

# CSS GRID

## Materi CSS Grid

### Nama Anggota Kelompok :

1. Ahsan Putra Ahyar
2. Muh Taufiq Adiguna
3. Nabil Shidiq

## XI RPL 1

---

## Fungsi CSS Grid

mempelajari CSS Grid membantu anda untuk membuat tata letak halaman web yang lebih kompleks dan responsif secara visual, beberapa fungsi dari mempelajari CSS Grid.

1. **Tata Letak yang Lebih Fleksibel:** Anda dapat membuat tata letak yang lebih kompleks dengan mudah, seperti tata letak dua dimensi dengan area grid yang terdiri dari beberapa kolom dan baris.
  2. **Responsif:** CSS Grid memungkinkan Anda membuat tata letak yang responsif tanpa harus tergantung pada teknik float atau clear. Anda dapat dengan mudah mengatur tata letak untuk berbagai perangkat dan ukuran layar.
  3. **Kode yang Lebih Bersih:** Dibandingkan dengan pendekatan tradisional menggunakan float dan positioning, CSS Grid dapat menghasilkan kode yang lebih bersih dan mudah dipahami.
  4. **Grid Lines dan Grid Tracks:** Anda dapat mengontrol grid lines dan grid tracks untuk menentukan posisi dan ukuran elemen di dalam grid dengan lebih presisi.
  5. **Overlap:** Anda dapat mengatur tata letak elemen-elemen yang saling tumpang tindih di dalam grid, yang sulit dicapai dengan pendekatan tradisional.
  6. **Grid Template Areas:** Anda dapat menggunakan fitur grid template areas untuk membuat tata letak dengan menyebutkan nama area, yang mempermudah dalam membuat dan memahami tata letak.
- 

## Property CSS Grid

### Container

# grid-template-columns

## Penjelasan

`grid-template-columns` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan lebar dari setiap kolom dalam grid. Properti ini memungkinkan Anda untuk secara fleksibel mendefinisikan lebar kolom dalam grid, baik secara absolut (dalam piksel, sentimeter, dll.) atau relatif (persentase dari lebar container atau fraksional).

## Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Template Columns</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 200px 100px; /* Kolom pertama 100px,
kolom kedua 200px, kolom ketiga 100px */
      gap: 10px; /* Jarak antara kolom */
    }
    .item {
      background-color: blue;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
</body>
</html>
```

## Hasil Program



# grid-template-row

## Penjelasan

`grid-template-rows` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan tinggi dari setiap baris dalam grid. Properti ini memungkinkan Anda untuk secara fleksibel mendefinisikan tinggi baris dalam grid, baik secara absolut (dalam piksel, sentimeter, dll.) atau relatif (persentase dari tinggi container atau fraksional).

## Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Template Rows</title>
  <style>
    .container {
      display: grid;
      grid-template-rows: 100px 200px 100px; /* Baris pertama 100px, baris
kedua 200px, baris ketiga 100px */
      gap: 10px; /* Jarak antara baris */
      height: 500px; /* Tinggi container */
    }
    .item {
      background-color: aqua;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
</body>
</html>
```

## Hasil Program

1

2

3

## grid-auto-columns

### Penjelasan

`grid-auto-columns` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan lebar default kolom untuk elemen-elemen dalam grid yang tidak memiliki ukuran kolom yang ditentukan secara eksplisit.

### Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Auto Columns</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px; /* Kolom pertama 100px */
      grid-auto-columns: 200px; /* Lebar default kolom untuk elemen-elemen
dalam grid */
      gap: 10px; /* Jarak antara kolom */
    }
    .item {
      background-color: #74f377;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
```

```
</body>
</html>
```

## Hasil Program



## grid-auto-rows

### Penjelasan

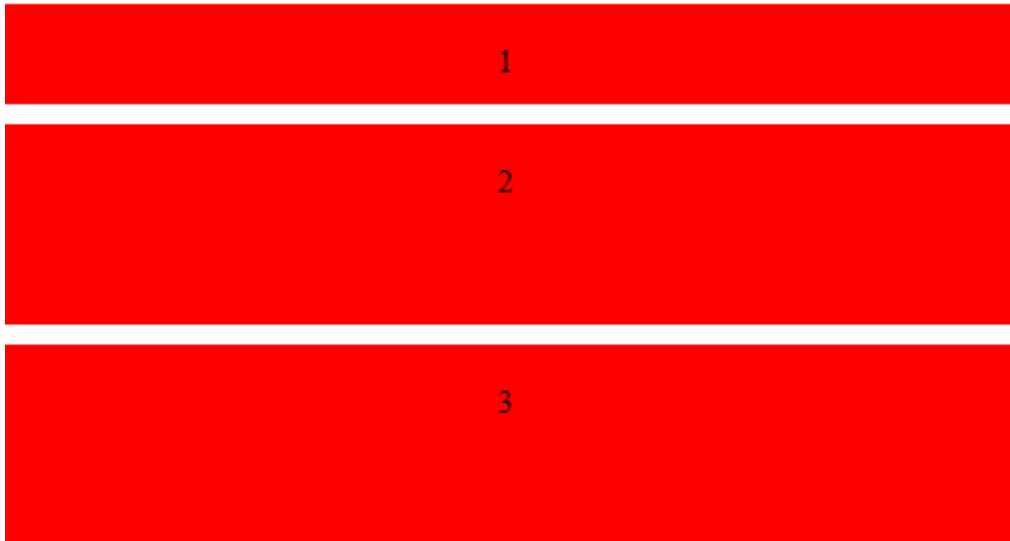
`grid-auto-rows` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan tinggi default baris untuk elemen-elemen dalam grid yang tidak memiliki ukuran baris yang ditentukan secara eksplisit.

### Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Auto Rows</title>
  <style>
    .container {
      display: grid;
      grid-template-rows: 100px; /* Baris pertama 100px */
      grid-auto-rows: 200px; /* Tinggi default untuk baris-baris dalam
grid */
      gap: 10px; /* Jarak antara baris */
    }
    .item {
      background-color: red;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
```

```
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
</body>
</html>
```

## Hasil Program



---

## grid-auto-flow

### Penjelasan

`grid-auto-flow` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan cara elemen-elemen baru ditempatkan dalam grid ketika elemen-elemen tersebut tidak memiliki area di grid yang didefinisikan secara eksplisit.

