

# Daftar Event pada addEventListener

## Mouse Events

### click

Penjelasan:

Terpicu saat pengguna mengklik elemen dengan tombol kiri mouse.

**Sintaks:**

```
element.addEventListener("click", function);
```

**Contoh:**

```
document.getElementById("btn").addEventListener("click", function() {  
    alert("Tombol diklik!");  
});
```

### dblclick

Penjelasan:

Terpicu saat pengguna mengklik elemen dua kali secara cepat.

**Sintaks:**

```
element.addEventListener("dblclick", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("dblclick", function() {  
    this.style.backgroundColor = "blue";  
});
```

### mousedown

Penjelasan:

Terpicu saat tombol mouse ditekan (belum dilepas) pada elemen.

**Sintaks:**

```
element.addEventListener("mousedown", function);
```

**Contoh:**

```
document.addEventListener("mousedown", function() {  
    console.log("Mouse ditekan");  
});
```

### mouseup

Penjelasan:

Terpicu saat tombol mouse dilepas setelah ditekan.

**Sintaks:**

```
element.addEventListener("mouseup", function);
```

**Contoh:**

```
document.addEventListener("mouseup", function() {  
    console.log("Mouse dilepas");  
});
```

## **mousemove**

Penjelasan:

Terpicu saat mouse digerakkan di atas elemen.

### **Sintaks:**

```
element.addEventListener("mousemove", function);
```

### **Contoh:**

```
document.addEventListener("mousemove", function(e) {  
    console.log(`X: ${e.clientX}, Y: ${e.clientY}`);  
});
```

## **mouseenter**

Penjelasan:

Terpicu saat mouse masuk ke elemen (tidak bubbling).

### **Sintaks:**

```
element.addEventListener("mouseenter", function);
```

### **Contoh:**

```
document.getElementById("hoverBox").addEventListener("mouseenter", function() {  
    this.style.border = "2px solid green";  
});
```

## **mouseleave**

Penjelasan:

Terpicu saat mouse keluar dari elemen (tidak bubbling).

### **Sintaks:**

```
element.addEventListener("mouseleave", function);
```

### **Contoh:**

```
document.getElementById("hoverBox").addEventListener("mouseleave", function() {  
    this.style.border = "none";  
});
```

## **mouseover**

Penjelasan:

Terpicu saat mouse masuk ke elemen atau elemen anaknya (bubbling).

### **Sintaks:**

```
element.addEventListener("mouseover", function);
```

### **Contoh:**

```
document.getElementById("menu").addEventListener("mouseover", function() {
```

```
    console.log("Mouse masuk menu");  
});
```

## mouseout

Penjelasan:

Terpicu saat mouse keluar dari elemen atau elemen anaknya (bubbling).

### Sintaks:

```
element.addEventListener("mouseout", function);
```

### Contoh:

```
document.getElementById("menu").addEventListener("mouseout", function() {  
    console.log("Mouse keluar dari menu");  
});
```

## contextmenu

Penjelasan:

Terpicu saat klik kanan dilakukan pada elemen.

### Sintaks:

```
element.addEventListener("contextmenu", function);
```

### Contoh:

```
document.addEventListener("contextmenu", function(e) {  
    e.preventDefault();  
    alert("Klik kanan dinonaktifkan");  
});
```

## Keyboard Events

## keydown

Penjelasan:

Terpicu saat tombol keyboard ditekan (saat tombol mulai ditekan).

### Sintaks:

```
element.addEventListener("keydown", function);
```

### Contoh:

```
document.addEventListener("keydown", function(e) {  
    console.log("Tombol ditekan:", e.key);  
});
```

## keyup

Penjelasan:

Terpicu saat tombol keyboard dilepas setelah ditekan.

### Sintaks:

```
element.addEventListener("keyup", function);
```

**Contoh:**

```
document.addEventListener("keyup", function(e) {  
    console.log("Tombol dilepas:", e.key);  
});
```

**keypress**

Penjelasan:

Terpicu saat tombol ditekan dan menghasilkan karakter. Sudah deprecated.

