

LAPORAN PRAKTIKUM PEMROGRAMAN WEB
MODUL CSS
GRID



Disusun Oleh:

Farhan Dwi Septian / 105221036

27 Oktober 2023

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
PEMROGRAMAN BERORIENTASI OBJEK
UNIVERSITAS PERTAMINA

2023/2024

Daftar Isi

A. PENDAHULUAN.....	3
B. METODE	4
C. PEMBAHASAN.....	9
D.DOKUMENTASI.....	13
E. PUNUTUP	14

PENDAHULUAN

A. Latar Belakang

Pada era digital yang berkembang pesat ini, industri film telah mengalami perubahan signifikan dalam hal distribusi, promosi, dan interaksi dengan penonton. Internet telah menjadi saluran utama bagi film-film untuk mencapai audiens global dengan cepat dan efisien. Dalam konteks ini, proyek pembuatan website film telah menjadi bagian integral dari strategi pemasaran dan promosi film.

Website film adalah alat yang berharga untuk menghadirkan informasi tentang film-film terbaru, membangun penggemar, dan meningkatkan interaksi dengan penonton. Situs web semacam ini berperan dalam memfasilitasi akses penonton ke trailer film, ulasan, berita, dan kadang-kadang juga memungkinkan penonton untuk menonton film secara online melalui layanan streaming.

B. Tujuan Proyek

Proyek pembuatan website film "FilmWebX" bertujuan untuk menyediakan sumber daya informatif dan menghibur bagi penggemar film, serta mempromosikan film-film terbaru secara efektif.

C. Batasan Masalah

Pada pembuatan website film pada praktikum 3 kali ini hanya membahas tentang bagaimana pembuatan hero section pada tampilan landing page website film menggunakan html dan css.

METODE

A. CSS

CSS adalah singkatan dari Cascading Style Sheet, digunakan untuk melengkapi file HTML, dan tugas utamanya adalah menetapkan aturan tampilan/style yang akan digunakan pada sebuah website. CSS diperkenalkan untuk pengembangan website pada tahun 1996.

Nama CSS di dapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap style. Setelah CSS distandarisasikan, Internet Explorer dan Netscape melepas browser terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

CSS adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan dalam kode HTML atau sekedar mejadi rujukan oleh HTML dalam pendefinisian style. CSS menggunakan kode2 yang tersusun untuk menetapkan style pda elemen HTML atau dapat juga digunakan membuat style baru yang biasa disebut class.

B. Kegunaan CSS

CSS dapat digunakan untuk mengubah besar kecilnya text, mengganti warna background pada sebuah halaman, atau dapat pula mengubah warna border pada tabel, dan masih banyak lagi hal yang dapat dilakukan oleh CSS.

Singkatnya, CSS digunakan untuk mengatur susunan tampilan pada halaman HTML. CSS dapat digunakan untuk menggantikan , , <u> dan <u>, dikarenakan hal berikut:

1. Sebuah file css dapat menjadi rujukan banyak halaman HTML. Hanya dibutuhkan 1 baris kode untuk melakukan hal tersebut. Ini berarti akan meminimalkan file2 HTML yang akan dibuat.
2. Jika ingin mengubah tampilan website yang telah dibuat, maka yang perlu dilakukan hanya mengganti baris-baris kode pada css nya saja, tanpa perlu mengutak-atik file-file HTML nya.
3. CSS dapat mengatur banyak atribut pada sebuah halaman secara mudah. Misalnya: warna background, border, shadow, yang berbeda pada masing-masing tag yang digunakan

C. Cara Penulisan CSS

Cara penulisan CSS (Cascading Style Sheets) dapat beragam, tergantung pada preferensi dan kebutuhan pengembang web. Beberapa pendekatan umum yang digunakan dalam penulisan CSS meliputi:

1. Inline CSS: CSS ditulis langsung dalam atribut "style" elemen HTML. Contohnya:

```
<p style="color: blue; font-size: 16px;">Ini adalah teks dengan inline CSS.</p>
```

2. Internal CSS: CSS ditulis dalam elemen `<style>` yang terletak di dalam elemen `<head>` dokumen HTML. Contohnya:

```
<head>
  <style>
    p {
      color: red;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <p>Ini adalah teks dengan internal CSS.</p>
</body>
```

3. External CSS: CSS ditulis dalam file terpisah dengan ekstensi `.css` dan kemudian dihubungkan dengan dokumen HTML menggunakan tag `<link>`. Contohnya:

File "styles.css":

```
/* styles.css */
p {
  color: green;
  font-size: 20px;
}
```

Dalam HTML:

```
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <p>Ini adalah teks dengan external CSS.</p>
</body>
```

D. Komponen CSS

CSS (Cascading Style Sheets) terdiri dari berbagai komponen yang digunakan untuk mengendalikan tampilan dan tata letak elemen-elemen HTML pada halaman web. Berikut adalah komponen-komponen utama dalam CSS:

1. Selektor (Selector):

Selektor adalah bagian CSS yang digunakan untuk memilih elemen HTML yang akan diubah tampilannya.

Contoh: `p` (mengganti semua elemen `<p>`), `.class` (mengganti semua elemen dengan kelas CSS), `#id` (mengganti elemen dengan ID tertentu).

2. Properti (Property):

Properti adalah atribut-atribut CSS yang mengendalikan aspek tampilan elemen yang dipilih.

Contoh: color (mengubah warna teks), font-size (mengatur ukuran font), background-color (mengubah warna latar belakang).

Nilai (Value):

3. Nilai adalah nilai yang diberikan kepada properti untuk mengendalikan tampilan elemen.

Contoh: "red" (warna merah), 16px (ukuran font 16 piksel), #f0f0f0 (kode warna heksadesimal).

Aturan CSS (CSS Rule):

E. Display CSS

Properti CSS "display" digunakan untuk mengontrol bagaimana elemen HTML ditampilkan dalam hal tata letak di halaman web. Properti "display" memungkinkan Anda untuk mengubah elemen dari perilaku tampilan defaultnya. Berikut beberapa nilai yang umum digunakan untuk properti "display":

1. block:

Elemen tampil sebagai blok yang menempati seluruh lebar kontainer yang tersedia. Elemen blok selalu dimulai pada baris baru dan menempati lebar penuh dari kontainer.

Contoh elemen block termasuk <div>, <p>, dan elemen heading <h1>, <h2>, dst.

```
element {  
  display: block;  
}
```

2. inline:

Elemen tampil sebagai elemen inline, yang berarti mereka hanya memakan sejumlah lebar yang dibutuhkan oleh kontennya.

Elemen inline tidak memulai baris baru dan bisa sejajar dengan elemen lain dalam satu baris.

Contoh elemen inline termasuk , <a>, dan elemen teks seperti , , dst.

```
element {  
  display: inline;  
}
```

3. inline-block:

Elemen tampil sebagai elemen inline yang memiliki sifat blok. Mereka memakan sejumlah lebar yang dibutuhkan oleh kontennya, namun dapat menerima properti CSS yang biasanya hanya berlaku untuk elemen blok.

Ini adalah pilihan yang umum digunakan untuk menggabungkan sifat-sifat elemen blok dan elemen inline dalam satu elemen.

```
element {  
  display: inline-block;  
}
```

4. none:

Elemen sepenuhnya disembunyikan dan tidak ada ruang yang ditinggalkan di halaman untuk elemen ini.

Ini digunakan untuk menyembunyikan elemen secara dinamis dengan JavaScript atau CSS.

```
element {  
    display: none;  
}
```

5. flex:

Elemen tampil sebagai flex container, yang memungkinkan Anda mengatur elemen anaknya dalam susunan fleksibel.

