

Eksplorasi CSS...

Materi Eksplorasi Animation

- Anggota Kelompok
 - ☐ Muh Farhan Maulana
 - ☐ Nur Rahmat Ramadan
- Kelaszz
 - XI Rekaysa Perangkat Lunak (1)

Apa Itu Animation

- **Animasi/Animation** CSS biasanya didefinisikan dengan menggunakan aturan `@keyframes`, yang memungkinkan Anda untuk menentukan perubahan-nilai properti pada titik-titik waktu tertentu selama animasi berlangsung. Selain itu, Anda juga dapat menggunakan properti-properti CSS seperti `animation-name`, `animation-duration`, `animation-timing-function`, `animation-delay`, `animation-iteration-count`, `animation-direction`, dan `animation-fill-mode` untuk mengontrol perilaku animasi

Property & Value

1. Animation Name & Keyframe

- Property animation name dalam CSS digunakan untuk menamai sebuah animasi yang didefinisikan menggunakan aturan `@keyframes`. Ini memungkinkan Anda untuk merujuk kembali ke animasi tersebut dalam kode CSS atau JavaScript untuk memulai atau menghentikan animasi, atau untuk mengaitkan animasi tersebut dengan elemen tertentu dalam dokumen HTML

```
/* Mendefinisikan animasi */
@keyframes nama-animasi {
  /* Aturan-aturan animasi */
}
/* Menggunakan animasi dengan nama tersebut */
.element {
  animation-name: nama-animasi;
}
```

- **Keyframes** dalam CSS digunakan untuk mendefinisikan perubahan-nilai properti CSS pada titik-titik waktu tertentu selama animasi. Ini dilakukan dengan menggunakan aturan `@keyframes`, diikuti oleh nama animasi dan blok aturan yang berisi perubahan-nilai pada titik-titik waktu tertentu, seperti `0%`, `50%`, dan `100%`.

```
@keyframes nama-animasi {
  0% {
    /* Nilai-nilai properti pada 0% waktu animasi */
  }
  50% {
```

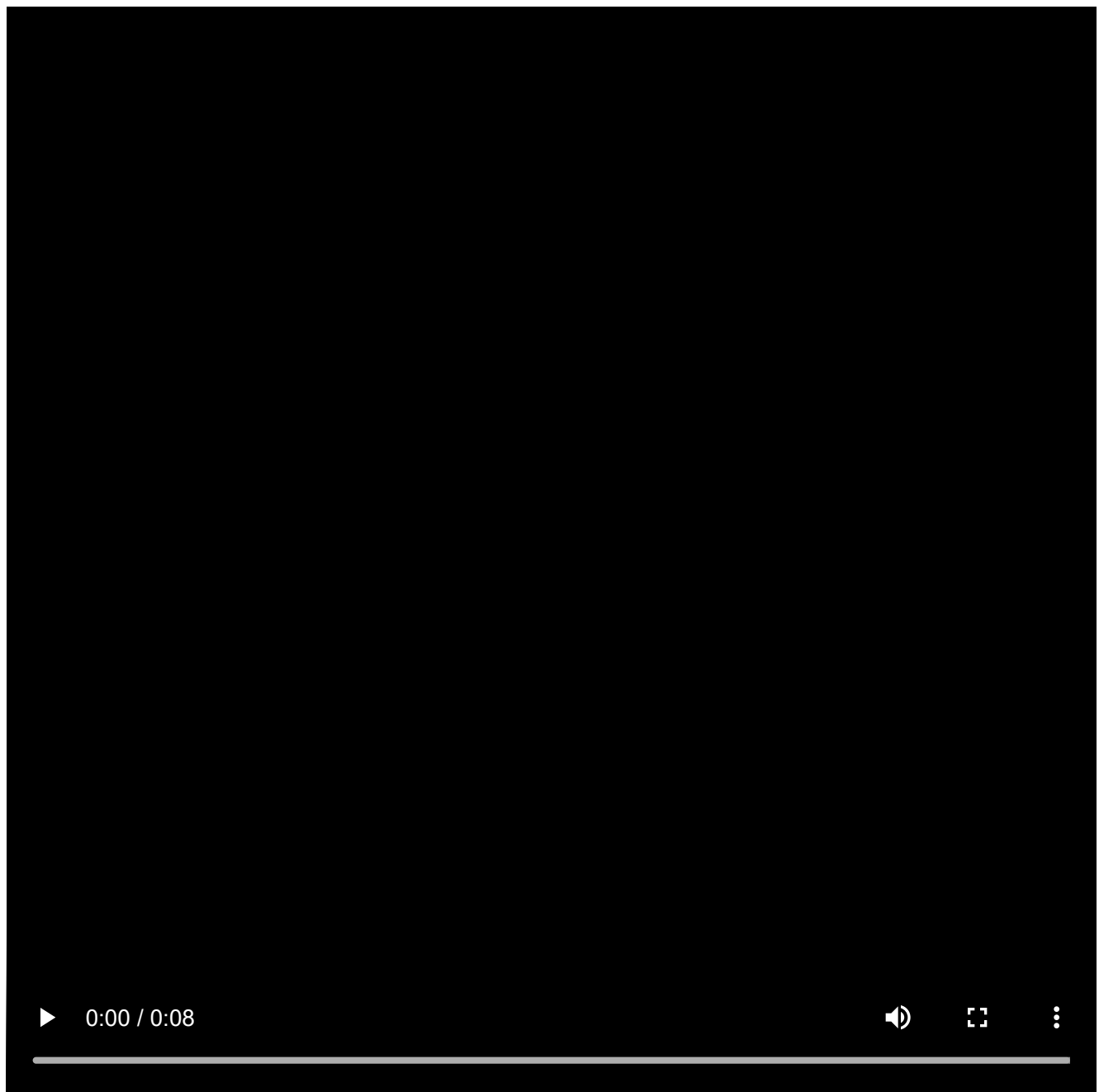
```
    /* Nilai-nilai properti pada 50% waktu animasi */  
  }  
  100% {  
    /* Nilai-nilai properti pada 100% waktu animasi */  
  }  
}
```

