**SOAL ESSAY**

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

**Jawab :**

bind() : Mengembalikan fungsi baru dengan this yang secara permanen diikat ke nilai yang diberikan.

call() : Langsung menjalankan fungsi dengan this yang ditentukan serta menerima argumen tambahan secara terpisah.

apply() : Serupa dengan call(), namun argumen tambahan diterima dalam bentuk array.

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

**Jawab :**

teknik optimisasi yang digunakan untuk meningkatkan performa fungsi dengan menyimpan hasil dari perhitungan fungsi yang mahal (intensif). Jika fungsi dipanggil lagi dengan argumen yang sama, hasil yang disimpan dapat digunakan kembali tanpa harus melakukan perhitungan ulang.

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

**Jawab :**

Synchronous Operasi dieksekusi satu per satu secara berurutan. Setiap perintah harus selesai sebelum perintah berikutnya dieksekusi.

Asynchronous Operasi dapat dimulai dan dijalankan di latar belakang, memungkinkan program melanjutkan eksekusi tanpa menunggu operasi tersebut selesai.

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

**Jawab :**

**Generator** adalah fungsi khusus yang dapat menghentikan dan melanjutkan eksekusinya.

Cara Kerja Generator adalah Ketika dipanggil, generator tidak langsung mengeksekusi fungsinya, melainkan mengembalikan objek iterator dan Setiap kali metode next() dipanggil pada objek generator, eksekusi dilanjutkan hingga menemukan yield.

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

**Jawab :**

Promise menggunakan .then() dan .catch() untuk menangani hasil atau error dan async/await menyediakan cara yang lebih mudah dan bersih untuk menangani Promise, membuat kode asynchronous terlihat seperti kode synchronous.

**SOAL STUDI KASUS**

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

A screenshot of a computer

Description automatically generated

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

A black and white check box

Description automatically generated

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

A close up of a check box

Description automatically generated

1. Tambahkan fitur penghapusan tugas dari daftar.

A close up of a sign

Description automatically generated with medium confidence

1. Tampilkan jumlah total tugas yang belum selesai.

A screenshot of a to do list

Description automatically generated

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

A black and white screen

Description automatically generated

1. Buatlah tombol untuk menghapus semua tugas sekaligus.

A screenshot of a computer

Description automatically generated