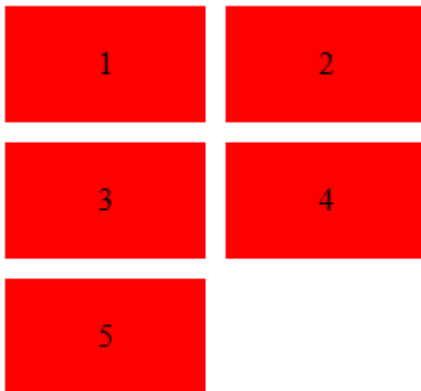
### Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Auto Flow</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      grid-auto-flow: row; /* Elemen-elemen baru ditempatkan dalam
```

```
baris baru */
    gap: 10px; /* Jarak antara elemen */
}

.item {
    background-color: #f0f0f0;
    padding: 20px;
    text-align: center;
}
</style>
</head>
<body>
    <div class="container">
        <div class="item">1</div>
        <div class="item">2</div>
        <div class="item">3</div>
        <div class="item">4</div>
        <div class="item">5</div>
    </div>
</body>
</html>
```

## Hasil Program



---

## grid-template-areas

### Penjelasan

`grid-template-areas` adalah properti dalam CSS Grid Layout yang memungkinkan Anda untuk secara visual mendefinisikan tata letak grid dengan menggunakan nama area. Dengan `grid-template-areas`, Anda dapat dengan mudah membuat tata letak grid yang kompleks dengan menetapkan nama untuk setiap area dalam grid dan kemudian mendefinisikan tata letak menggunakan nama-nama tersebut.

## Kode Program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Grid Template Areas</title>
  <style>
    .container {
      display: grid;
      grid-template-areas:
        "header header header"
        "sidebar content content"
        "footer footer footer"; /* Definisi grid template areas */
      grid-template-rows: auto; /* Tinggi baris disesuaikan dengan
konten */
      grid-template-columns: 200px 1fr 1fr; /* Lebar kolom pertama
200px, sisanya fr */
      gap: 10px; /* Jarak antara elemen */
    }

    .header {
      grid-area: header; /* Mengatur elemen dengan kelas .header di
area header */
      background-color: crimson;
      padding: 20px;
      text-align: center;
    }

    .sidebar {
      grid-area: sidebar; /* Mengatur elemen dengan kelas .sidebar di
area sidebar */
      background-color: crimson;
      padding: 20px;
      text-align: center;
    }

    .content {
      grid-area: content; /* Mengatur elemen dengan kelas .content di
area content */
      background-color: crimson;
      padding: 20px;
      text-align: center;
    }

    .footer {
      grid-area: footer; /* Mengatur elemen dengan kelas .footer di
area footer */
      background-color: crimson;
      padding: 20px;
```



```

        text-align: center;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="header">Header</div>
        <div class="sidebar">Sidebar</div>
        <div class="content">Content</div>
        <div class="footer">Footer</div>
    </div>
</body>
</html>

```

## Hasil Program



## grid-template

### Penjelasan

`grid-template` adalah singkatan dari gabungan tiga properti dalam CSS Grid Layout: `grid-template-rows`, `grid-template-columns`, dan `grid-template-areas`. Properti ini digunakan untuk mendefinisikan tata letak grid secara lengkap dalam satu deklarasi.

### Kode Program

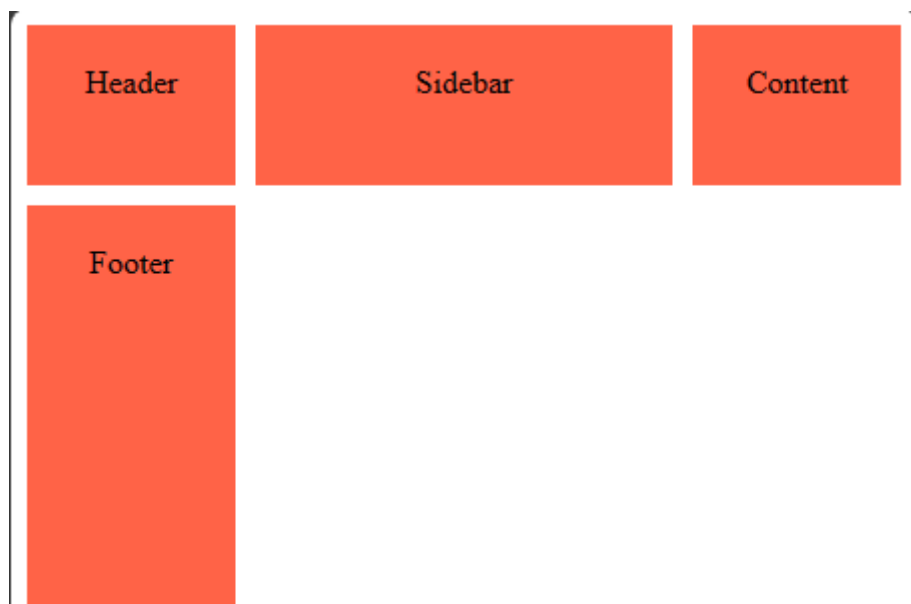
```

<!DOCTYPE html>
<html>
<head>
    <title>Grid Template</title>
    <style>
        .container {
            display: grid;
            grid-template:
                "header header header" 80px
                "sidebar content content" 200px
                "footer footer footer" 80px /
                1fr 2fr 1fr; /* Lebar kolom dalam fr */
            gap: 10px; /* Jarak antara sel */
        }
    </style>

```

```
        .header, .sidebar, .content, .footer {
            background-color: tomato;
            padding: 20px;
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="header">Header</div>
        <div class="sidebar">Sidebar</div>
        <div class="content">Content</div>
        <div class="footer">Footer</div>
    </div>
</body>
</html>
```

## Hasil Program



---

## grid-column-gap

### Penjelasan

`grid-column-gap` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan jarak antara kolom-kolom dalam grid. Properti ini mengatur jarak horizontal antara kolom-kolom dalam grid container.

### Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Column Gap</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      grid-column-gap: 20px; /* Jarak antara kolom */
    }

    .item {
      background-color: red;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
  </div>
</body>
</html>
```

## Hasil Program



# grid-row-gap

## Penjelasan

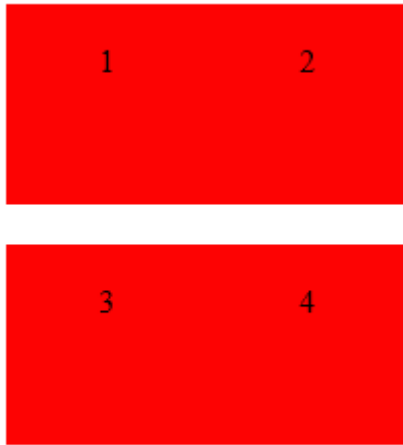
`grid-row-gap` adalah properti dalam CSS Grid Layout yang digunakan untuk menentukan jarak antara baris-baris dalam grid. Properti ini mengatur jarak vertikal antara baris-baris dalam grid container.

## Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid Row Gap </title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      grid-template-rows: 100px 100px; /* Dua baris dengan tinggi
100px */
      grid-row-gap: 20px; /* Jarak antara baris 20px */
    }

    .item {
      background-color: red;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
  </div>
</body>
</html>
```

## Hasil Program



---

## justify-items

### Penjelasan

`justify-items` adalah properti dalam CSS Grid Layout yang digunakan untuk mengatur penempatan elemen-elemen di dalam sel grid secara horizontal. Properti ini mengontrol penempatan elemen-elemen di sepanjang sumbu baris dalam setiap sel grid.

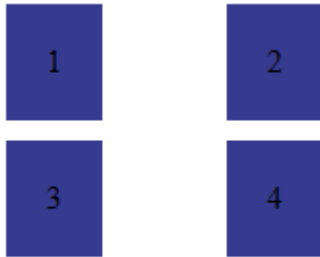
### Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Justify Items</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      justify-items: center; /* Elemen-elemen ditempatkan di tengah-
tengah kolom */
      gap: 10px; /* Jarak antara elemen */
    }

    .item {
      background-color: #363b8f;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
```

```
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</div>
</body>
</html>
```

## Hasil Program



---

## align-items

### Penjelasan

`align-items` adalah properti dalam CSS Grid Layout yang digunakan untuk mengatur penempatan elemen-elemen di dalam sel grid secara vertikal. Properti ini mengontrol penempatan elemen-elemen di sepanjang sumbu kolom dalam setiap sel grid.

### Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Align Items</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      align-items: center; /* Elemen-elemen akan ditempatkan di tengah
secara vertikal */
      height: 200px; /* Tinggi container */
    }

    .item {
```

```

        background-color: #755b5b;
        padding: 20px;
        text-align: center;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="item">1</div>
        <div class="item">2</div>
        <div class="item">3</div>
        <div class="item">4</div>
    </div>
</body>
</html>

```

## Hasil Program



## place-items

### Penjelasan

`place-items` adalah singkatan dari properti gabungan CSS Grid Layout yang digunakan untuk mengatur penempatan elemen-elemen di dalam sel grid secara horizontal dan vertikal secara bersamaan. Properti ini menggabungkan properti `justify-items` untuk mengatur penempatan horizontal dan `align-items` untuk mengatur penempatan vertikal.

### Kode Program

```

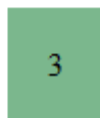
<!DOCTYPE html>
<html>
<head>
    <title>Place Items </title>
    <style>

```

```
.container {
    display: grid;
    place-items: center; /* Elemen-elemen ditempatkan di tengah
container secara horizontal dan vertikal */
    width: 300px;
    height: 300px;
}

.item {
    background-color: #7bb78d;
    padding: 20px;
    text-align: center;
}
</style>
</head>
<body>
    <div class="container">
        <div class="item">1</div>
        <div class="item">2</div>
        <div class="item">3</div>
    </div>
</body>
</html>
```

## Hasil Program



---

## justify-content

### Penjelasan



`justify-content` adalah properti dalam CSS Grid Layout yang digunakan untuk mengatur penempatan grid dalam container sepanjang sumbu baris. Properti ini mengontrol penempatan grid secara horizontal dalam container.

## Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Justify Content</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: repeat(3, 100px); /* Tiga kolom dengan
lebar 100px */
      justify-content: center; /* Grid ditempatkan di tengah-tengah
container secara horizontal */
      gap: 10px; /* Jarak antara kolom */
    }

    .item {
      background-color: #471c54;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
  </div>
</body>
</html>
```

## Hasil Program



---

## align-content

## Penjelasan

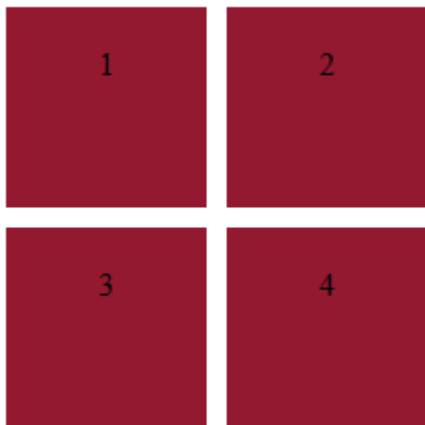
`align-content` adalah properti dalam CSS Grid Layout yang digunakan untuk mengatur penempatan grid dalam container sepanjang sumbu kolom. Properti ini mengontrol penempatan grid secara vertikal dalam container.

## Kode Program

```
<!DOCTYPE html>
<html>
<head>
  <title>Align Content</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      grid-template-rows: 100px 100px; /* Dua baris dengan tinggi
100px */
      align-content: center; /* Menengahkan grid secara vertikal dalam
container */
      gap: 10px; /* Jarak antara elemen */
      height: 300px; /* Tinggi container */
    }