Elemen-elemen dalam flex container berperan dalam tata letak flexbox, memungkinkan penyesuaian yang lebih baik dalam penataan elemen dalam satu baris atau kolom.

```
element {  
    display: flex;  
}
```

6. grid:

Elemen tampil sebagai grid container, yang memungkinkan Anda mengatur elemen anaknya dalam susunan grid dua dimensi.

Ini memungkinkan Anda untuk membuat tata letak yang lebih kompleks daripada flexbox dengan baris dan kolom yang terdefinisi dengan jelas.

```
element {  
    display: grid;  
}
```

Nilai "display" dapat sangat memengaruhi cara elemen-elemen HTML diatur dalam tata letak halaman web. Memahami berbagai nilai dan cara menggunakannya adalah kunci untuk mengendalikan tampilan elemen-elemen Anda dengan CSS.

F. Detail Grid Properti

- celah kolom: Digunakan untuk menentukan jumlah celah antar kolom di mana teks tertentu dibagi menggunakan properti jumlah kolom.
- gap : Digunakan untuk mengatur jarak juga pemanggil talang antara baris dan kolom.
- grid: Menawarkan sistem tata letak berbasis grid, dengan baris dan kolom, membuatnya lebih mudah untuk mendesain halaman web tanpa float dan positioning.
- area grid: Digunakan untuk mengatur ukuran dan lokasi item grid dalam tata letak grid.
- grid-auto-columns: Digunakan untuk menentukan ukuran kolom kontainer grid yang dihasilkan secara implisit.
- grid-auto-flow: Ini menentukan dengan tepat bagaimana item yang ditempatkan secara otomatis mengalir ke dalam grid.
- grid-auto-rows: Digunakan untuk menentukan ukuran baris container grid yang dihasilkan secara implisit.
- grid-column: Ini menjelaskan jumlah properti yang memungkinkan untuk merancang struktur grid dan mengontrol penempatan item grid menggunakan CSS.
- grid-column-end: Ini menjelaskan jumlah kolom yang akan direntang suatu item, atau di baris kolom mana item tersebut akan berakhir.

- `grid-column-gap`: Digunakan untuk mengatur ukuran jarak antar kolom dalam tata letak grid.
- `grid-column-start`: Ini menentukan item baris kolom mana yang akan dimulai.
- `grid-gap`: Digunakan untuk mengatur ukuran jarak antara baris dan kolom dalam tata letak grid.
- `grid-row`: Digunakan untuk menentukan ukuran dan lokasi dalam tata letak grid.
- `grid-row-end`: Digunakan untuk menentukan posisi akhir item grid dalam baris grid dengan menentukan tepi inline dari area gridnya.
- `grid-row-gap`: Digunakan untuk menentukan ukuran jarak antar elemen grid.
- `grid-row-start`: Digunakan untuk menentukan posisi awal item grid dalam baris grid dengan menentukan tepi awal inline dari area gridnya.
- `grid-template`: Ini adalah properti singkatan untuk mendefinisikan kolom, baris, dan area grid.
- `grid-template-areas`: Digunakan untuk menentukan area dalam tata letak grid.
- `grid-template-columns`: Digunakan untuk mengatur jumlah kolom dan ukuran kolom grid.
- `grid-template-rows`: Digunakan untuk mengatur jumlah baris dan tinggi baris dalam grid.

Display Grid adalah salah satu teknik tata letak dalam CSS yang memungkinkan kita untuk membuat tata letak berbasis grid (kisi) dengan lebih fleksibel dan terstruktur. Dengan menggunakan Grid, kita dapat mengatur elemen-elemen HTML dalam baris dan kolom, menciptakan struktur yang rapi dan mudah diatur.

Grid terdiri dari dua komponen utama: kontainer grid dan item-item grid. Kontainer grid adalah elemen yang mengelilingi item-item yang akan diatur dalam tata letak grid. Properti `display: grid` diterapkan pada kontainer ini. Item-item grid adalah elemen-elemen yang akan ditempatkan dalam grid.