**Sintaks:**

```
element.addEventListener("keypress", function);
```

**Contoh:**

```
document.addEventListener("keypress", function(e) {  
    console.log("Karakter ditekan:", e.key);  
});
```

## Form & Input Events

**input**

Penjelasan:

Terpicu setiap kali nilai input berubah (real-time saat mengetik).

**Sintaks:**

```
element.addEventListener("input", function);
```

**Contoh:**

```
document.getElementById("textInput").addEventListener("input", function(e) {  
    console.log("Nilai sekarang:", e.target.value);  
});
```

**change**

Penjelasan:

Terpicu saat nilai input selesai diubah (biasanya saat kehilangan fokus atau setelah dipilih).

**Sintaks:**

```
element.addEventListener("change", function);
```

**Contoh:**

```
document.getElementById("selectOption").addEventListener("change", function(e) {  
    console.log("Pilihan berubah:", e.target.value);  
});
```

**submit**

Penjelasan:

Terpicu saat form dikirim (submit).

**Sintaks:**

```
formElement.addEventListener("submit", function);
```

**Contoh:**

```
document.getElementById("myForm").addEventListener("submit", function(e) {  
    e.preventDefault();  
    alert("Form dikirim!");  
});
```

**reset**

Penjelasan:

Terpicu saat form di-reset ke nilai awal.

**Sintaks:**

```
formElement.addEventListener("reset", function);
```

**Contoh:**

```
document.getElementById("myForm").addEventListener("reset", function() {  
    alert("Form direset!");  
});
```

**focus**

Penjelasan:

Terpicu saat elemen mendapatkan fokus. Tidak bubble.

**Sintaks:**

```
element.addEventListener("focus", function);
```

**Contoh:**

```
document.getElementById("textInput").addEventListener("focus", function() {  
    this.style.backgroundColor = "#e0f7fa";  
});
```

**blur**

Penjelasan:

Terpicu saat elemen kehilangan fokus. Tidak bubble.

**Sintaks:**

```
element.addEventListener("blur", function);
```

**Contoh:**

```
document.getElementById("textInput").addEventListener("blur", function() {  
    this.style.backgroundColor = "";  
});
```

**focusin**

Penjelasan:

Terpicu saat fokus masuk ke elemen. Bisa bubble.

**Sintaks:**

```
element.addEventListener("focusin", function);
```

**Contoh:**

```
document.getElementById("formWrapper").addEventListener("focusin", function() {  
    console.log("Fokus masuk ke dalam form");  
});
```

**focusout**

Penjelasan:

Terpicu saat fokus keluar dari elemen. Bisa bubble.

**Sintaks:**

```
element.addEventListener("focusout", function);
```

**Contoh:**

```
document.getElementById("formWrapper").addEventListener("focusout", function() {  
    console.log("Fokus keluar dari form");  
});
```

## Drag & Drop Events

**drag**

Penjelasan:

Terpicu saat elemen sedang di-drag.

**Sintaks:**

```
element.addEventListener("drag", function);
```

**Contoh:**

```
document.getElementById("draggable").addEventListener("drag", function() {  
    console.log("Sedang di-drag...");  
});
```

**dragstart**

Penjelasan:

Terpicu saat drag dimulai.

**Sintaks:**

```
element.addEventListener("dragstart", function);
```

**Contoh:**

```
document.getElementById("draggable").addEventListener("dragstart", function(e) {  
    e.dataTransfer.setData("text/plain", e.target.id);  
    console.log("Drag dimulai");  
});
```

**dragend**

Penjelasan:

Terpicu saat drag selesai, baik berhasil atau dibatalkan.

**Sintaks:**

```
element.addEventListener("dragend", function);
```

**Contoh:**

```
document.getElementById("draggable").addEventListener("dragend", function() {  
    console.log("Drag selesai");  
});
```

## dragenter

Penjelasan:

Terpicu saat elemen draggable memasuki area target drop.

**Sintaks:**

```
element.addEventListener("dragenter", function);
```

**Contoh:**

```
document.getElementById("dropzone").addEventListener("dragenter", function() {  
    this.style.backgroundColor = "#e0f7fa";  
});
```

## dragleave

Penjelasan:

Terpicu saat elemen draggable keluar dari area drop.

