**Soal Studi Kasus (TODO LIST)**

1. Buatlah sebuah form input yang memungkinkan karyawan untuk memasukkan tugas baru.



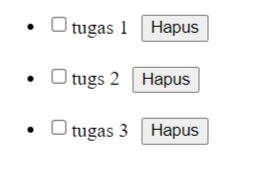
<form name="myForm" id="todoForm">

    <input placeholder="masukkan tugas" id="todo" name="fname" required>

    <button type="button" onclick="AddTodo()">Tambah</button>

  </form>

1. Tampilkan daftar tugas yang sudah ditambahkan dengan menggunakan elemen HTML yang sesuai.



function AddTodo() {

  var item = document.getElementById("todo").value

  if (item === "") {

    alert("input text tidak boleh kosong!");

  } else {

    var text = document.createTextNode(item);

    var newItem = document.createElement("li");

    wrap = document.getElementById("todoList");

    wrap.appendChild(newItem);

    var checkBox = document.createElement("input");

    checkBox.type = "checkbox";

    checkBox.onclick = checkedTodo.bind(checkBox);

    var btn = document.createElement("button");

    btn.style.margin = "10px"

    btn.textContent = "Hapus"

    btn.addEventListener('click', () => newItem.remove());

    newItem.appendChild(checkBox);

    newItem.appendChild(text);

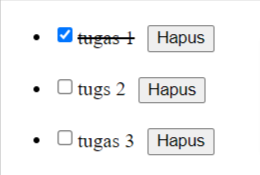
    newItem.appendChild(btn);

    document.getElementById("todo").value = "";

  }

}

1. Berikan opsi untuk menandai tugas sebagai selesai atau belum selesai.



function checkedTodo() {

    console.log("unchecked", unchecked())

    if (this.checked) {

      this.parentNode.style.textDecoration = "line-through";

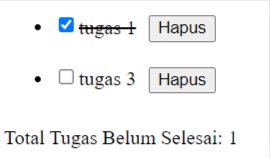
    } else {

      this.parentNode.style.textDecoration = "none";

    }

  }

1. Tambahkan fitur penghapusan tugas dari daftar.



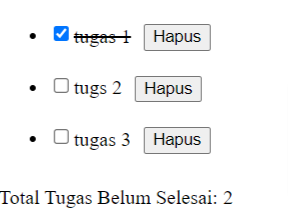
var btn = document.createElement("button");

    btn.style.margin = "10px"

    btn.textContent = "Hapus"

    btn.addEventListener('click', () => newItem.remove());

1. Tampilkan jumlah total tugas yang belum selesai.



 function unchecked() {

    var checkboxes = document.querySelectorAll('#todoList input[type="checkbox"]');

    var uncheckedTodo = 0;

    checkboxes.forEach(function (checkbox) {

      if (!checkbox.checked) {

        uncheckedTodo++;

      }

    });

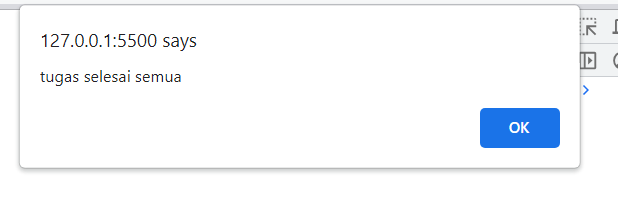
    document.getElementById("total-uncheced").innerHTML = `<p>

        Total Tugas Belum Selesai: ${uncheckedTodo}

      </p>`

  }

1. Tampilkan pesan atau notifikasi jika karyawan telah menyelesaikan semua tugas.



 document.getElementById("total-uncheced").innerHTML = `<p>

        Total Tugas Belum Selesai: ${uncheckedTodo}

      </p>`

  if (uncheckedTodo === 0) alert("tugas selesai semua")

1. Buatlah tombol untuk menghapus semua tugas sekaligus.



var btnremove = document.getElementById("remove-all")

var todoList = document.getElementById("todoList")

btnremove.addEventListener('click', function () {

  while (todoList.firstChild) {

    todoList.removeChild(todoList.firstChild);

  }

})

**Soal Essay**

1. Jelaskan perbedaan antara "bind()", "call()", dan "apply()" dalam konteks pengaturan nilai "this" dalam fungsi.

Bind(): mengembalikan salinan fungsi dengan konteks tertentu.

Call(): mengembalikan salinan fungsi dengan konteks tertentu dan argument yang diteruskan secara terpisah sebagai parameter fungsi.

Apply(): fungsinya sperti call tetapi argument yang diteruskan dalam bentuk array.

1. Soal: 2. Apa itu "memoization" dalam konteks pemrograman JavaScript? Jelaskan bagaimana memoization dapat meningkatkan performa fungsi.

Memorization memungkinkan untuk mempercepat pemanggilan fungsi dengan meng-cache nilai kembalian yang sama. Memoisasi menghindari perhitungan ulang yang tidak perlu dengan menggunakan hasil yang telah disimpan sebelumnya.

1. Soal: 3. Jelaskan perbedaan antara "synchronous" dan "asynchronous" dalam konteks pemrograman JavaScript.

Synchronous proses eksekusi perintah kode secara bersamaan sedangkan asynchronous akan mengeksekusi perintah setelah perintah sebelumnya terpenuhi.

1. Soal: 4. Apa itu "generator" dalam JavaScript? Jelaskan bagaimana cara kerja generator.

Fungsi khusus untuk memberhentikan sementara dan dilanjutkan kemudian.

Generator menggunakan kata kunci **function\*** dan menggunakan pernyataan **yield**

Untuk mengeluarkan nilai. Pernyataan **yield** untuk mengeluarkan nilai secara berurutan. Setiap kali memanggil **generator.next()** generator akan berhenti setelah pernyataan **yield** terakhir dan akan melanjutkan eksekusi ketika dipanggil kembali.

1. Soal: 5. Jelaskan perbedaan antara "Promise" dan "async/await" dalam JavaScript.

Promise menggunakan metode “then()” dan “catch()” untuk menangani hasil promise sedangkan async/await menggunakan “await” untuk menunggu hasil promise dan menambahkan “async” sebelum nama function