Keunggulan utama dari Grid adalah kemampuannya untuk mengatur elemen-elemen secara dua dimensi, yang berarti kita dapat mengendalikan baik baris maupun kolom. Ini memberi kita kontrol yang luar biasa dalam mendefinisikan tata letak halaman web.

Selain itu, Grid juga mendukung alur konten yang responsif, yang berarti kita dapat menentukan seberapa banyak ruang yang harus digunakan oleh setiap item grid berdasarkan ukuran layar atau tampilan. Hal ini sangat bermanfaat dalam mendesain tata letak yang responsif untuk berbagai perangkat, mulai dari desktop hingga perangkat seluler.

Dalam tata letak web yang semakin kompleks, penggunaan Grid menjadi penting untuk mencapai desain yang konsisten dan terstruktur. Ini memungkinkan pengembang web untuk menciptakan tampilan yang estetis dan fungsional dengan lebih sedikit kode CSS, menghemat waktu dan usaha dalam pengembangan web.

PEMBAHASAN

Pada bab ini akan dijelaskan secara rinci css grid dari website yang saya bikin. Berikut source codenya dan pembasahannya:

A screenshot of a code editor with a dark background and light-colored text. The code is CSS for a class named .container. It sets width to 100%, height to 20%, and uses grid-template-areas with four named areas: header, kiri, kanan, and footer. It also sets display to grid, grid-template-columns to 5fr 5fr, and grid-template-rows to 2fr 4fr 5fr 2fr. The code is numbered from 1 to 12.

```
1 .container {  
2   width: 100%;  
3   height: 20%;  
4   grid-template-areas:  
5     "header header"  
6     "kiri kanan"  
7     "kiri kanan"  
8     "footer footer";  
9   display: grid;  
10  grid-template-columns: 5fr 5fr;  
11  grid-template-rows: 2fr 4fr 5fr 2fr;  
12 }
```

Kode CSS di atas mendefinisikan tata letak (layout) dari suatu elemen dengan kelas .container menggunakan Grid Layout dalam CSS. Mari kita jabarkan secara rinci:

1. .container: Ini adalah pemilihan kelas CSS yang akan diberikan tata letak Grid. Semua elemen dengan kelas ini akan mengikuti aturan tata letak yang didefinisikan.
2. width: 100%; height: 20%;: Properti width mengatur lebar dari .container menjadi 100% lebar dari elemennya, sehingga elemen ini akan mengambil seluruh lebar yang tersedia. Sementara itu, properti height mengatur tinggi elemen menjadi 20% dari tinggi parent elemennya. Artinya, elemen ini akan memiliki tinggi yang relative terhadap parent elemennya.
3. grid-template-areas: Properti ini mendefinisikan susunan elemen-elemen dalam grid. Grid ini terdiri dari 4 area yang diberi nama: "header", "kiri", "kanan", dan "footer". Tanda kutip ganda digunakan untuk menyusun grid dengan baris-baris yang diberi nama dan elemen-elemen yang akan mengisi area tersebut.
4. display: grid;: Properti ini mendefinisikan bahwa elemen .container menggunakan Grid Layout untuk tata letaknya.
5. grid-template-columns: 5fr 5fr;: Properti ini mengatur jumlah dan lebar kolom dalam grid. Di sini, terdapat dua kolom yang memiliki lebar yang sama, yaitu 5fr. "fr" adalah singkatan dari "fraction," yang berarti bahwa kolom akan mendapatkan sejumlah fraksi yang sama dari lebar yang tersedia.
6. grid-template-rows: 2fr 4fr 5fr 2fr;: Properti ini mengatur jumlah dan tinggi baris dalam grid. Ada empat baris yang masing-masing memiliki tinggi yang berbeda. Seperti pada grid-template-columns, "fr" digunakan di sini untuk mendistribusikan tinggi berdasarkan fraksi.

