**Menambah Elemen Event - Event Listener**

Kamu sudah berhasil membuat elemen HTML menggunakan JavaScript pada topik sebelumnya. Pada topik ini, kita akan membahas cara membuat elemen *event listener* pada HTML melalui JavaScript. *Event listener* adalah metode JavaScript yang memungkinkan kita untuk memasukkan suatu *event* ke dalam suatu elemen tanpa menghapus/menghilangkan *event* yang sudah ada sebelumnya.

*Syntax* dalam menggunakan *event listener* adalah:

element.addEventListener(event, function, useCapture);

1. ***event*** (**required**) adalah tipe *event* yang kita butuhkan seperti click, mouseover, atau HTML DOM *event* lainnya.
2. ***function*** (**required**) adalah fungsi yang kita panggil saat *event* sedang terjadi.
3. ***useCapture*** (**optional**) merupakan nilai *boolean* yang menentukan apakah *event* dieksekusi secara *bubbling* (\*event element **parent** \* dieksekusi **setelah** \* event element **child** \* dieksekusi) atau *capturing* (\*event element **child** \* dieksekusi **setelah** \*event element **parent** \* dieksekusi).

Contoh yang akan kita gunakan adalah membuat elemen div yang akan dibentuk menjadi sebuah kotak, kemudian kotak ini memiliki *event listener*. Saat kotak ini dilakukan *hover* oleh *mouse* pengguna (*mouse* diarahkan ke kotak), maka akan tampil *listener* berupa tampilan kotak yang berubah **warna** dan **teks**.

Berikut ini langkah-langkah dalam membuat *event listener* menggunakan DOM pada JavaScript:

**Langkah 1**: Kita akan membuat *file* bernama index.html yang berisi kode seperti berikut ini:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>DOM - Event Elemen</title>

</head>

<body>

<div id="test-event" style="background-color:pink;width:120px;height:20px;padding:40px;">

Mouse Over Me!

</div>

<script src="elemen.js"></script>

</body>

</html>

dari kode HTML tersebut kita akan mendapatkan tampilan seperti ini:



Lalu dari tampilan ini kita hendak mengganti teks dan warna *background* ketika *mouse* melakukan *hover* ke kotak div berwarna pink tersebut.

**Langkah 2**: Kita akan membuat *file* bernama elemen.js. Lalu tambahkan kode berikut ini untuk mendapatkan elemen div dengan id bernama test-event:

const boxPink = document.getElementById("test-event");

**Langkah 3**: Buat fungsi untuk mengganti teks dan warna *background*:

function mouseOver() {

boxPink.innerHTML = "Hai!"; // mengganti teks ketika mouse diarahkan ke dalam box

boxPink.style.backgroundColor = "blue"; // mengganti warna background ketika mouse diarahkan ke dalam box

}

function mouseOut() {

boxPink.innerHTML = "Mouse Over Me!"; // mengganti teks ketika mouse diarahkan ke luar box

boxPink.style.backgroundColor = "pink"; // mengganti warna background ketika mouse diarahkan ke luar box

}

**Langkah 4**: Panggil fungsi pada langkah ketiga dengan method .addEventListener()

boxPink.addEventListener("mouseover", mouseOver);

boxPink.addEventListener("mouseout", mouseOut);

Maka dari langkah 1-4 jika dijalankan, hasilnya akan menjadi seperti berikut ini:



Menarik bukan? dengan *event listener* kamu dapat menambahkan interaksi yang lebih menarik antara pengguna dan aplikasi kamu.

📝**Catatan**:

Program di atas dijalankan menggunakan *web browser* Chrome

# Menghapus Elemen Event - Event Listener

Pada topik sebelumnya kita telah belajar menambahkan event listener ke dalam sebuah elemen HTML. Lalu bagaimana cara menghapus event listener yang telah kita tambahkan?

Caranya mudah, kita cukup mengganti .addEventListener() menjadi removeEventListener(). Berikut struktur kode untuk method removeEventListener():

element.removeEventListener(event, function, useCapture);

Pada topik sebelumnya, kita telah membuat sebuah kotak yang jika di-hover oleh mouse, maka akan berubah warna background dan konten teksnya. Lalu bagaimana cara kita menghapus event listener tersebut agar kotak tidak lagi berubah warna background dan konten teksnya saat di-hover?

Berikut langkah-langkah solusi untuk menghapus event listener:

**Langkah 1**: Kita akan membuat elemen button di bawah kotak pink pada kode HTML yang telah kita buat di topik sebelumnya, seperti berikut ini:

<button onclick="removeHandler()">Remove Event</button>

Sehingga file index.html akan menjadi seperti ini:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>DOM - Event Elemen</title>

</head>

<body>

<div id="test-event" style="background-color:pink;width:120px;height:20px;padding:40px;">

Mouse Over Me!

</div>

<button onclick="removeHandler()">Remove Event</button>

<script src="elemen.js"></script>

</body>

</html>

dari kode HTML tersebut kita akan mendapatkan tampilan seperti ini:



**Langkah 2**: Kita akan menambahkan fungsi removeHandler() dalam file elemen.js yang telah kita buat pada topik sebelumnya, seperti berikut ini:

function removeHandler () {

boxPink.removeEventListener("mouseover", mouseOver);

boxPink.removeEventListener("mouseout", mouseOut);

}

Sehingga file elemen.js akan menjadi seperti ini:

const boxPink = document.getElementById("test-event")

function mouseOver() {

boxPink.innerHTML = "Hai!"; // mengganti text ketika mouse diarahkan ke dalam box

boxPink.style.backgroundColor = "blue"; // mengganti warna background ketika mouse diarahkan ke dalam box

}

function mouseOut() {

boxPink.innerHTML = "Mouse Over Me!"; // mengganti text ketika mouse diarahkan ke luar box

boxPink.style.backgroundColor = "pink"; // mengganti warna background ketika mouse diarahkan ke luar box

}

boxPink.addEventListener("mouseover", mouseOver);

boxPink.addEventListener("mouseout", mouseOut);

function removeHandler () {

boxPink.removeEventListener("mouseover", mouseOver);

boxPink.removeEventListener("mouseout", mouseOut);

}

Maka jika index.html ini kita jalankan, maka akan terlihat seperti berikut ini:



Dari gambar di atas, dapat kita lihat bahwa sebelum button Remove Event di-click, kotak pink masih berubah warna latar dan teks, namun setelah button Remove Event di-click, maka kotak tidak lagi berubah warna dan teks.