, fungsi mengevaluasi kondisi isSuccess untuk menentukan apakah tindakan berhasil atau gagal. Jika isSuccess bernilai benar, fungsi memanggil resolve dengan pesan sukses "Media berhasil diputar!" untuk menyelesaikan promise. Jika isSuccess bernilai salah, fungsi memanggil reject dengan objek Error dan pesan kesalahan "Kesalahan saat memutar media." untuk menandakan kegagalan promise. Fungsi ini memberi fleksibilitas untuk mensimulasikan hasil sukses atau gagal dalam waktu tunggu tertentu, sesuai kebutuhan pengguna.