2. Animation Duration

- **animation-duration** adalah properti dalam CSS yang digunakan untuk menentukan durasi dari suatu animasi. Properti ini menentukan waktu yang dibutuhkan untuk animasi dari awal hingga akhir. Nilainya bisa dalam satuan detik (s) atau milidetik (ms)

```
.element {  
  animation-duration: 5s; /* Animasi berlangsung selama 2 detik */  
}
```

- **Contoh Hasil**



Keterangan Hasil

Atribut `animation-duration` akan membuat box di atas bergerak dengan valuenya yaitu 5s(detik) dengan adanya `@keyframe` box akan bergeser dari kiri ke kanan dengan waktu yang telah di berikan.

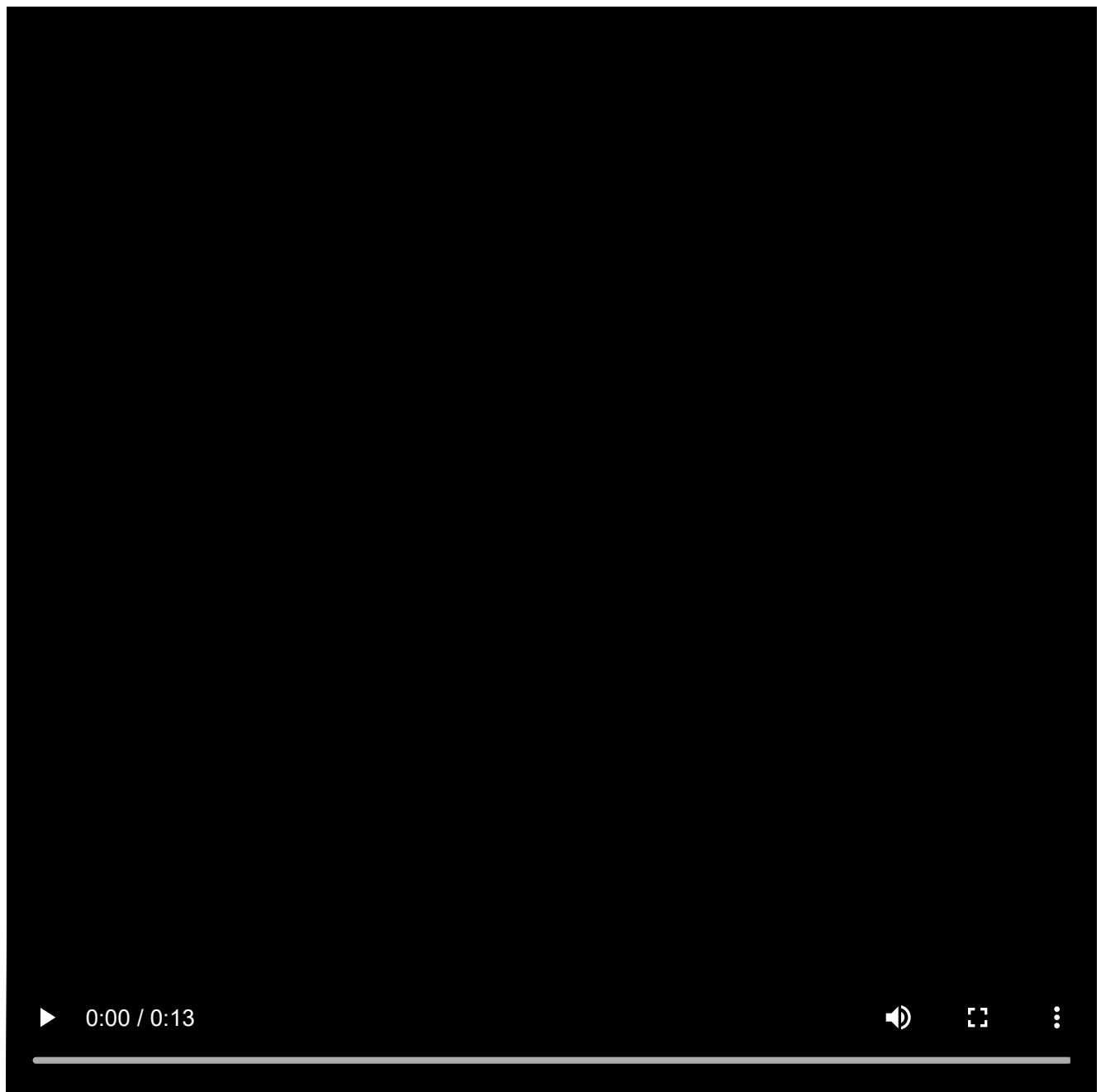
3. Animation Timing Function

- Fungsi-timing animasi (animation timing function) dalam CSS digunakan untuk mengontrol bagaimana perubahan nilai animasi terjadi selama durasi animasi. Fungsi ini memungkinkan Anda untuk menentukan percepatan (acceleration) atau perlambatan (deceleration) dari animasi, atau bahkan membuat animasi terjadi dengan kecepatan konstan.
- Ada beberapa jenis fungsi-timing yang umum digunakan:
 - `ease`: Ini adalah fungsi-timing standar yang memberikan percepatan dan perlambatan perlahan ke dan dari animasi. Ini adalah default jika tidak ada fungsi-timing yang ditentukan.

- **linear**: Memberikan kecepatan konstan selama animasi. Perubahan nilai berlangsung dengan laju konstan.
- **ease-in**: Memberikan perlambatan pada awal animasi, tetapi percepatan konstan setelahnya.
- **ease-out**: Memberikan percepatan pada awal animasi, tetapi perlambatan konstan saat berakhir.
- **ease-in-out**: Memberikan perlambatan pada awal dan percepatan pada pertengahan animasi, serta perlambatan kembali saat berakhir.
- **cubic-bezier()**: Fungsi-timing kustom yang memungkinkan Anda untuk mengontrol percepatan dan perlambatan dengan lebih detail menggunakan koordinat kubik bezier. Ini memungkinkan Anda membuat kurva-timing animasi yang sangat kustom.

```
.element {  
  animation-timing-function: ease-in-out; /* Menggunakan fungsi-timing ease-in-out */  
}
```

- **Contoh Hasil**



⚠ Keterangan Hasil

Box di atas bergerak ke arah kiri dengan perintah `@keyframe` di `styles.css` dengan value ``ease-in-out`
akan bergerak di awal dengan secara perlahan (ease-in) dan mencapai kecepatan penuh di tengah, dan
kemudian berkurang kecepatannya secara perlahan (ease-out), ini memberi kesan pergerakan yang halus

