

1. Script Hover animasi

```
C: > xampp > htdocs > animasi > hover.html > html > body > div.kotak
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Animasi</title>
7      <style>
8          .kotak {
9              width: 100px;
10             height: 100px;
11             background-color: lightblue;
12             margin: 20px;
13             transition-property: background-color, width, height; /* Properti yang dianimasikan */
14             transition-duration: 0.3s, 0.5s, 0.5s; /* Durasi untuk setiap properti */
15             transition-timing-function: ease-in-out; /* Kurva kecepatan */
16         }
17
18         .kotak:hover {
19             background-color: lightcoral;
20             width: 120px;
21             height: 120px;
22         }
23     </style>
24 </head>
25 <body>
26     <div class="kotak">Hover me!</div>
27 </body>
28 </html>
```

Penjelasan:

.kotak: Gaya awal untuk elemen kotak.

transition-property: Menentukan properti CSS mana yang akan dianimasikan. Di sini, kita menganimasikan background-color, width, dan height.

transition-duration: Menentukan berapa lama transisi akan berlangsung untuk setiap properti yang sesuai (dalam detik atau milidetik).

transition-timing-function: Menentukan kurva kecepatan transisi. ease-in-out membuat animasi berjalan lambat di awal dan di akhir, serta cepat di tengah. Pilihan lain termasuk linear, ease, ease-in, ease-out, dan cubic-bezier().

.kotak:hover: Gaya yang diterapkan saat kursor mouse berada di atas elemen. Perubahan pada properti yang didefinisikan dalam transition-property akan dianimasikan secara halus.

2. Scrip Opasitas

```
C: > xampp > htdocs > animasi > opasitas.html > html > body > p.teks
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Opasitas</title>
7      <style>
8          .teks {
9              opacity: 1;
10             transition: opacity 0.5s ease-in; /* Shorthand untuk semua properti dengan durasi dan
11         }
12
13         .teks:hover {
14             opacity: 0.5;
15         }
16     </style>
17 </head>
18 <body>
19     <p class="teks">Teks ini akan memudar saat di-hover.</p>
20 </body>
21 </html>
```

Penjelasan:

Di sini, kita menggunakan shorthand transition untuk menggabungkan semua properti transisi dalam satu baris.

3. Gerak Putar

```
C: > xampp > htdocs > animasi > <> gerakputar.html > html > body > div.lingkaran
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Gerak Putar</title>
7      <style>
8          .lingkaran {
9              width: 80px;
10             height: 80px;
11             background-color: purple;
12             border-radius: 50%;
13             opacity: 1;
14             transform: scale(1);
15             animation-name: memudarMembesar;
16             animation-duration: 1.5s;
17             animation-iteration-count: infinite;
18             animation-direction: alternate; /* Animasi berjalan maju lalu mundur */
19         }
20
21         @keyframes memudarMembesar {
22             0% {
23                 opacity: 1;
24                 transform: scale(1);
25             }
26             100% {
27                 opacity: 0.3;
28                 transform: scale(1.5);
29             }
30         }
31     </style>
32 </head>
33 <body>
    <div class="lingkaran"></div>
</body>
</html>
```

Penjelasan:

.kotak-animasi: Gaya awal untuk kotak animasi. position: relative; memungkinkan kita memanipulasi properti left.

animation-name: Menghubungkan elemen dengan aturan @keyframes bernama bergerakBerputar.

animation-duration: Durasi total satu siklus animasi (2 detik).

animation-iteration-count: Berapa kali animasi diulang (infinite untuk terus-menerus).

animation-timing-function: Kurva kecepatan untuk seluruh animasi (linear untuk kecepatan konstan).

@keyframes bergerakBerputar: Mendefinisikan urutan animasi.

0%: Pada awal animasi, kotak berada di kiri (left: 0) dan tidak berputar (rotate(0deg)), berwarna tomato.

50%: Di tengah animasi, kotak bergerak ke kanan (menggunakan calc() untuk menghitung posisi relatif terhadap lebar viewport) dan berputar 180 derajat, warnanya berubah menjadi orange.

100%: Di akhir animasi, kotak kembali ke kiri dan berputar 360 derajat, kembali berwarna tomato.

