

Nama : Ade Amanah Putri  
NIM : K3524018  
Kelas : D  
Mata Kuliah: DESAIN WEB

## LAPORAN PRAKTIKUM DESAIN WEB 4

### A. JUDUL

Responsive Web Design (RWD)

### B. TUJUAN PRAKTIKUM

1. Mampu menerapkan konsep desain responsive ke dalam website
2. Mampu menghasilkan desain web responsive
3. Dalap menerapkan HTML dan CSS tingkat lanjut

### C. ALAT DAN BAHAN

- Laptop/PC
- Text Editor seperti VS Code
- Web Browser
- Logo uns PNG
- internet

### D. RINGKASAN MATERI

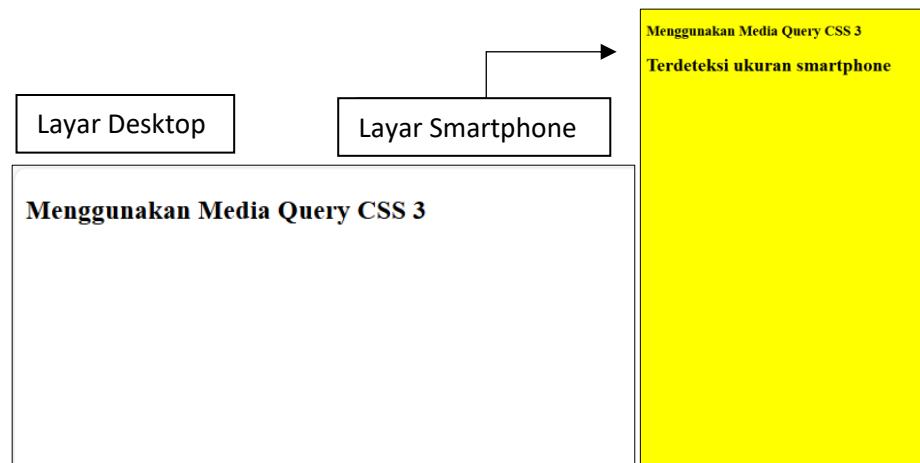
1. Desain web responsive merupakan konsep pengembangan web yang memungkinkan layout menyesuaikan dengan resolusi layer pengguna sehingga pengguna tidak harus menyeuaikan perangkat yang digunakan.
2. RWD direalisasikan melalui CSS3 dengan media query yaitu filter yang akan diberlakukan terhadap halaman web sesuai identifikasi browser.
3. Langkah pengembangan RWD dibagi menjadi 3, yaitu:
  - a. Mendefinisikan Meta Tag Viewpoint

Meta tag viewpoint digunnakan untuk mereset ulang dan untuk memberitahukan kepada browser agar menggunakan perangkat sebagai acuan lebar viewpoint serta menonaktifkan skala awal. Didalamnya ada beberapa property seperti width, height, initial-scale, minimum-scale, maximum-scale, dan user-scalable.

- b. Menentukan layout dan struktur HTML  
Merencanakan tinggi dan lebar elemen sesuai kebutuhan. 1024(desktop), 768 (tablet), atau 420 (smartphone).
  - c. Menetapkan resolusi layer  
Menetapkan resolusi sesuai standart(aspect ratio) untuk memudahkan dalam menentukan resolusi.
4. Teknik membuat RWD dengan media query css untuk memerintahkan browser mengikuti ukuran sesuai script yang ditulis. Menggunakan pendekatan graceful degradation jika menulis media query dari tampilan desktop dulu, dan menggunakan pendekatan mobile first jika menulis media query dari tampilan mobile dulu.
  5. Flex Box CSS yaitu model layout 1 dimensi yang mengatur jarak dan alignment antar item dalam sebuah container. Dengan beberapa elemen didalamnya seperti: container, items (box), main axis, cross axis, main size, cross size, main start/end, dan cross start/end.
  6. Flex-direction untuk menentukan item diurutkan ke arah mana. Flex-wrap membuat item pada satu baris pindah ke baris berikutnya. Justify-content untuk penataan antar items terhadap main axis, dan terakhir ada align-items yang membantu pengaturan kesejajaran antar item.