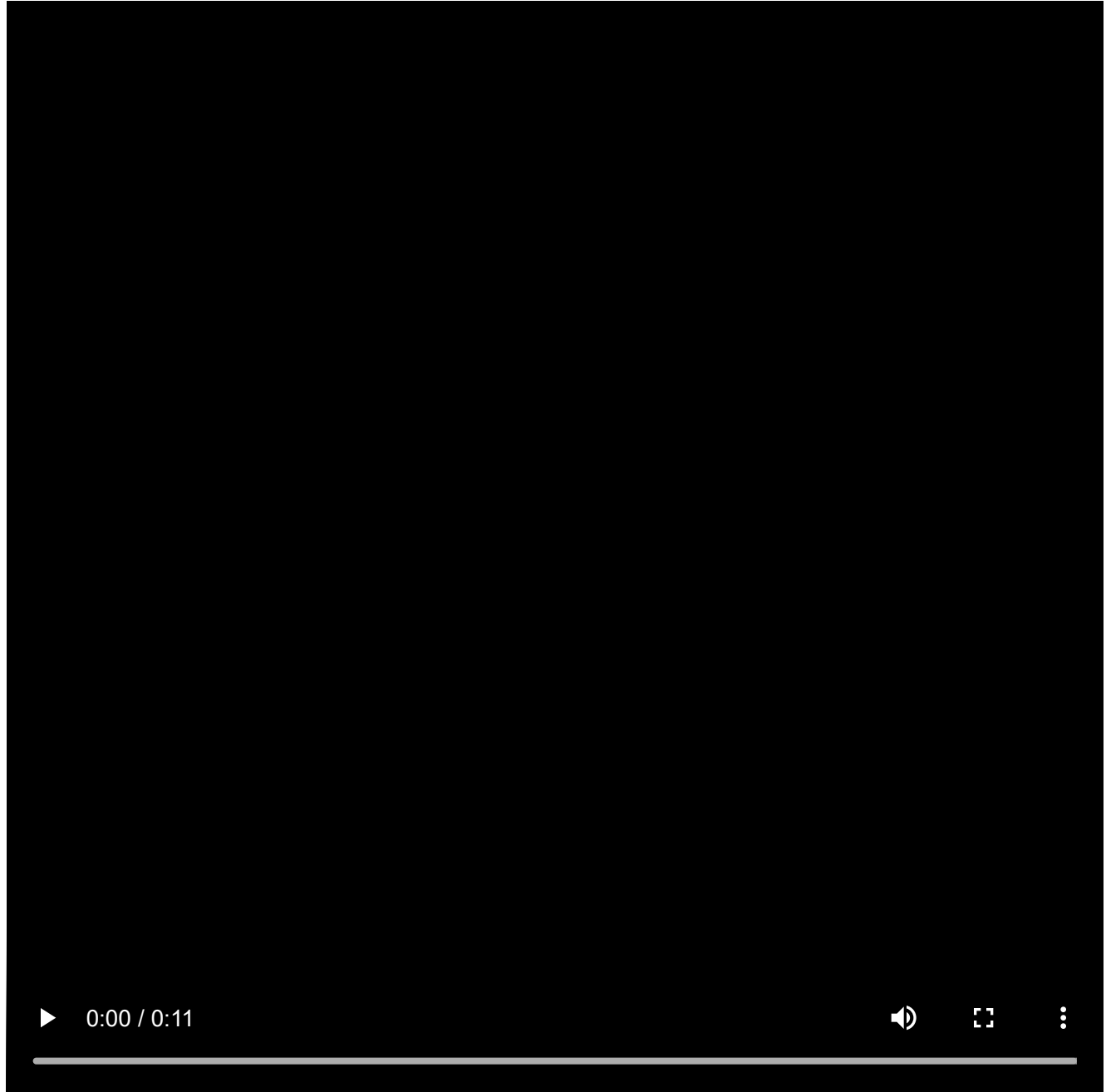
4. Animation Delay

- `animation-delay` dalam CSS adalah properti yang memungkinkan Anda menunda awal dari suatu animasi. Dengan menentukan nilai delay, Anda dapat mengatur waktu tunggu sebelum animasi dimulai, baik dalam detik (s) atau milidetik (ms). Ini berguna untuk mengatur kapan animasi

harus dimulai setelah suatu peristiwa tertentu, seperti saat elemen muncul di layar atau setelah interaksi pengguna tertentu terjadi.