    .item {
      background-color: #931931;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
  </div>
</body>
</html>
```

## Hasil Program



---

## place-content

### Penjelasan

`place-content` adalah singkatan dari properti gabungan CSS Grid Layout yang digunakan untuk mengatur penempatan grid dalam container secara horizontal dan vertikal secara bersamaan. Properti ini menggabungkan properti `justify-content` untuk mengatur penempatan horizontal dan `align-content` untuk mengatur penempatan vertikal.

### Kode Program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Place Content</title>
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Dua kolom dengan lebar
100px */
      grid-template-rows: 100px 100px; /* Dua baris dengan tinggi
100px */
      place-content: center; /* Elemen-elemen ditempatkan di tengah
container secara horizontal dan vertikal */
      gap: 10px; /* Jarak antara elemen */
      height: 300px; /* Tinggi container */
    }

    .item {
      background-color: #f0f0f0;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
  </div>
</body>
</html>
```

```
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
  </div>
</body>
</html>
```

## Hasil Program



---

## Item

### Grid-column-start

#### Penjelasan

Property `grid-column-start` dalam CSS digunakan untuk menentukan awal dari suatu grid item di dalam grid container pada sumbu horizontal. Properti ini mengatur posisi awal grid item berdasarkan garis grid di sepanjang sumbu horizontal, dimulai dari garis pertama (1) hingga garis terakhir (tergantung dari jumlah kolom yang telah didefinisikan). Anda bisa menggunakan nilai numerik atau nama dari garis grid sebagai nilai dari properti ini. Misalnya, jika Anda ingin sebuah item grid dimulai dari garis kedua, Anda dapat menetapkan nilai `grid-column-start: 2;`

#### kode program

```
<!DOCTYPE html>
<html lang="en">
  <head>
```

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Grid-column-end Example</title>
<style>
  .container {
    display: grid;
    grid-template-columns: repeat(3, 100px);
    grid-gap: 10px;
  }
  .item {
    background-color: #ccc;
    padding: 20px;
    text-align: center;
  }
  .item:nth-child(1) {
    grid-column-start: 1;
  }
  .item:nth-child(2) {
    grid-column-start: 2;
  }
  .item:nth-child(3) {
    grid-column-start: 3;
  }
  .item:nth-child(4) {
    grid-column-start: 1;
  }
  .item:nth-child(5) {
    grid-column-start: 2;
  }
  .item:nth-child(6) {
    grid-column-start: 3;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
    <div class="item">6</div>
  </div>
</body>
</html>
```

## Hasil Program

1	2	3
4	5	6

---

## Grid-column-end

### Penjelasan

Property `grid-column-end` digunakan dalam CSS Grid Layout untuk menentukan kolom terakhir tempat suatu item grid akan ditempatkan. Properti ini bekerja bersama dengan `grid-column-start` untuk menentukan rentang kolom yang diinginkan untuk suatu item.

Beberapa hal yang perlu diingat tentang `grid-column-end`:

- **Nilai Angka:** Anda dapat menetapkan nilai angka untuk `grid-column-end`, yang menunjukkan nomor kolom terakhir di mana item akan ditempatkan.
- **Keyword:** Selain nilai angka, Anda juga dapat menggunakan kata kunci seperti `span`, yang menentukan jumlah kolom yang akan ditempati oleh suatu item.
- **Menentukan Rentang Kolom:** Anda juga bisa menggunakan `grid-column-end` untuk menentukan rentang kolom di mana suatu item akan ditempatkan. Misalnya, `grid-column-end: span 2;` akan membuat item menempati dua kolom.

### Kode Program

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid-column-end Example</title>
    <style>
      .container {
        display: grid;
        grid-template-columns: repeat(3, 100px);
        grid-gap: 10px;
      }
      .item {
        background-color: #ccc;
        padding: 20px;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
      <div class="item">5</div>
      <div class="item">6</div>
    </div>
  </body>
</html>
```

```
.item:nth-child(1) {  
  grid-column-start: 1;  
}  
.item:nth-child(2) {  
  grid-column-start: 2;  
}  
.item:nth-child(3) {  
  grid-column-start: 3;  
}  
.item:nth-child(4) {  
  grid-column-start: 1;  
}  
.item:nth-child(5) {  
  grid-column-start: 2;  
}  
.item:nth-child(6) {  
  grid-column-start: 3;  
}  
</style>  
</head>  
<body>  
  <div class="container">  
    <div class="item">1</div>  
    <div class="item">2</div>  
    <div class="item">3</div>  
    <div class="item">4</div>  
    <div class="item">5</div>  
    <div class="item">6</div>  
  </div>  
</body>  
</html>
```

## Hasil Program

1	2	3
4	5	6

---

## Grid-row-start

### Penjelasan

Property `grid-row-start` digunakan dalam CSS Grid Layout untuk menentukan baris pertama tempat suatu item grid akan ditempatkan. Properti ini bekerja bersama dengan `grid-column-start` untuk menentukan posisi awal suatu item dalam grid.

Beberapa hal yang perlu diingat tentang `grid-row-start` :

- **Nilai Angka:** Anda dapat menetapkan nilai angka untuk `grid-row-start` , yang menunjukkan nomor baris pertama di mana item akan ditempatkan.
- **Keyword:** Selain nilai angka, Anda juga dapat menggunakan kata kunci seperti `span` , yang menentukan jumlah baris yang akan ditempati oleh suatu item.
- **Menentukan Rentang Baris:** Anda juga bisa menggunakan `grid-row-start` untuk menentukan rentang baris di mana suatu item akan ditempatkan. Misalnya, `grid-row-start: span 2;` akan membuat item menempati dua baris.