4. Gerak memudar membesar

```
C: > xampp > htdocs > animasi > memudarbesar.html > html > body > div.lingkaran
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Memudar membesar</title>
7    <style>
8      .lingkaran {
9        width: 80px;
10       height: 80px;
11       background-color: purple;
12       border-radius: 50%;
13       opacity: 1;
14       transform: scale(1);
15       animation-name: memudarMembesar;
16       animation-duration: 1.5s;
17       animation-iteration-count: infinite;
18       animation-direction: alternate; /* Animasi berjalan maju lalu mundur */
19     }
20
21     @keyframes memudarMembesar {
22       0% {
23         opacity: 1;
24         transform: scale(1);
25       }
26       100% {
27         opacity: 0.3;
28         transform: scale(1.5);
29       }
30     }
31   </style>
32 </head>
```

```

33     <body>
34     |     <div class="lingkaran"></div>
35     </body>
36     </html>

```

Penjelasan:

.lingkaran: Gaya awal untuk lingkaran.

animation-direction: alternate;; Membuat animasi berjalan maju dari 0% ke 100% lalu mundur dari 100% ke 0% pada iterasi berikutnya.

@keyframes memudarMembesar: Membuat lingkaran memudar dan membesar secara bergantian.

5. Script timeout

```

C: > xampp > htdocs > animasi > <> timeout.html > html > head > style
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Timeout</title>
7      <style>
8          #kotakJs {
9              width: 50px;
10             height: 50px;
11             background-color: blue;
12             position: absolute;
13             left: 10px;
14             top: 50px;
15         }
16     </style>
17 </head>

```

```

18 <body>
19   <div id="kotakJs"></div>
20   <script>
21     const kotakJs = document.getElementById('kotakJs');
22     let posisiX = 10;
23     const kecepatan = 5; // pixel per langkah
24
25     Windsurf: Refactor | Explain | Generate Function Comment | X
26     function animasikanKotak() {
27       posisiX += kecepatan;
28       kotakJs.style.left = posisiX + 'px';
29       if (posisiX < 300) {
30         setTimeout(animasikanKotak, 50); // Ulangi setelah 50ms
31       }
32     }
33
34     animasikanKotak();
35   </script>
36 </body>
</html>

```

Penjelasan:

Kita mendapatkan referensi ke elemen div dengan ID kotakJs.

Variabel posisiX menyimpan posisi horizontal kotak.

Fungsi animasikanKotak() mengubah properti left dari elemen, membuatnya bergerak ke kanan.

setTimeout() digunakan untuk memanggil fungsi animasikanKotak() lagi setelah jeda waktu tertentu, menciptakan efek animasi langkah demi langkah. Animasi berhenti ketika posisiX mencapai 300px.

6. Script setInterval

```
C: > xampp > htdocs > animasi > interval.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Interval</title>
7      <style>
8          #lingkaranJs {
9              width: 30px;
10             height: 30px;
11             background-color: green;
12             border-radius: 50%;
13             position: absolute;
14             left: 10px;
15             top: 100px;
16         }
17     </style>
18 </head>
19 <body>
20     <div id="lingkaranJs"></div>
21
22     <script>
23         const lingkaranJs = document.getElementById('lingkaranJs');
24         let opacity = 1;
25         const intervalId = setInterval(() => {
26             opacity -= 0.05;
27             lingkaranJs.style.opacity = opacity;
28             if (opacity <= 0) {
29                 clearInterval(intervalId); // Hentikan interval jika opasitas mencapai 0
30             }
31         }, 100); // Jalankan setiap 100ms
32     </script>
33
34 </body>
35 </html>
```

Penjelasan:

setInterval() memanggil fungsi anonim setiap 100 milidetik, secara bertahap mengurangi opasitas elemen.

clearInterval() digunakan untuk menghentikan interval setelah opasitas mencapai 0, mencegah fungsi terus berjalan tanpa perlu.

Kelemahan setTimeout() dan setInterval():

Tidak sinkron dengan refresh rate layar: Ini dapat menyebabkan animasi yang tidak mulus atau jank.

Tidak dioptimalkan oleh browser: Browser mungkin tidak memprioritaskan animasi ini, terutama jika tab tidak aktif.

7. Script Animation Frame

```
C: > xampp > htdocs > animasi > <> animationframe.html > html > body > script > [0] kecepatan

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Animation Frame</title>
7      <style>
8          #pesawat {
9              width: 80px;
10             height: 60px;
11             background-color: gold;
12             position: absolute;
13             left: -80px; /* Mulai di luar layar */
14             top: 150px;
15         }
16     </style>
17 </head>
18 <body>
19     <div id="pesawat">✈️</div>
20
21 <script>
22     const pesawat = document.getElementById('pesawat');
23     let posisiX = -80;
24     const kecepatan = 2; // pixel per frame
25
26     function animasikanPesawat() {
27         posisiX += kecepatan;
28         pesawat.style.left = posisiX + 'px';
29         if (posisiX < window.innerWidth) {
30             requestAnimationFrame(animasikanPesawat);
31         }
32     }
33
34     requestAnimationFrame(animasikanPesawat);
35 </script>
36 </body>
37
38 </html>
```


Penjelasan:

- Fungsi animasikanPesawat() dipanggil berulang kali oleh requestAnimationFrame().
- Setiap kali dipanggil, posisi pesawat diperbarui, menciptakan ilusi gerakan.
- Animasi berlanjut hingga pesawat mencapai tepi kanan layar.