```
.element {  
  animation-delay: 3s; /* Animasi akan dimulai setelah 0.5 detik */  
}
```

- **Contoh Hasil**



Keterangan Hasil

Box di atas akan muncul dari yang awalnya tidak terlihat menjadi terlihat dengan opacity di dalam

`@keyframe` box tersebut akan tetap diam selama 3s(detik) sesuai valuenya sebelum animasi fadein-nya

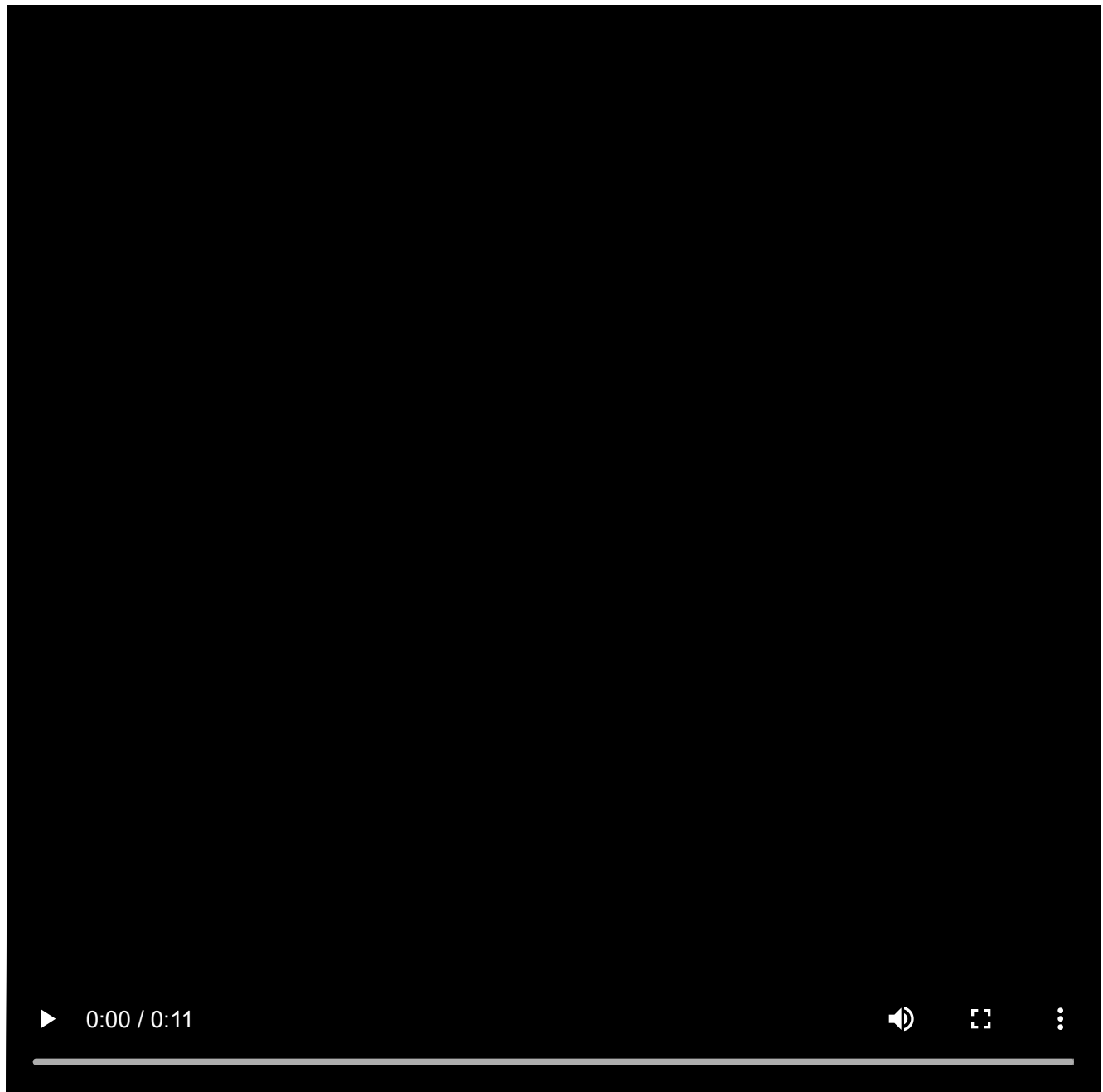
dimulai yang dari awalnya warna box ini buram dan kemudian menjadi pekat atau cerah.