**Sintaks:**

```
element.addEventListener("dragleave", function);
```

**Contoh:**

```
document.getElementById("dropzone").addEventListener("dragleave", function() {  
    this.style.backgroundColor = "";  
});
```

## dragover

Penjelasan:

Terpicu saat elemen draggable berada di atas area drop.

**Sintaks:**

```
element.addEventListener("dragover", function);
```

**Contoh:**

```
document.getElementById("dropzone").addEventListener("dragover", function(e) {  
    e.preventDefault();  
});
```

## drop

Penjelasan:

Terpicu saat elemen draggable dilepaskan di dalam area drop.

**Sintaks:**

```
element.addEventListener("drop", function);
```

**Contoh:**

```
document.getElementById("dropzone").addEventListener("drop", function(e) {  
    e.preventDefault();  
    const data = e.dataTransfer.getData("text/plain");  
    const dragged = document.getElementById(data);  
    this.appendChild(dragged);  
});
```

## Animation & Transition Events

### animationstart

Penjelasan:

Terpicu saat animasi CSS dimulai.

**Sintaks:**

```
element.addEventListener("animationstart", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("animationstart", function() {  
    console.log("Animasi dimulai");  
});
```

### animationend

Penjelasan:

Terpicu saat animasi CSS selesai.

**Sintaks:**

```
element.addEventListener("animationend", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("animationend", function() {  
    console.log("Animasi selesai");  
});
```

### animationiteration

Penjelasan:

Terpicu setiap kali animasi CSS mengulang.

**Sintaks:**

```
element.addEventListener("animationiteration", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("animationiteration", function()  
{  
    console.log("Animasi diulang");  
});
```

### transitionstart



Penjelasan:

Terpicu saat transisi CSS dimulai.

**Sintaks:**

```
element.addEventListener("transitionstart", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("transitionstart", function() {  
    console.log("Transisi dimulai");  
});
```

### **transitionend**

Penjelasan:

Terpicu saat transisi CSS selesai.

**Sintaks:**

```
element.addEventListener("transitionend", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("transitionend", function() {  
    console.log("Transisi selesai");  
});
```

### **transitionrun**

Penjelasan:

Terpicu saat transisi mulai berjalan.

**Sintaks:**

```
element.addEventListener("transitionrun", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("transitionrun", function() {  
    console.log("Transisi mulai berjalan");  
});
```

### **transitioncancel**

Penjelasan:

Terpicu saat transisi dibatalkan sebelum selesai.

**Sintaks:**

```
element.addEventListener("transitioncancel", function);
```

**Contoh:**

```
document.getElementById("box").addEventListener("transitioncancel", function() {  
    console.log("Transisi dibatalkan");  
});
```

## **Touch Events (Mobile)**

## touchstart

Penjelasan:

Terpicu saat satu atau lebih jari mulai menyentuh layar.

### Sintaks:

```
element.addEventListener("touchstart", function);
```

### Contoh:

```
document.getElementById("touchArea").addEventListener("touchstart", function(e) {
    console.log("Sentuhan dimulai:", e.touches.length, "jari");
});
```

## touchmove

Penjelasan:

Terpicu saat jari digerakkan di atas layar setelah touchstart.

### Sintaks:

```
element.addEventListener("touchmove", function);
```

### Contoh:

```
document.getElementById("touchArea").addEventListener("touchmove", function(e) {
    console.log("Sentuhan bergerak ke:", e.touches[0].clientX,
e.touches[0].clientY);
});
```

## touchend

Penjelasan:

Terpicu saat jari diangkat dari layar (mengakhiri sentuhan).

### Sintaks:

```
element.addEventListener("touchend", function);
```

### Contoh:

```
document.getElementById("touchArea").addEventListener("touchend", function(e) {
    console.log("Sentuhan selesai");
});
```

## touchcancel

Penjelasan:

Terpicu saat sistem menghentikan sentuhan (misalnya karena interupsi).

### Sintaks:

```
element.addEventListener("touchcancel", function);
```

### Contoh:

```
document.getElementById("touchArea").addEventListener("touchcancel", function() {
    console.log("Sentuhan dibatalkan oleh sistem");
});
```

```
});
```

## Media Events

### play

Penjelasan:

Terpicu saat media mulai diputar.