8. Script animation AOS

Penjelasan Kode:

`<link rel="stylesheet" href="https://unpkg.com/aos@2.3.1/dist/aos.css">`: Baris ini menambahkan stylesheet AOS dari CDN unpkg. Ini berisi semua gaya CSS yang dibutuhkan untuk animasi AOS.

`<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>`: Baris ini menambahkan script JavaScript AOS dari CDN unpkg. Script ini berisi logika untuk mengaktifkan animasi saat elemen di-scroll ke dalam tampilan.

`<script> AOS.init(); </script>`: Kode JavaScript ini menginisialisasi AOS. Anda perlu memanggil AOS.init() agar library mulai bekerja. Anda juga bisa menambahkan opsi konfigurasi di dalam fungsi init(), seperti durasi default, easing, dll.

`<div class="element" data-aos="fade-up">...</div>`: Atribut data-aos adalah kunci untuk menggunakan AOS. Anda menambahkan atribut ini ke elemen HTML yang ingin Anda animasikan. Nilai dari atribut ini menentukan jenis animasi yang akan diterapkan. Beberapa contoh nilai yang umum adalah:

fade-up

fade-down

fade-left

fade-right

fade-up-right

fade-down-left

zoom-in

zoom-out

slide-up

slide-down

slide-left

slide-right

flip-left

flip-right

flip-up

flip-down

Atribut Data Tambahan: Anda juga bisa menambahkan atribut data-aos-* lainnya untuk mengontrol lebih lanjut animasi:

data-aos-duration="<milidetik>": Mengatur durasi animasi dalam milidetik.

data-aos-delay="<milidetik>": Menunda dimulainya animasi dalam milidetik.

data-aos-easing="<nama-easing>": Mengubah kurva kecepatan animasi (misalnya, ease-in-out-cubic, linear, dll.).

data-aos-offset="<pixel>": Menentukan seberapa jauh (dalam piksel) elemen harus di-scroll ke dalam tampilan sebelum animasi dimulai.

data-aos-once="true": Mencegah animasi diulang saat elemen di-scroll keluar dan masuk kembali ke tampilan.

data-aos-anchor="#id-elemen-lain": Menganimasikan elemen ini berdasarkan posisi elemen lain (dengan ID yang ditentukan).

data-aos-anchor-placement="top-bottom": Menentukan posisi relatif dari anchor elemen dan jendela untuk memicu animasi.

9. Scrip Animation GSAP

Tentu, berikut adalah cara menggunakan library animasi JavaScript GSAP (GreenSock Animation Platform) tanpa npm, beserta contoh kode HTML lengkap yang bisa Anda coba:

Langkah 1: Sertakan GSAP melalui CDN

Tambahkan script JavaScript GSAP dari CDN ke dalam bagian <body> file HTML Anda (biasanya sebelum tag </body> penutup). Jika Anda ingin menggunakan plugin GSAP seperti ScrollTrigger, sertakan juga script plugin tersebut.