5. Animation Iteration Count

- `animation-iteration-count` dalam CSS digunakan untuk menentukan berapa kali sebuah animasi harus diulang. Nilai dari properti ini dapat berupa bilangan bulat (integer) atau kata kunci spesifik. Berikut adalah beberapa nilai yang dapat digunakan:
 1. `infinite`: Animasi akan terus diulang tanpa henti sampai dihentikan secara manual.
 2. Bilangan bulat positif: Menunjukkan jumlah kali animasi harus diulang. Misalnya, nilai 2 akan membuat animasi diulang dua kali.

```
.element {  
  animation-iteration-count: 3; /* Animasi akan diulang sebanyak 3 kali */  
  /* atau bisa juga */  
}
```

- Contoh Hasil



Keterangan Hasil

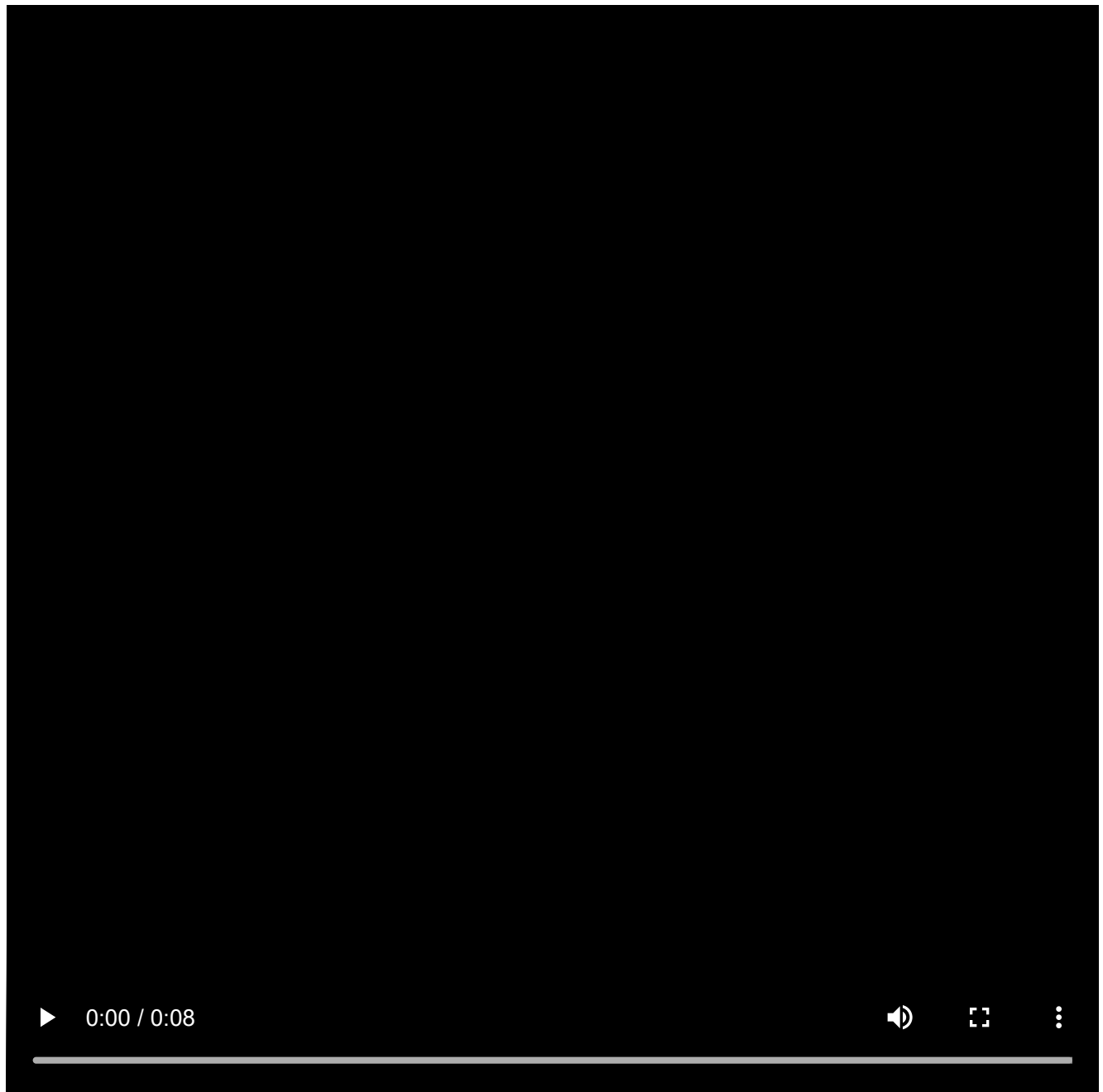
Box di atas akan mengarah kekiri secara berulang sebanyak 3 kali dikarenakan valuenya adalah 3, dan setiap animasi bergerak selama 2 detik setelah box terulang 3 kali maka animasi akan selesai dan box akan berhenti bergerak.