## Kode Program

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid-row-start Example</title>
    <style>
      .container {
        display: grid;
        grid-template-columns: repeat(3, 100px);
        grid-gap: 10px;
      }
      .item {
        background-color: #ccc;
        padding: 20px;
        text-align: center;
      }
      .item:nth-child(1) {
        grid-row-start: 1; /* Item dimulai dari baris ke-1 */
      }
      .item:nth-child(2) {
        grid-row-start: 2; /* Item dimulai dari baris ke-2 */
      }
      .item:nth-child(3) {
        grid-row-start: 1; /* Item dimulai dari baris ke-1 */
      }
      .item:nth-child(4) {
        grid-row-start: 3; /* Item dimulai dari baris ke-3 */
      }
      .item:nth-child(5) {
        grid-row-start: 2; /* Item dimulai dari baris ke-2 */
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
      <div class="item">5</div>
    </div>
  </body>
</html>
```



```

        .item:nth-child(6) {
            grid-row-start: 3; /* Item dimulai dari baris ke-3 */
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="item">1</div>
        <div class="item">2</div>
        <div class="item">3</div>
        <div class="item">4</div>
        <div class="item">5</div>
        <div class="item">6</div>
    </div>
</body>
</html>

```

## Hasil Program

1	3
2	5
4	6

## Grid-row-end

### Penjelasan

Property `grid-row-end` digunakan dalam CSS Grid Layout untuk menentukan baris terakhir tempat suatu item grid akan ditempatkan. Properti ini bekerja bersama dengan `grid-row-start` untuk menentukan rentang baris yang diinginkan untuk suatu item.

Beberapa hal yang perlu diingat tentang `grid-row-end`:

- Nilai Angka: Anda dapat menetapkan nilai angka untuk `grid-row-end`, yang menunjukkan nomor baris terakhir di mana item akan ditempatkan.
- Keyword: Selain nilai angka, Anda juga dapat menggunakan kata kunci seperti `span`, yang menentukan jumlah baris yang akan ditempati oleh suatu item.

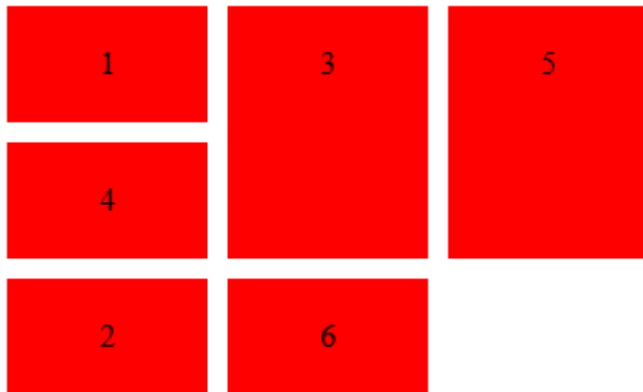
- Menentukan Rentang Baris: Anda juga bisa menggunakan grid-row-end untuk menentukan rentang baris di mana suatu item akan ditempatkan. Misalnya, grid-row-end: span 2; akan membuat item menempati dua baris.

## kode program

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid-row-end Example</title>
    <style>
      .container {
        display: grid;
        grid-template-columns: repeat(3, 100px);
        grid-gap: 10px;
      }
      .item {
        background-color: red;
        padding: 20px;
        text-align: center;
      }
      .item:nth-child(1) {
        grid-row-end: 2; /* Item berakhir di baris ke-2 */
      }
      .item:nth-child(2) {
        grid-row-end: 4; /* Item berakhir di baris ke-4 */
      }
      .item:nth-child(3) {
        grid-row-end: span 2; /* Item menempati 2 baris */
      }
      .item:nth-child(4) {
        grid-row-end: 3; /* Item berakhir di baris ke-3 */
      }
      .item:nth-child(5) {
        grid-row-end: span 2; /* Item menempati 2 baris */
      }
      .item:nth-child(6) {
        grid-row-end: 4; /* Item berakhir di baris ke-4 */
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
```

```
<div class="item">5</div>
<div class="item">6</div>
</div>
</body>
</html>
```

## Hasil Program



---

## Grid-column

### Penjelasan

Property `grid-column` adalah shorthand property dalam CSS Grid Layout yang digunakan untuk menentukan baik `grid-column-start` maupun `grid-column-end` secara bersamaan dalam satu aturan CSS. Ini memungkinkan untuk mengatur posisi horizontal suatu item dalam grid dengan lebih mudah.

Nilai yang diterima oleh `grid-column` bisa berupa:

- Single Number: Menentukan kolom di mana item akan ditempatkan. Ini setara dengan menentukan baik `grid-column-start` maupun `grid-column-end` dengan nilai yang sama.
- Two Numbers: Menentukan `grid-column-start` dan `grid-column-end` secara terpisah.
- Single Keyword: Menentukan kata kunci seperti `span` untuk menentukan jumlah kolom yang akan ditempati oleh suatu item.
- Two Keywords: Kombinasi dari dua kata kunci yang menentukan `span` untuk `grid-column-start` dan `grid-column-end`.