Dengan demikian, kode di atas mendefinisikan suatu grid layout dengan 4 area, 2 kolom, dan 4 baris. Setiap area memiliki nama yang memudahkan penempatan elemen-elemen di dalamnya. Grid ini akan mengisi seluruh lebar parent elemennya dan tinggi 20% dari tinggi

parent elemennya, dengan kolom dan baris yang telah diatur sesuai dengan properti `grid-template-columns` dan `grid-template-rows`.

```
1  header {
2    grid-area: header;
3    top: 0;
4    left: 0;
5    z-index: 100;
6    color: rgb(184, 179, 179);
7    display: flex;
8    height: 12vh;
9    width: 100%;
10   background-color: rgba(0, 0, 0, 0.5);
11   justify-content: space-between;
12   align-items: center;
13 }
```

Kode CSS di atas adalah tag `header` dan disini saya telah menambahkan `grid area`, untuk penjelasannya dibawah ini:

1. `header`: Ini adalah pemilihan elemen HTML yang diberi gaya. Semua elemen `<header>` di dalam dokumen HTML akan mematuhi aturan tata letak dan gaya yang didefinisikan.
2. `grid-area: header;`: Properti `grid-area` menentukan bahwa elemen `<header>` akan mengisi area yang telah ditentukan dalam tata letak Grid. Dalam kasus ini, elemen `header` akan mengisi area "header" yang telah didefinisikan sebelumnya dalam grid utama yaitu container.
3. `top: 0; left: 0;`: Properti `top` dan `left` mengatur posisi elemen `<header>` di dalam area "header" grid. Dalam hal ini, elemen ini akan terletak di pojok kiri atas area "header" (0px dari atas dan 0px dari kiri).
4. `z-index: 100;`: Properti `z-index` mengontrol tumpukan atau hierarki tampilan elemen. Nilai 100 menunjukkan bahwa elemen `<header>` akan muncul di atas elemen lain yang memiliki nilai `z-index` yang lebih rendah.
5. `color: rgb(184, 179, 179);`: Properti ini mengatur warna teks dalam elemen `<header>`. Dalam hal ini, teks akan memiliki warna abu-abu.
6. `display: flex;`: Properti ini mengatur elemen `<header>` menggunakan tata letak Flexbox, yang memungkinkan pengaturan dan penempatan elemen-elemen anak di dalamnya dengan lebih fleksibel.
7. `height: 12vh;`: Properti ini mengatur tinggi elemen `<header>` sebesar 12% dari tinggi viewport (vh adalah satuan yang berkaitan dengan tinggi viewport).
8. `width: 100%;`: Properti ini mengatur lebar elemen `<header>` sebesar 100% dari lebar parent elemennya. Artinya, elemen ini akan mengisi seluruh lebar area "header" dalam grid.
9. `background-color: rgba(0, 0, 0, 0.5);`: Properti ini mengatur warna latar belakang elemen `<header>`. Warna latar belakang diatur sebagai `rgba` (red, green, blue, alpha), di mana nilai "0, 0, 0" adalah warna hitam (RGB), dan nilai "0.5" adalah tingkat transparansi (alpha) sebesar 50%. Sebagai hasilnya, elemen ini akan memiliki latar belakang hitam dengan tingkat transparansi sebanyak 50%, sehingga tampak setengah transparan.
10. `justify-content: space-between;`: Properti ini mengatur penyusunan elemen-elemen anak di dalam elemen `<header>` dengan jarak sebanyak mungkin antara elemen-elemen tersebut

sehingga ada ruang yang terisi di sisi kiri dan kanan elemen <header>. Ini menciptakan efek perataan ke kiri dan ke kanan.