6. Animation Direction

- **animation-direction** dalam CSS digunakan untuk menentukan arah perubahan nilai properti selama animasi. Properti ini memiliki beberapa nilai yang dapat digunakan:
 1. **normal**: Ini adalah nilai default. Animasi akan berjalan dari awal ke akhir sesuai dengan keyframes yang didefinisikan.
 2. **reverse**: Animasi akan berjalan dari akhir ke awal, membalik urutan keyframes yang didefinisikan.
 3. **alternate**: Animasi akan bergantian antara berjalan dari awal ke akhir dan dari akhir ke awal pada setiap iterasi.
 4. **alternate-reverse**: Mirip dengan **alternate**, namun animasi pertama kali berjalan dari akhir ke awal.

```
.element {  
  animation-direction: reverse; /* Animasi akan berjalan dari akhir ke awal */  
}
```

- Contoh Hasil



Keterangan Hasil

Property ini akan mengatur arah animasi akan kemana disini terdapat value `reverse` jadi box ini akan bergerak dari akhir ke awal atau dari kanan ke kiri layar.

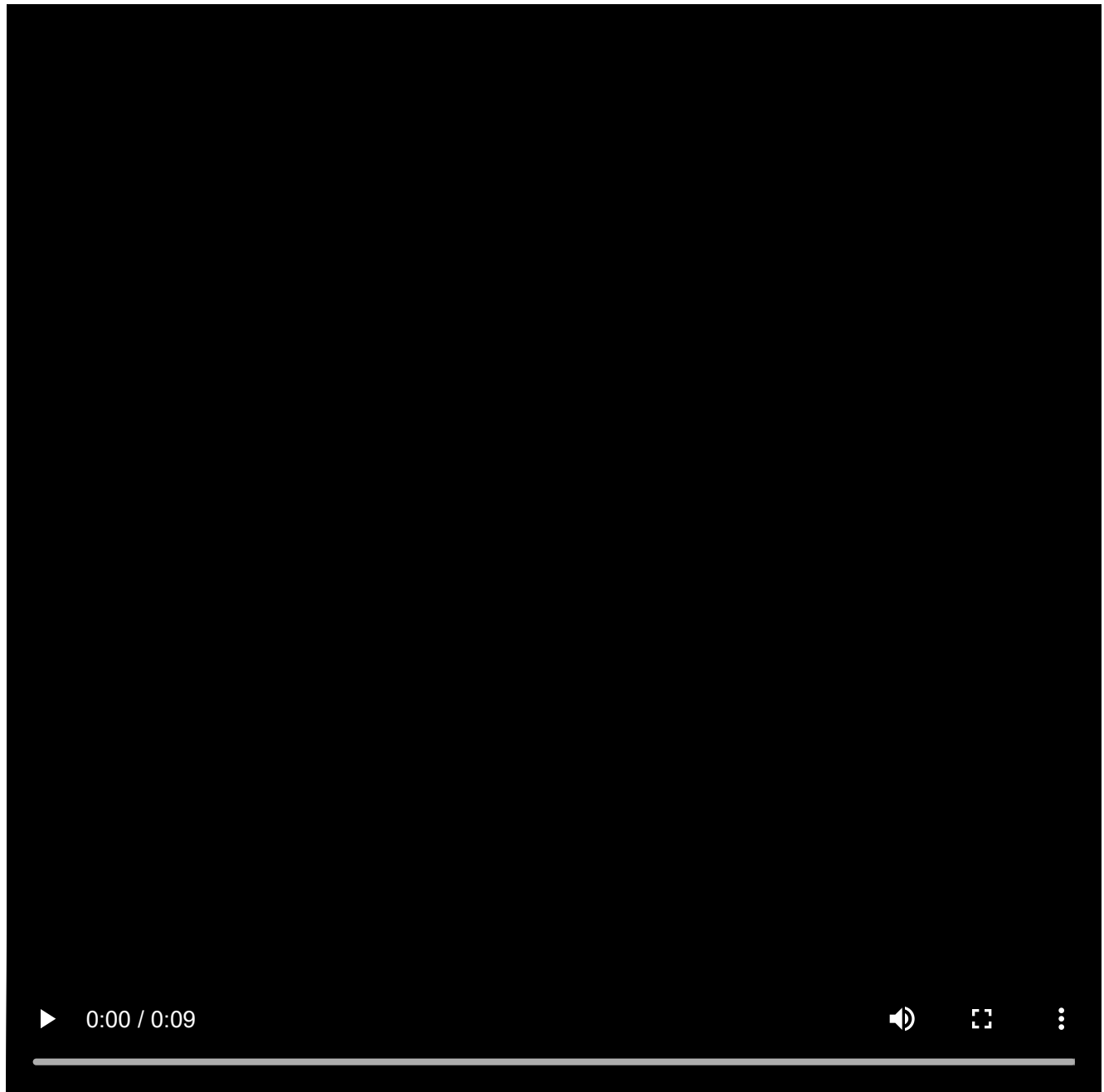
7. Animation Fill Mode

- `animation-fill-mode` dalam CSS digunakan untuk menentukan bagaimana nilai properti animasi diterapkan sebelum dan setelah animasi berjalan. Properti ini memengaruhi perilaku animasi sebelum dan setelah animasi sesuai dengan siklus animasi dan apakah animasi dimulai dan berhenti.
- Properti `animation-fill-mode` memiliki beberapa nilai yang dapat digunakan:
 1. `none`: Ini adalah nilai default. Tidak ada nilai properti animasi yang diterapkan sebelum atau setelah animasi berjalan.