### Kode Program

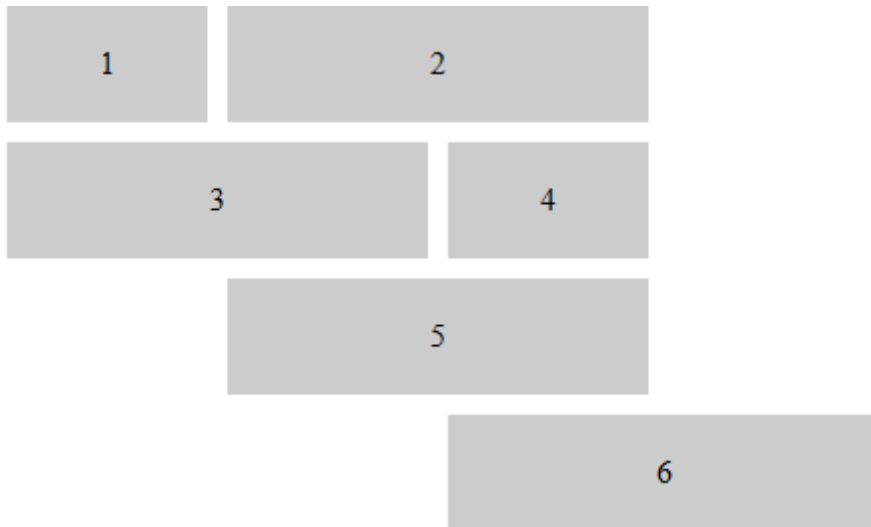
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Grid-column Example</title>
<style>
  .container {
    display: grid;
    grid-template-columns: repeat(3, 100px);
    grid-gap: 10px;
  }
  .item {
    background-color: #ccc;
    padding: 20px;
    text-align: center;
  }
  .item:nth-child(1) {
    grid-column: 1; /* Item dimulai dari kolom ke-1 dan berakhir di
kolom ke-1 */
  }
  .item:nth-child(2) {
    grid-column: 2 / 4; /* Item dimulai dari kolom ke-2 dan berakhir di
kolom ke-4 */
  }
  .item:nth-child(3) {
    grid-column: span 2; /* Item menempati 2 kolom (sama dengan grid-
column-start: span 2;) */
  }
  .item:nth-child(4) {
    grid-column: 3; /* Item dimulai dari kolom ke-3 dan berakhir di
kolom ke-3 */
  }
  .item:nth-child(5) {
    grid-column: span 2 / 4; /* Item dimulai dari kolom dengan lebar 2
dan berakhir di kolom ke-4 */
  }
  .item:nth-child(6) {
    grid-column: 3 / span 2; /* Item dimulai dari kolom ke-3 dan
menempati 2 kolom ke depan */
  }
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
    <div class="item">6</div>
  </div>
</body>
</html>

```

## Hasil Program



## Grid-row

### Penjelasan

Property `grid-row` adalah shorthand property dalam CSS Grid Layout yang digunakan untuk menentukan baik `grid-row-start` maupun `grid-row-end` secara bersamaan dalam satu aturan CSS. Ini memungkinkan untuk mengatur posisi vertikal suatu item dalam grid dengan lebih mudah.

Nilai yang diterima oleh `grid-row` bisa berupa:

- Single Number: Menentukan baris di mana item akan ditempatkan. Ini setara dengan menentukan baik `grid-row-start` maupun `grid-row-end` dengan nilai yang sama.
- Two Numbers: Menentukan `grid-row-start` dan `grid-row-end` secara terpisah.
- Single Keyword: Menentukan kata kunci seperti `span` untuk menentukan jumlah baris yang akan ditempati oleh suatu item.
- Two Keywords: Kombinasi dari dua kata kunci yang menentukan span untuk `grid-row-start` dan `grid-row-end`

### Kode Program

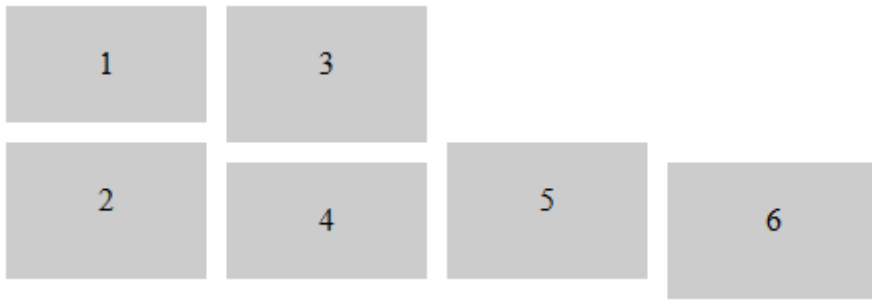
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid-row Example</title>
    <style>
```

```

.container {
  display: grid;
  grid-template-columns: repeat(3, 100px);
  grid-gap: 10px;
}
.item {
  background-color: #ccc;
  padding: 20px;
  text-align: center;
}
.item:nth-child(1) {
  grid-row: 1; /* Item dimulai dari baris ke-1 dan berakhir di baris
ke-1 */
}
.item:nth-child(2) {
  grid-row: 2 / 4; /* Item dimulai dari baris ke-2 dan berakhir di
baris ke-4 */
}
.item:nth-child(3) {
  grid-row: span 2; /* Item menempati 2 baris (sama dengan grid-row-
start: span 2;) */
}
.item:nth-child(4) {
  grid-row: 3; /* Item dimulai dari baris ke-3 dan berakhir di baris
ke-3 */
}
.item:nth-child(5) {
  grid-row: span 2 / 4; /* Item dimulai dari baris dengan tinggi 2 dan
berakhir di baris ke-4 */
}
.item:nth-child(6) {
  grid-row: 3 / span 2; /* Item dimulai dari baris ke-3 dan menempati
2 baris ke depan */
}
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
    <div class="item">6</div>
  </div>
</body>
</html>

```

## Hasil Program



## Grid area

### Penjelasan

Property `grid-area` adalah bagian dari CSS Grid Layout Module yang digunakan untuk menentukan letak dan ukuran dari sebuah grid item di dalam sebuah grid container.

Property ini menggabungkan properti `grid-row-start`, `grid-column-start`, `grid-row-end`, dan `grid-column-end` menjadi satu deklarasi singkat.

Dengan menggunakan `grid-area`, kita dapat menentukan nama area yang akan diduduki oleh sebuah grid item di dalam grid container, sehingga memudahkan pengaturan layout secara lebih terstruktur dan mudah dibaca.

### Kode Program

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8"> <meta name="viewport" content="width=device-width,
initial scale=1.0">
<title>Grid Area Example</title>
<style>
.container {
  display: grid;
  grid-template-columns: 100px 100px;
  grid-template-rows: 100px 100px;
  gap: 10px;
}
.item1 {
  grid-area: 1 / 1 / 2 / 3;
  background-color: aqua;
}
.item2 {
  grid-area: 1 / 2 / 3 / 3;
}
.item3 {
  grid-area: 2 / 1 / 3 / 2;
```

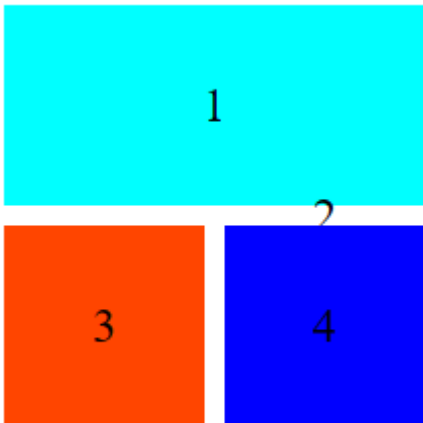
```