**Sintaks:**

```
mediaElement.addEventListener("play", function);
```

**Contoh:**

```
document.getElementById("video").addEventListener("play", function() {  
    console.log("Video mulai diputar");  
});
```

### pause

Penjelasan:

Terpicu saat media dijeda.

**Sintaks:**

```
mediaElement.addEventListener("pause", function);
```

**Contoh:**

```
document.getElementById("video").addEventListener("pause", function() {  
    console.log("Video dijeda");  
});
```

### ended

Penjelasan:

Terpicu saat media selesai diputar.

**Sintaks:**

```
mediaElement.addEventListener("ended", function);
```

**Contoh:**

```
document.getElementById("video").addEventListener("ended", function() {  
    alert("Video selesai diputar");  
});
```

### volumechange

Penjelasan:

Terpicu saat volume media diubah.

**Sintaks:**

```
mediaElement.addEventListener("volumechange", function);
```

**Contoh:**

```
document.getElementById("audio").addEventListener("volumechange", function() {
```

```
    console.log("Volume berubah:", this.volume);  
});
```

## timeupdate

Penjelasan:

Terpicu secara berkala saat waktu pemutaran media berubah.

### Sintaks:

```
mediaElement.addEventListener("timeupdate", function());
```

### Contoh:

```
document.getElementById("video").addEventListener("timeupdate", function() {  
    console.log("Waktu saat ini:", this.currentTime);  
});
```

## seeking

Penjelasan:

Terpicu saat pengguna mulai mencari posisi (seek) pada media.

### Sintaks:

```
mediaElement.addEventListener("seeking", function());
```

### Contoh:

```
document.getElementById("video").addEventListener("seeking", function() {  
    console.log("Pengguna sedang mencari posisi...");  
});
```

## seeked

Penjelasan:

Terpicu saat proses pencarian posisi (seek) selesai.

### Sintaks:

```
mediaElement.addEventListener("seeked", function());
```

### Contoh:

```
document.getElementById("video").addEventListener("seeked", function() {  
    console.log("Posisi baru ditemukan:", this.currentTime);  
});
```

## loadeddata

Penjelasan:

Terpicu saat data pertama media dimuat.

### Sintaks:

```
mediaElement.addEventListener("loadeddata", function());
```

### Contoh:

```
document.getElementById("video").addEventListener("loadeddata", function() {  
    console.log("Data media siap diputar");  
});
```

```
});
```

## loadedmetadata

Penjelasan:

Terpicu saat metadata media tersedia.

### Sintaks:

```
mediaElement.addEventListener("loadedmetadata", function);
```

### Contoh:

```
document.getElementById("video").addEventListener("loadedmetadata", function() {  
    console.log("Durasi video:", this.duration);  
});
```

## error

Penjelasan:

Terpicu saat terjadi kesalahan saat memuat media.

### Sintaks:

```
mediaElement.addEventListener("error", function);
```

### Contoh:

```
document.getElementById("video").addEventListener("error", function() {  
    console.error("Gagal memuat video");  
});
```

## Window & Document Events

## load

Penjelasan:

Terpicu saat seluruh halaman (termasuk gambar, skrip, dan style) selesai dimuat.

### Sintaks:

```
window.addEventListener("load", function);
```

### Contoh:

```
window.addEventListener("load", function() {  
    console.log("Halaman selesai dimuat");  
});
```

## resize

Penjelasan:

Terpicu saat ukuran jendela browser diubah.

### Sintaks:

```
window.addEventListener("resize", function);
```

### Contoh:

```
window.addEventListener("resize", function() {
```

```
console.log("Ukuran jendela:", window.innerWidth, "x", window.innerHeight);
});
```

## scroll

Penjelasan:

Terpicu saat pengguna menggulir (scroll) halaman atau elemen yang bisa di-scroll.

### Sintaks:

```
window.addEventListener("scroll", function);
```

### Contoh:

```
window.addEventListener("scroll", function() {
    console.log("Scroll posisi Y:", window.scrollY);
});
```

## beforeunload

Penjelasan:

Terpicu sebelum halaman ditutup atau dimuat ulang.