2. **forwards**: Properti animasi diterapkan pada elemen target setelah animasi selesai. Artinya, nilai terakhir dari animasi tetap berlaku setelah animasi selesai.
3. **backwards**: Properti animasi diterapkan pada elemen target sebelum animasi dimulai. Ini berarti nilai pertama dari animasi diterapkan bahkan sebelum animasi dimulai.
4. **both**: Kombinasi dari **forwards** dan **backwards**, yang berarti properti animasi diterapkan sebelum animasi dimulai dan tetap berlaku setelah animasi selesai.

```
.element {  
  animation-fill-mode: forwards; /* Nilai terakhir animasi akan diterapkan setelah animasi  
selesai */  
}
```

- **Contoh Hasil**



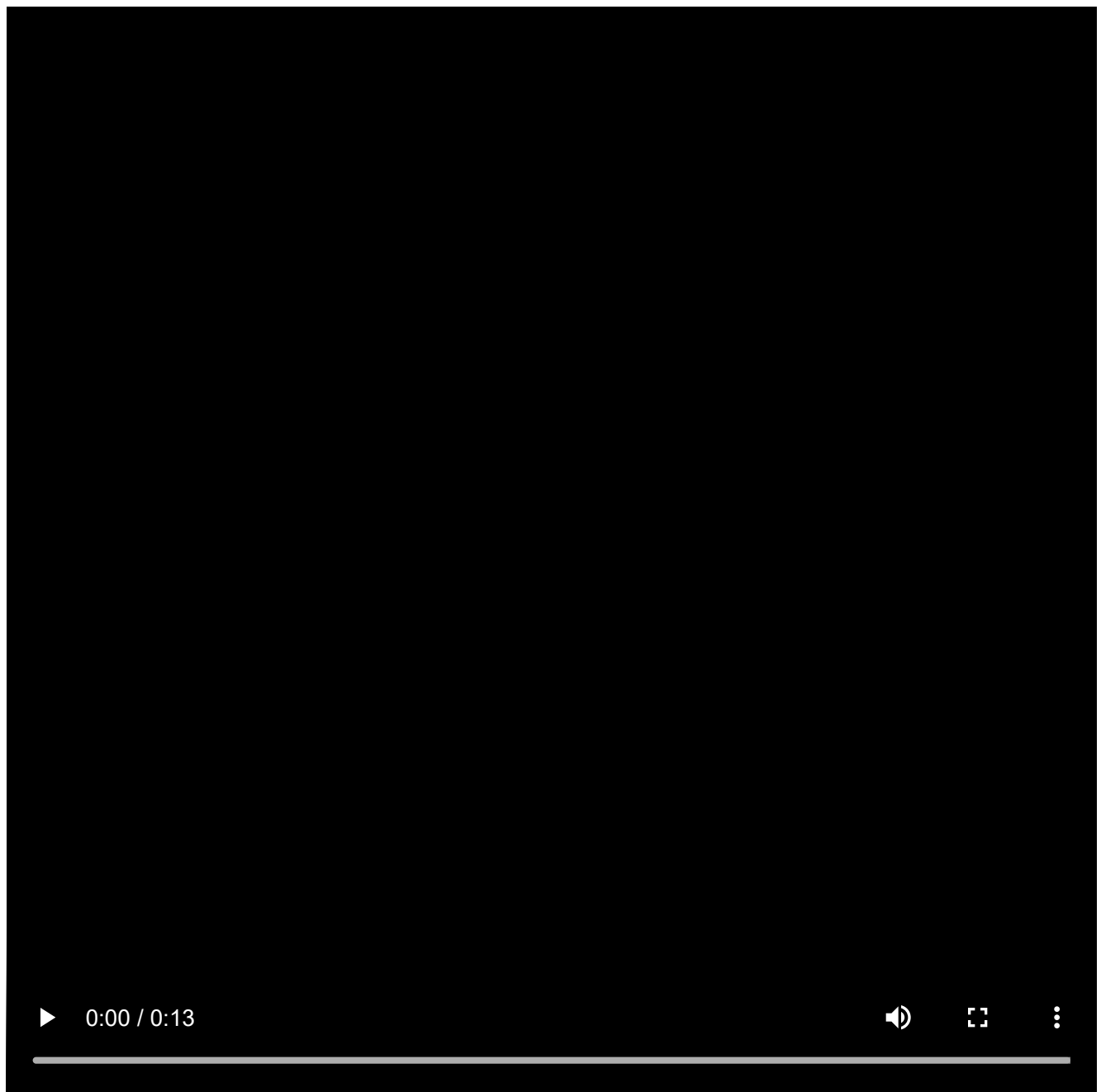
Dis sini box akan bergerak dari arah kiri ke kanan dengan adanya value **forwards** dengan kata lain box dari posisi A akan bergerak ke posisi B ketika animasi selesai box akan tetap berada di posisi B.