background-color: orangered;
}
.item4 {
  grid-area: 2 / 2 / 3 / 3;
  background-color: blue;
}
/* Styling untuk visualisasi */
.container > div {
  background-color: #ccc;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 24px;
}
</style>
<body>
  <div class="container">
    <div class="item1">1</div>
    <div class="item2">2</div>
    <div class="item3">3</div>
    <div class="item4">4</div>
  </div>
</body>
</html>

```

## Hasil Program

---



## Justify-self

### Penjelasan

Property `justify-self` adalah bagian dari CSS Grid Layout Module yang digunakan untuk mengatur penempatan horizontal (justify) dari sebuah grid item di dalam sebuah grid cell. Property ini mengontrol posisi horizontal dari grid item di dalam sel grid yang dimilikinya.



Nilai yang dapat diberikan kepada justify-self antara lain:

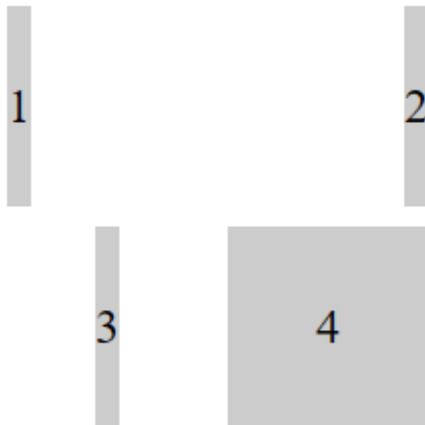
- `start` : Grid item akan diposisikan di sisi awal (kiri) dari sel grid yang dimilikinya.
- `end` : Grid item akan diposisikan di sisi akhir (kanan) dari sel grid yang dimilikinya.
- `center` : Grid item akan diposisikan di tengah-tengah secara horizontal di dalam sel grid yang dimilikinya.
- `stretch` (default): Grid item akan meregang secara horizontal untuk mengisi sel grid yang dimilikinya.

## Kode Program

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Justify Self Example</title>
    <style>
      .container {
        display: grid;
        grid-template-columns: repeat(2, 100px);
        grid-template-rows: repeat(2, 100px);
        gap: 10px;
      }
      .item {
        background-color: #ccc;
        display: flex;
        justify-content: center;
        align-items: center;
        font-size: 24px;
      }
      .item1 {
        justify-self: start;
      }
      .item2 {
        justify-self: end;
      }
      .item3 {
        justify-self: center;
      }
      .item4 {
        /* menggunakan nilai default, yaitu stretch */
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="item item1">1</div>
```

```
<div class="item item2">2</div>
<div class="item item3">3</div>
<div class="item item4">4</div>
</div>
</body>
</html>
```

## Hasil Program



## Align-self

### Penjelasan

Property `align-self` adalah bagian dari CSS Grid Layout Module yang digunakan untuk mengatur penempatan vertikal (align) dari sebuah grid item di dalam sebuah grid cell. Property ini mengontrol posisi vertikal dari grid item di dalam sel grid yang dimilikinya.

Nilai yang dapat diberikan kepada `align-self` antara lain:

- `start` : Grid item akan diposisikan di sisi awal (atas) dari sel grid yang dimilikinya.
- `end` : Grid item akan diposisikan di sisi akhir (bawah) dari sel grid yang dimilikinya.
- `center` : Grid item akan diposisikan di tengah-tengah secara vertikal di dalam sel grid yang dimilikinya.
- `stretch` (default) : Grid item akan meregang secara vertikal untuk mengisi sel grid yang dimilikinya

### Kode Program

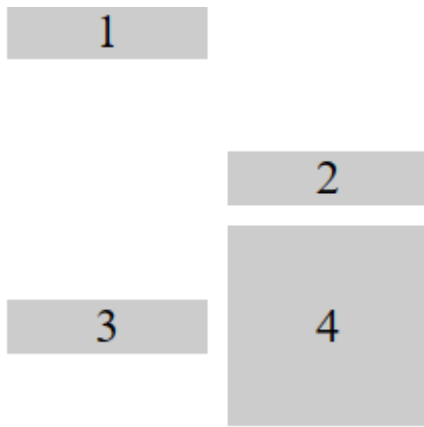
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>Align Self Example</title>
<style>
    .container {
        display: grid;
        grid-template-columns: repeat(2, 100px)
        grid-template-rows: repeat(2, 100px);
        gap: 10px;
    }
    .item {
        background-color: #ccc;
        display: flex;
        justify-content: center;
        align-items: center;
        font-size: 24px;
    }
    .item1 {
        align-self: start;
    }
    .item2 {
        align-self: end;
    }
    .item3 {
        align-self: center;
    }
    .item4 {
        /* menggunakan nilai default, yaitu stretch */
    }
</style>
</head>
<body>
    <div class="container">
        <div class="item item1">1</div>
        <div class="item item2">2</div>
        <div class="item item3">3</div>
        <div class="item item4">4</div>
    </div>
</body>
</html>

```

## Hasil Program



## Place-self

### Penjelasan

Property `place-self` adalah singkatan dari kombinasi dua property CSS Grid Layout, yaitu `align-self` dan `justify-self`. Property ini digunakan untuk secara bersamaan mengatur penempatan vertikal dan horizontal dari sebuah grid item di dalam sebuah grid cell.

Nilai yang dapat diberikan kepada `place-self` mirip dengan nilai yang dapat diberikan kepada `align-self` dan `justify-self`. Beberapa nilai umumnya antara lain:

- `start` : Grid item akan diposisikan di sisi awal (atas dan kiri) dari sel grid yang dimilikinya.
- `end` : Grid item akan diposisikan di sisi akhir (bawah dan kanan) dari sel grid yang dimilikinya.
- `center` : Grid item akan diposisikan di tengah-tengah secara vertikal dan horizontal di dalam sel grid yang dimilikinya.
- `stretch` (default) : Grid item akan meregang secara vertikal dan horizontal untuk mengisi sel grid yang dimilikinya.

### Kode Program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Place Self Example</title>
<style>
  .container {
    display: grid;
    grid-template-columns: repeat(2, 100px);
    grid-template-rows: repeat(2, 100px);
```

```

        gap: 10px;
    }
    .item {
        background-color: #ccc;
        display: flex;
        justify-content: center;
        align-items: center;
        font-size: 24px;
    }
    .item1 {
        place-self: start start;
    }
    .item2 {
        place-self: end end;
    }
    .item3 {
        place-self: center center;
    }
    .item4 {
        /* menggunakan nilai default, yaitu stretch stretch */
    }
</style>
</head>
<body>
    <div class="container">
        <div class="item item1">1</div>
        <div class="item item2">2</div>
        <div class="item item3">3</div>
        <div class="item item4">4</div>
    </div>
</body>
</html>