### Sintaks:

```
window.addEventListener("beforeunload", function);
```

### Contoh:

```
window.addEventListener("beforeunload", function(e) {
    e.preventDefault();
    e.returnValue = "";
});
```

## unload

Penjelasan:

Terpicu saat halaman ditutup.

### Sintaks:

```
window.addEventListener("unload", function);
```

### Contoh:

```
window.addEventListener("unload", function() {
    console.log("Halaman ditutup");
});
```

## error

Penjelasan:

Terpicu saat terjadi kesalahan runtime (bisa pada skrip, gambar, dsb.).

### Sintaks:

```
window.addEventListener("error", function);
```

### Contoh:

```
window.addEventListener("error", function(e) {
```

```
console.error("Kesalahan:", e.message);
});
```

## DOMContentLoaded

Penjelasan:

Terpicu saat struktur DOM selesai dimuat.

**Sintaks:**

```
document.addEventListener("DOMContentLoaded", function);
```

**Contoh:**

```
document.addEventListener("DOMContentLoaded", function() {
    console.log("DOM siap digunakan");
});
```

## Clipboard Events

### copy

Penjelasan:

Terpicu saat pengguna menyalin (copy) konten dari halaman.

**Sintaks:**

```
element.addEventListener("copy", function);
```

**Contoh:**

```
document.addEventListener("copy", function(e) {
    console.log("Konten disalin");
});
```

### cut

Penjelasan:

Terpicu saat pengguna memotong (cut) konten dari halaman.

**Sintaks:**

```
element.addEventListener("cut", function);
```

**Contoh:**

```
document.addEventListener("cut", function(e) {
    console.log("Konten dipotong");
});
```

### paste

Penjelasan:

Terpicu saat pengguna menempelkan (paste) konten ke dalam elemen.

**Sintaks:**

```
element.addEventListener("paste", function);
```

**Contoh:**

```
document.addEventListener("paste", function(e) {  
    const pasted = (e.clipboardData || window.clipboardData).getData("text");  
    console.log("Konten ditempelkan:", pasted);  
});
```

## Security & Network Events

### online

Penjelasan:

Terpicu saat browser kembali terhubung ke internet.

#### Sintaks:

```
window.addEventListener("online", function);
```

#### Contoh:

```
window.addEventListener("online", function() {  
    console.log("Kembali online");  
});
```

### offline

Penjelasan:

Terpicu saat koneksi internet terputus.

#### Sintaks:

```
window.addEventListener("offline", function);
```

#### Contoh:

```
window.addEventListener("offline", function() {  
    console.log("Koneksi terputus");  
});
```

### message

Penjelasan:

Terpicu saat menerima pesan dari iframe, worker, atau koneksi WebSocket.

#### Sintaks:

```
window.addEventListener("message", function);
```

#### Contoh:

```
window.addEventListener("message", function(e) {  
    console.log("Pesan diterima:", e.data);  
});
```

### popstate

Penjelasan:

Terpicu saat pengguna melakukan navigasi kembali/maju.

#### Sintaks:

```
window.addEventListener("popstate", function);
```



**Contoh:**

```
window.addEventListener("popstate", function(e) {  
  console.log("Navigasi kembali/maju:", e.state);  
});
```

## Custom Events

### CustomEvent

Penjelasan:

Event yang dibuat secara manual oleh developer menggunakan `new CustomEvent()`.

**Sintaks:**

```
const event = new CustomEvent("namaEvent", {  
  detail: { ...dataTambahkan }  
});  
element.dispatchEvent(event);  
element.addEventListener("namaEvent", function);
```

**Contoh:**

```
// Buat event kustom  
const myEvent = new CustomEvent("produkTerpilih", {  
  detail: {  
    id: 123,  
    nama: "Laptop"  
  }  
});  
  
// Dengarkan event  
document.addEventListener("produkTerpilih", function(e) {  
  console.log("Produk dipilih:", e.detail);  
});  
  
// Dispatch event  
document.dispatchEvent(myEvent);
```