HTML

```
<!DOCTYPE html>
```

```
<html lang="id">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Animasi dengan GSAP (Tanpa NPM)</title>
```

```
<style>
```

```
  body {
```

```
    font-family: sans-serif;
```

```
    margin: 0;
```

```
    padding: 20px;
```

```
    background-color: #f0f0f0;
```

```
  }
```

```
  .kotak {
```

```
    width: 100px;
```

```
    height: 100px;
```

```
    background-color: lightblue;
```

```
    margin-bottom: 20px;
```

```
    position: relative;
```

```
    display: flex;
```

```
    justify-content: center;
```

```
align-items: center;
color: white;
font-weight: bold;
}
```

```
.lingkaran {
width: 50px;
height: 50px;
background-color: lightcoral;
border-radius: 50%;
position: absolute;
top: 20px;
left: 20px;
display: flex;
justify-content: center;
align-items: center;
color: white;
font-size: 12px;
}
```

```
.section {
height: 500px;
background-color: #e0f7fa;
margin-bottom: 30px;
display: flex;
justify-content: center;
align-items: center;
font-size: 20px;
```

```
        color: #26a69a;
    }
</style>
</head>
<body>
```

```
<h2>Contoh Animasi GSAP Tanpa NPM</h2>
```

```
<div class="kotak" id="kotak1">Kotak 1</div>
```

```
<div class="kotak" id="kotak2">Kotak 2</div>
```

```
<h3>Animasi Timeline</h3>
```

```
<div class="lingkaran" id="lingkaran1">Gerak</div>
```

```
<h3>Animasi ScrollTrigger</h3>
```

```
<div class="section" id="section1">Scroll Pemicu</div>
```

```
<div class="kotak" id="kotak3" style="background-color: lightgreen;">Geser</div>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.5/gsap.min.js"></script>
```

```
<script>
```

```
    // Animasi sederhana menggunakan gsap.to()
```

```
    gsap.to("#kotak1", { x: 150, rotation: 90, duration: 1 });
```

```
    // Animasi dengan delay
```

```
    gsap.to("#kotak2", { x: 200, backgroundColor: "orange", duration: 0.7, delay: 0.5 });
```

```
    // Animasi menggunakan Timeline
```

```
    const tl = gsap.timeline({ repeat: -1, yoyo: true });
```

```
tl.to("#lingkaran1", { x: 100, duration: 0.5 })  
  .to("#lingkaran1", { y: 50, duration: 0.3 })  
  .to("#lingkaran1", { scale: 1.2, duration: 0.4 })  
  .to("#lingkaran1", { backgroundColor: "purple", duration: 0.2 });  
</script>
```

<h3>Contoh dengan ScrollTrigger (Optional)</h3>

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.5/ScrollTrigger.min.js"></script>
```

```
<script>  
  gsap.registerPlugin(ScrollTrigger);
```

```
  gsap.to("#kotak3", {  
    xPercent: 100,  
    scrollTrigger: {  
      trigger: "#section1",  
      start: "top center",  
      end: "bottom top",  
      scrub: true,  
      pin: true,  
      markers: true  
    }  
  });  
</script>
```

```
</body>
```

```
</html>
```

Penjelasan Kode:

`<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.5/gsap.min.js"></script>`: Baris ini menambahkan script utama GSAP dari CDN cdnjs. Pastikan Anda menggunakan versi terbaru atau versi yang Anda inginkan.

`<script>` (Blok Animasi Dasar): Di dalam blok script ini, kita menggunakan fungsi-fungsi dasar GSAP:

`gsap.to(target, vars)`: Menganimasikan properti dari nilai saat ini ke nilai yang ditentukan dalam vars.

`target`: Selector CSS (`"#kotak1"`) atau elemen DOM yang ingin dianimasikan.

`vars`: Objek yang berisi properti CSS atau properti GSAP lainnya yang ingin dianimasikan (misalnya, `x`, `y`, `rotation`, `backgroundColor`, `duration`, `delay`, `ease`).

`gsap.timeline(vars)`: Membuat sebuah timeline untuk mengatur urutan animasi. Properti `repeat: -1` membuat animasi berulang tanpa batas, dan `yoyo: true` membuat animasi bergerak maju lalu mundur pada setiap iterasi.

`.to()` pada Timeline: Menambahkan animasi ke dalam timeline. Animasi akan dijalankan secara berurutan sesuai dengan urutan `.to()` yang ditambahkan.

`<script`

`src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.5/ScrollTrigger.min.js"></script>`

(Opsional untuk ScrollTrigger): Jika Anda ingin menggunakan animasi yang dipicu oleh scroll, Anda perlu menyertakan plugin ScrollTrigger.

`<script>` (Blok Animasi ScrollTrigger):

`gsap.registerPlugin(ScrollTrigger)`: Mendaftarkan plugin ScrollTrigger agar dapat digunakan.

`gsap.to(target, vars)` dengan scrollTrigger: Menganimasikan elemen berdasarkan posisi scroll.

`scrollTrigger`: Objek yang mengonfigurasi pemicu scroll:

`trigger`: Selector CSS (`"#section1"`) atau elemen DOM yang menjadi pemicu animasi.

`start`: Posisi awal pemicu dan viewport untuk memulai animasi (misalnya, `"top center"` berarti saat bagian atas `#section1` mencapai tengah viewport).

end: Posisi akhir pemicu dan viewport untuk mengakhiri animasi (misalnya, "bottom top" berarti saat bagian bawah #section1 mencapai bagian atas viewport, atau "+=500" berarti setelah 500 piksel scroll).

scrub: true: Membuat animasi terkait langsung dengan pergerakan scroll. Anda bisa menggunakan nilai numerik (misalnya, scrub: 1) untuk memberikan efek delay.

pin: true: Membuat elemen pemicu tetap di tempatnya selama animasi berlangsung.

markers: true: Menampilkan garis bantu visual untuk start dan end (berguna untuk pengembangan).