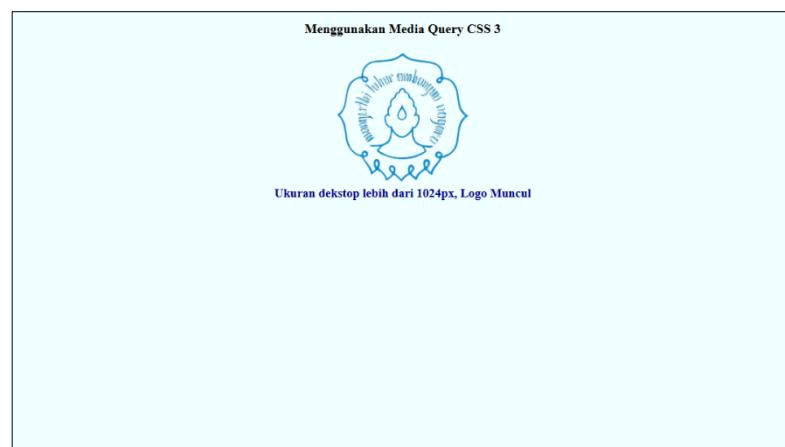
## E. OUTPUT PROGRAM

### Latihan 1a:



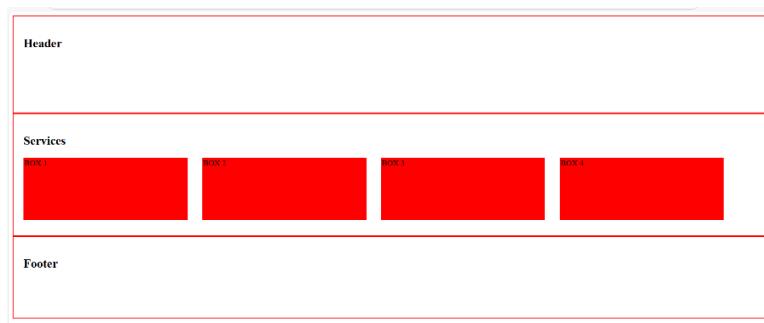
### Latihan 1b:

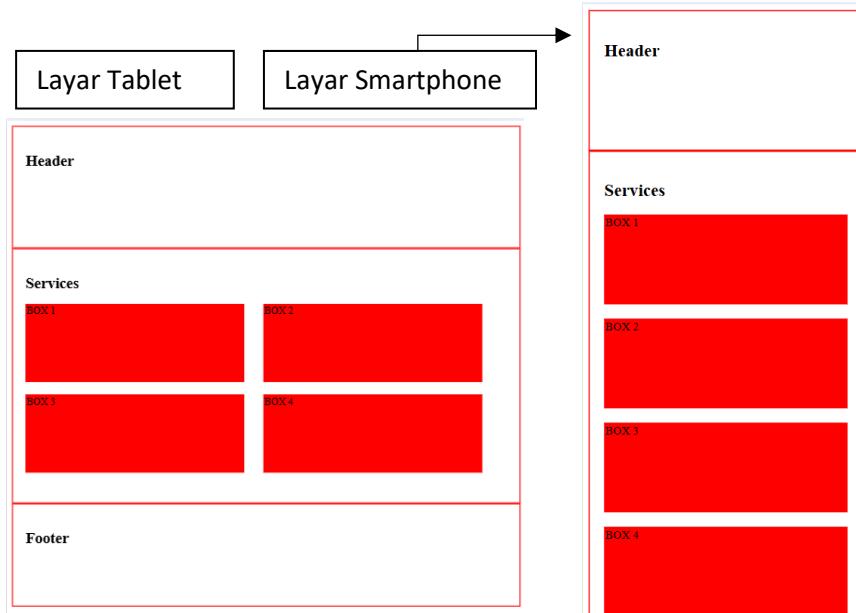
Saat ukuran layar lebih dari 1024px



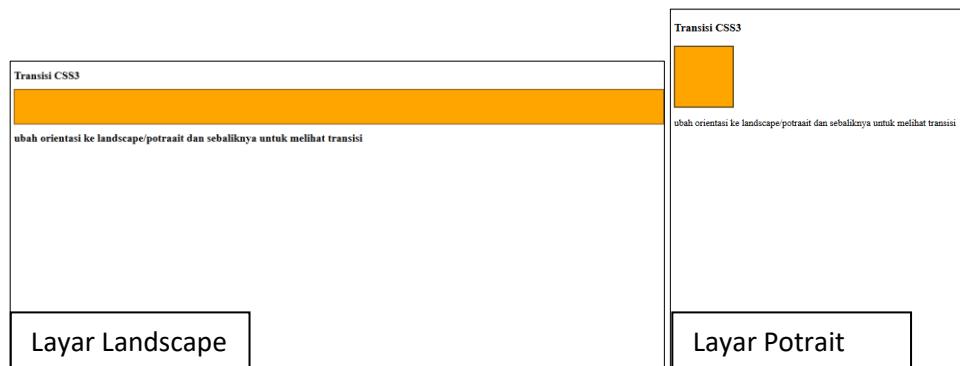
### Latihan 2a dan 2b:

Layar browser desktop

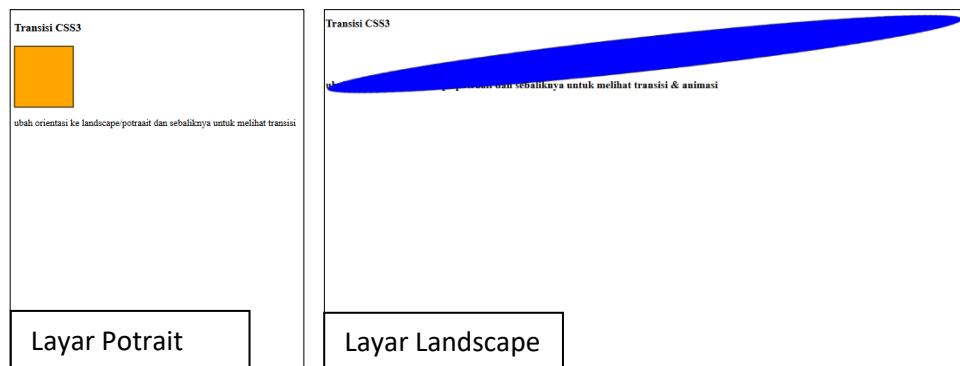




### Latihan 3a:



### Latihan 3b:



## F. LANGKAH KERJA

### Latihan 1a:

1. Masuk ke teks editor (VS Code)
2. Mengetikkan scrip HTML seperti pada gambar dibawah:

```
Latihan1.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Menggunakan media Query CSS3</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <style>
8          .smartphone {
9              display: none;
10         }
11
12         @media only screen and (max-width: 428px) {
13             body {
14                 background-color: yellow;
15             }
16
17             .smartphone {
18                 display: block;
19             }
20         }
21     </style>
22
23 </head>
24 <body>
25     <h3>Menggunakan Media Query CSS 3</h3>
26     <div class="smartphone">
27         <h2>Terdeteksi ukuran smartphone</h2>
28     </div>
29
30 </body>
31 </html>
```

- Mendefinisikan meta tag viewport dengan atribut content:width=device-width dan initial-scale=1.0 untuk memastikan lebar viewport menyesuaikan lebar device dan skala awal 1.0.
  - Menentukan style default dengan menyembunyikan class .smartphone menggunakan display: none; dan menampilkan teks utama “Menggunakan Media Query CSS 3”
  - Menerapkan media query yaitu layar smartphone dengan @media only screen and (max-width: 428px) untuk menentukan aturan CSS bagi layer yang berukuran maksimal 428px. Kemudian didalamnya mengubah background-color menjadi kuning. Dan menampilkan teks “Terdeteksi ukuran smartphone” dengan memblokir display utama.
3. Menyimpan file dengan nama Latihan 1, dan menjalankan di web browser. (tampilan dapat dilihat pada Output Program)