11. align-items: center;; Properti ini mengatur penyusunan elemen-elemen anak di dalam elemen <header> secara vertikal dengan cara bahwa elemen-elemen anak berada di tengah-tengah elemen <header>.



```
1 .heropage {  
2   grid-area: kiri;  
3   color: white;  
4   margin-top: 100px;  
5   margin-left: 100px;  
6 }
```

Kode CSS di atas adalah definisi gaya untuk elemen HTML yang memiliki kelas CSS bernama "heropage". Berikut Penjelasan kode:

1. .heropage: Ini adalah sebuah selektor CSS yang digunakan untuk memilih elemen HTML yang memiliki atribut class="heropage". Semua elemen yang memiliki kelas CSS ini akan tunduk pada aturan tampilan yang didefinisikan dalam blok gaya ini.
2. grid-area: kiri;; Properti grid-area digunakan untuk menentukan bagian atau area dari tata letak Grid di mana elemen dengan kelas "heropage" akan ditempatkan. Dalam hal ini, elemen ini akan ditempatkan di dalam area dengan nama "kiri" yang telah didefinisikan sebelumnya dalam tata letak Grid. Ini mengatur tata letak elemen dalam grid secara spesifik.
3. color: white;; Properti ini mengatur warna teks dalam elemen dengan kelas "heropage" menjadi putih. Artinya, teks dalam elemen tersebut akan ditampilkan dengan warna putih.
4. margin-top: 100px;; Properti margin-top mengatur jarak antara elemen dengan elemen lain di atasnya. Dalam hal ini, elemen "heropage" akan memiliki jarak 100 piksel dari elemen di atasnya.
5. margin-left: 100px;; Properti margin-left mengatur jarak antara elemen dengan elemen lain di sebelah kiri. Dalam hal ini, elemen "heropage" akan memiliki jarak 100 piksel dari elemen di sebelah kirinya.

Penggunaan grid-area pada elemen dengan kelas "heropage" menunjukkan bahwa elemen ini telah dimasukkan ke dalam tata letak Grid dengan presisi tertentu, yang telah didefinisikan sebelumnya(container), sehingga elemen tersebut akan tampil sesuai dengan aturan tata letak yang telah ditentukan dalam tata letak Grid. Selain itu, pengaturan warna teks, margin atas, dan margin kiri akan memengaruhi penampilan dan posisi elemen "heropage" pada halaman web.