8. Animation Play State

- **animation-play-state** dalam CSS digunakan untuk mengontrol apakah suatu animasi sedang berjalan atau dijeda. Properti ini memungkinkan Anda untuk mengontrol secara dinamis status animasi menggunakan CSS
- Properti **animation-play-state** memiliki dua nilai yang dapat digunakan:
 1. **running**: Ini adalah nilai default. Animasi sedang berjalan.
 2. **paused**: Animasi dijeda pada titik waktu saat ini. Animasi akan berhenti pada posisi di mana ia dijeda.

```
.element {  
  animation-play-state: running; /* Animasi bergerak */  
}
```

- Contoh Hasil



⚠ Keterangan Hasil

Animasi menggunakan aturan `@keyframe` untuk mendefinisikan posisi awal dan akhir box akan bergerak cepat karena valuenya adalah `running` dimana box akan bergerak tanpa hambatan dan akan terus bergerak dengan tambahan property `animation-iteration-count: infinite;`

Implementasi

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Implementasi</title>
  <link rel="stylesheet" href="styles.css">
```

```

</head>
<body>
  <header>
    <div class="logo">ShoopiCoo</div>
    <nav>
      <ul>
        <li><a href="#">Beranda</a></li>
        <li><a href="#">Tentang</a></li>
        <li><a href="#">Layanan</a></li>
        <li><a href="#">Kontak</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <div class="container">
      <div class="card">
        
        <div class="card-content">
          <h3>Foto Berkelas</h3>
          <p>Cuman ada satu didunia.</p>
          <button>Lihat Detail</button>
        </div>
      </div>
      <div class="card">
        
        <div class="card-content">
          <h3>Foto Berkelas</h3>
          <p>Cuman ada satu didunia.</p>
          <button>Lihat Detail</button>
        </div>
      </div>
      <div class="card">
        
        <div class="card-content">
          <h3>Foto Berkelas</h3>
          <p>Cuman Satu didunia.</p>
          <button>Lihat Detail</button>
        </div>
      </div>
    </div>
  </main>
  <footer class="footer-animation">
    <p>Hak Cipta &copy; 2024 Website Animasi</p>
  </footer>
</body>
</html>

```

Index.html

- Ini adalah kerangka dasar halaman web dalam format HTML.
- Header (`<header>`) terdiri dari sebuah div dengan kelas `logo` dan sebuah `nav` yang berisi daftar menu navigasi. - Konten utama (`<main>`) dari halaman web terletak di dalam elemen `main`. - Konten utama tersebut terdiri dari sebuah `div` dengan kelas `container` yang berisi beberapa

kartu (`div` dengan kelas `card`) yang masing-masing memiliki gambar, judul, deskripsi, dan tombol "Lihat Detail". - Di bagian bawah halaman (`<footer>`), terdapat pesan hak cipta dengan kelas `footer-animation`.

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f0f0f0;
}

header {
  background-color: #333;
  color: #fff;
  padding: 20px 0;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo {
  font-size: 24px;
  font-weight: bold;
}

nav ul {
  list-style-type: none;
  padding: 0;
  margin: 0;
}

nav ul li {
  display: inline;
  margin-right: 20px;
}

nav ul li a {
  color: #fff;
  text-decoration: none;
}

main {
  padding: 20px;
}

.container {
  display: flex;
  justify-content: space-between;
  align-items: flex-start;
}

.card {
  width: 30%;
```

```
background-color: #fff;
border-radius: 8px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
overflow: hidden;
margin-bottom: 20px;
transition: transform 0.3s ease;
}

.card:hover {
  transform: scale(1.05);
}

.card img {
  width: 100%;
  height: auto;
}

.card-content {
  padding: 20px;
}

.card-content h3 {
  margin-top: 0;
  font-size: 20px;
}

.card-content p {
  margin-bottom: 10px;
}

button {
  padding: 10px 20px;
  background-color: #333;
  color: #fff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: lightblue;
}

button:active {
  background-color: #666;
}

footer {
  background-color: #333;
  color: #fff;
  text-align: center;
  padding: 10px 0;
}
```

```

@keyframes slideInFromBottom {
  0% {
    opacity: 0;
    transform: translateY(100%);
  }
  100% {
    opacity: 1;
    transform: translateY(0);
  }
}

.footer-animation {
  animation: slideInFromBottom 2s ease forwards;
}

footer {
  background-color: #333;
  color: #fff;
  text-align: center;
  padding: 10px 0;
}

@keyframes moveRightLeft {
  0% {
    transform: translateX(0);
  }
  50% {
    transform: translateX(50%);
  }
  100% {
    transform: translateX(0);
  }
}

.logo {
  font-size: 24px;
  font-weight: bold;
  animation: moveRightLeft 4s linear infinite;
}

```

Styles.css

- Mendefinisikan gaya untuk elemen-elemen HTML di halaman web.
- Misalnya, **body** memiliki gaya font, warna latar belakang, dan mengatur margin dan padding menjadi 0.
- Bagian **header** memiliki latar belakang hitam (**#333**), teks putih (**#fff**), dan diatur untuk tampil sebagai flexbox dengan penyebaran ruang antara elemen-elemen di dalamnya.
Ini hanya sebagian kecil dari kode CSS, tetapi Anda bisa melihatnya untuk mengatur tata letak, warna, animasi, dll.
- **Contoh Hasil**