```

## Hasil Program

1

2

3

4



# Implementasi

## Penjelasan

Web ini dibuat menggunakan Grid dengan memakai `grid-template-areas` sebagai tempat atau posisi elemen ingin di simpan **Item1** : Bagian header halaman web dengan judul "KACAMATA SASHA". **Item2** : Bagian navbar dengan dua paragraf teks yang menjelaskan tentang perlindungan dan gaya dari kaca mata Transitions. Ini juga memiliki grid layout untuk menyusun dua paragraf dalam dua kolom. **Item3** : Bagian utama dari halaman web dengan latar belakang gambar (bg.jpeg) dan teks selamat datang. **Item4** : Bagian berita yang berisi manfaat penggunaan kaca mata dalam bentuk daftar. **Item5 dan Item6** : Dua bagian konten yang masing-masing menampilkan judul "ARTIKEL", gambar (bg2.jpeg), dan harga. **Item7** : Bagian footer yang berisi logo, informasi kontak seperti nomor WhatsApp, akun Instagram, dan alamat.

## Kode Program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    .container {
      display: grid;
      grid-template-areas:
        "header header header header"
        "main main navbar navbar"
        "main main berita content1"
        "main main berita content2"
        "footer footer footer footer";
      grid-template-columns:265px 350px 265px 350px 265px;
      grid-template-rows: 70px 250px 250px 250px 250px;
      gap: 15px;
    }

    .item1 {
      grid-area: header;
      background-color:grey;
      text-align: center;
    }

    .item2 {
      grid-area: navbar;
      background-color:white;
      display:grid ;
      padding: 20px;
      grid-template-columns:50% 50%;
```

```
    grid-auto-flow: row;
    border: 3px solid gray;
    border-radius: 10px;
}

.teks-1 {
    padding: 20px;
    line-height: 25px;
}

.teks-2 {
    padding: 20px;
    line-height: 25px;
}

.judul {
    font-size: 18px;
    font-weight: bold;
}

.item3 {
    grid-area: main;
    background-image: url(bg.jpeg);
    background-size: cover;
    background-position: center;
    border-radius: 20px;
    line-height: 35px;
}

.head {
    font-size: 35px;
    color: whitesmoke;
    font-weight: bold;
    line-height: 30px;
}

.teks {
    padding: 30px;
    color: whitesmoke;
    font-size: 20px;
    font-family: Arial, Helvetica, sans-serif;
    padding-top: 150px;
}

.item4 {
    grid-area: berita;
    border-radius: 10px;
    border: 3px solid gray;
    line-height: 25px;
    padding: 10px;
```





```

        <p class="teks-1"><span class="judul">PERLINDUNGAN
        TANPA HENTI.</span><br>
        Lensa Transitions® menghalangi 100% sinar UVA dan UVB serta
        menyaring sinar biru-violet* di dalam dan di luar ruangan.</p>
        <p class="teks-2"><span class="judul">PENAMBAH
        GAYA.</span><br>
        Choose from a wide range of lens colors and express your
        personal style with Transitions lenses tailored to your
        eyecare needs.</p>
    </div>

    <div class="item3">
        <p class="teks">
            <span class="head">Selamat Datang<br> di Web Kacamata
            Optik</span>
            <br>
            Konsultasi Sekarang dengan dokter kami tentang Mata Anda!!
        </p>
    </div>

    <div class="item4">
        <h2 class="judul-berita">Manfaat Penggunaan kacamata</h2>
        <ul class="">
            <li>Mencegah kelelahan mata dan sakit kepala saat membaca atau
            bekerja di depan komputer.</li>
            <li>Memperbaiki penglihatan seperti presbiopi,
            miopi, hipermetropi</li>
            <li>Mencegah kerusakan mata lebih lanjut.</li>
            <li>Melindungi mata dari debu, polusi, dan paparan sinar
            matahari</li>
            <li>Membuat penglihatan tetap nyaman dan melindungi kulit di
            sekitar mata.</li>
        </ul>
    </div>

    <div class="item5">
        <h3>ARTIKEL</h3>
        
        <h4 class="harga">Price : Rp.250.000</h4>
    </div>

    <div class="item6">
        <h3>ARTIKEL</h3>
        
        <h4 class="harga">Price : Rp.250.000</h4>
    </div>

    <div class="item7">
        
        <div class="informasi">

```


```

<p>WhatsApp : +62 821-9700-0975</p>
<p>Instagram : kacamata_sasha</p>
<p>Alamat : Jl.Moh. Yamin no.9</p>
</div>
</div>
</div>
</body>
</html>

```

## Hasil Program

KACAMATA SASHA



**Selamat Datang  
di Web Kacamata Optik**

Konsultasi Sekarang dengan dokter kami tentang Mata Anda!!

Better  
Sight  
Better  
Style

**PERLINDUNGAN TANPA HENTI.**

Lensa Transitions® menghalangi 100% sinar UVA dan UVB serta menyaring sinar biru-violet\* di dalam dan di luar ruangan.


**PENAMBAH GAYA.**

Choose from a wide range of lens colors and express your personal style with Transitions lenses tailored to your eyecare needs.

**Manfaat Penggunaan kacamata**


- Mencegah kelelahan mata dan sakit kepala saat membaca atau bekerja di depan komputer.
- Memperbaiki penglihatan seperti presbiopi, miopi, hipermetropi.
- Mencegah kerusakan mata lebih lanjut.
- Melindungi mata dari debu, polusi, dan paparan sinar matahari.
- Membuat penglihatan tetap nyaman dan melindungi kulit di sekitar mata.

**ARTIKEL**




Price : Rp.250.000

**ARTIKEL**



Price : Rp.250.000



WhatsApp : +62 821-9700-0975

Instagram : kacamata\_sasha

Alamat : Jl.Moh. Yamin no.9