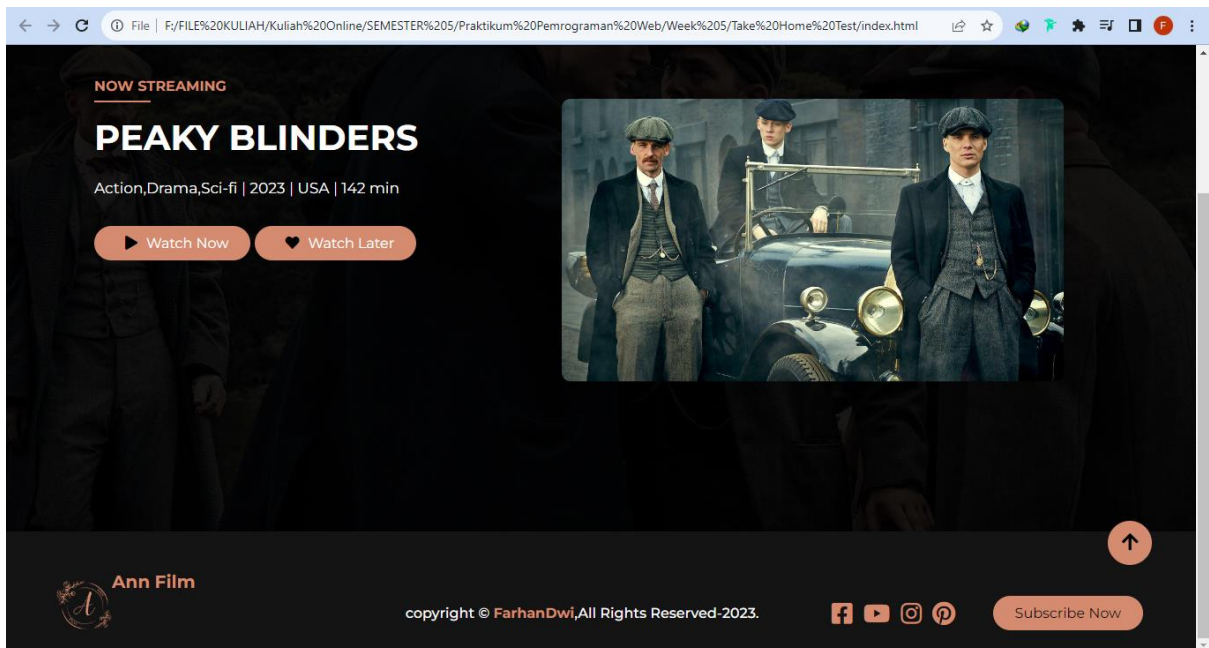
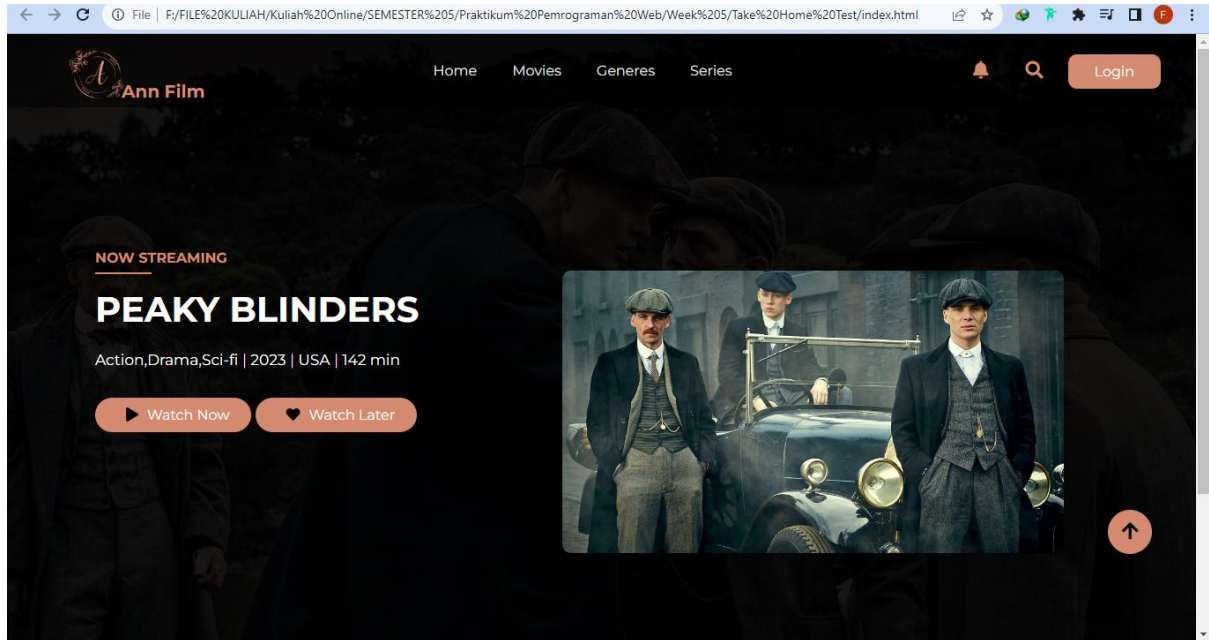
```
1 .footer {  
2   grid-area: footer;  
3   height: 20vh;  
4   color: white;  
5   background: #141414;  
6   width: 100%;  
7 }
```

Kode CSS di atas adalah definisi gaya untuk elemen HTML yang memiliki kelas CSS bernama "footer". Berikut Penjelasannya:

1. `.footer`: Ini adalah sebuah selektor CSS yang digunakan untuk memilih elemen HTML yang memiliki atribut `class="footer"`. Semua elemen yang memiliki kelas CSS ini akan tunduk pada aturan tampilan yang didefinisikan dalam blok gaya ini.
2. `grid-area: footer`:: Properti `grid-area` digunakan untuk menentukan bagian atau area dari tata letak Grid di mana elemen dengan kelas "footer" akan ditempatkan. Ini mengatur posisi elemen dalam grid dengan mengacu pada area "footer" yang telah didefinisikan sebelumnya dalam tata letak Grid.
3. `height: 20vh`:: Properti ini mengatur tinggi elemen "footer" menjadi 20% dari tinggi viewport (vh). Ini memastikan bahwa elemen "footer" akan memiliki tinggi yang sesuai relatif terhadap ukuran tampilan pengguna, dengan tinggi yang mendekati 20% dari tinggi tampilan.
4. `color: white`:: Properti ini mengatur warna teks dalam elemen "footer" menjadi putih. Artinya, teks dalam elemen tersebut akan ditampilkan dengan warna putih.
5. `background: #141414`:: Properti `background` mengatur warna latar belakang elemen "footer" menjadi abu-abu tua dengan kode warna #141414.
6. `width: 100%`:: Properti ini mengatur lebar elemen "footer" menjadi 100% dari lebar viewport. Ini memastikan bahwa elemen "footer" akan mengisi seluruh lebar viewport.

Penggunaan `grid-area: footer` pada elemen dengan kelas "footer" menunjukkan bahwa elemen ini telah dimasukkan ke dalam tata letak Grid dengan presisi tertentu, yang telah didefinisikan sebelumnya(container), sehingga elemen tersebut akan tampil sesuai dengan aturan tata letak yang telah ditentukan dalam tata letak Grid. Selain itu, pengaturan tinggi, warna teks, warna latar belakang, dan lebar akan memengaruhi penampilan elemen "footer" pada halaman web.

DOKUMENTASI



PUNUTUP

A. Kesimpulan

Dalam laporan praktikum ini, kita telah mengeksplorasi Display Grid, sebuah fitur kuat dalam CSS yang memungkinkan pengaturan tata letak elemen-elemen HTML secara lebih fleksibel dan efisien. Melalui serangkaian latihan dan contoh, kita telah memahami konsep dasar dari grid layout, mulai dari pendefinisian grid container hingga penempatan item-item di dalamnya.

Kita telah mempelajari properti-properti yang dapat mengontrol ukuran dan tata letak grid, seperti `grid-template-columns`, `grid-template-rows`, dan `grid-gap`. Selain itu, kita juga mendalami konsep penempatan elemen menggunakan `grid-row` dan `grid-column`, serta pengaturan aliran tata letak dengan `grid-auto-flow`.

Penting untuk diingat bahwa Display Grid adalah alat yang sangat bermanfaat dalam pengembangan web modern. Dengan pemahaman yang mendalam tentang grid layout, kita dapat menciptakan tampilan web yang responsif, estetis, dan mudah diatur. Ini akan memberikan pengalaman pengguna yang lebih baik dan memungkinkan kita untuk mengatasi tantangan tata letak dalam pengembangan web.

Selain itu, penggunaan Display Grid juga dapat mempercepat pengembangan web dengan menyederhanakan tata letak elemen-elemen dalam desain. Dengan begitu, kita dapat mengalokasikan waktu lebih banyak untuk aspek kreatif dari proyek dan meminimalkan kerumitan dalam menangani tata letak pada web.

B. Daftar Pustaka

skyridetim. (10 Februari 2020). CSS Grid Layout Module. [geeksforgeeks.org](https://www.geeksforgeeks.org/css-grid-layout-module/).
<https://www.geeksforgeeks.org/css-grid-layout-module/>

C. Link Github

<https://github.com/farhandwi/praktikum-web/tree/Modul-5>