## Latihan 1b:

1. Masuk ke teks editor (VS Code)
2. Mengetikkan scrip HTML seperti pada gambar dibawah:

```
Latihan1b.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Menggunakan media query CSS</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <style>
8          .smartphone {
9              display: none;
10         }
11         .logo {
12             display: none;
13             width: 150px;
14             margin: 20px auto;
15         }
16         .text {
17             display: none;
18         }
19
20     @media only screen and (max-width: 428px) {
21         body {
22             background-color: #yellow;
23             text-align: center;
24         }
25
26         .smartphone {
27             display: block;
28         }
29     }
30
31     @media only screen and (min-width: 1025px){
32         body {
33             background-color: #azure;
34             text-align: center;
35         }
36     }
37
38         .logo {
39             display: block;
40             width: 50px;
41             max-width: 80%;
42             margin: 40px auto 10px auto;
43         }
44         .text {
45             display: block;
46             font-size: 30px;
47             color:#darkkhaki;
48             font-weight: bold;
49             margin-top: 10px;
50         }
51     </style>
52 </head>
53 <body>
54     <h1>Menggunakan Media Query CSS 3</h1>
55     <div class="smartphone">
56         <h2>terdeteksi ukuran smartphone</h2>
57     </div>
58
59     
60     <div class="text">
61         <p>ukuran dekstop lebih dari 1024px, Logo Muncul
62     </div>
63 </body>
64 </html>
```

- Mendefinisikan meta tag viewport dengan atribut content:width=device-width dan initial-scale=1.0 untuk memastikan lebar viewport menyesuaikan lebar device dan skala awal 1.0.
- Membuat class smartphone dengan aturan yang sama pada Latihan 1a
- Menetapkan media query untuk tampilan layar dengan lebar minimal 1025px atau lebih. Ketika aktif latar belakang berubah menjadi azure dan text-align menjadi center.

- Logo uns akan muncul melalui media <img class=logo> dengan properti max-width= 80% ditengah halaman dengan margin 40px, auto, dan 10px.
  - Elemen pada class .text akan muncul dengan ukuran 30px berwarna biru gelap.
3. menyimpan file dengan nama Latihan 1b, dan menjalankan di web browser. (tampilan dapat dilihat pada Output Program)

### Latihan 2a:

1. Masuk ke teks editor (VS Code)
2. Mengetikkan scrip HTML seperti pada gambar dibawah:

```
<!DOCTYPE html>
<html>
<head>
    <title>Desain Responsif</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style type="text/css">
        #wrapper{
            margin: auto;
            width: 100%;
        }
        #header{
            height: 150px;
            padding: 20px;
            border: 2px solid red;
        }
        #nav{
            margin: auto;
            padding: 20px;
            height: 200px;
            border: 2px solid red;
        }
        .box{
            float: left;
            width: 22%;
            margin: 0 2% 0 0;
            height: 120px;
            background: red;
            border: 2px solid red;
        }
        #footer{
```

```

        padding: 20px;
        height: 120px;
        border: 2px solid red;
    }
    @media only screen and (max-width: 1080px){
        #nav{
            height: 360px;
        }
        .box{
            width: 45%;
            margin: 0 4% 20px 0;
        }
    }
    @media only screen and (max-width: 640px){
        #nav{
            height: 640px;
        }
        .box{
            width: 100%;
            margin: 0 0 20px 0;
        }
    }

```

</style>

</head>

<body>

<div id="wrapper">

<div id="header">

<h2>Header</h2>

</div>

<div id="nav">

<h2>Services</h2>

<div class="box"> BOX 1 </div>

<div class="box"> BOX 2 </div>

<div class="box"> BOX 3 </div>

<div class="box"> BOX 4 </div>

</div>

<div id="footer">

<h2>Footer</h2>

</div>

</div>

</body>

</html>

- Mendefinisikan meta tag viewport dengan atribut content:width=device-width dan initial-scale=1.0 untuk memastikan lebar viewport menyesuaikan lebar device dan skala awal 1.0.

- Membuat aturan style default (tampilan desktop) dengan #wrapper yang diatur width=100% dan posisi margin: auto (tengah).
  - .Box (ada 4) yang diatur menggunakan properti float:left dan memiliki lebar 22%. Menampilkan 4 kotak/kolom yang berjajar secara horizontal dengan jarak margin kanan 2%. Dan tinggi navigasi 200px untuk menampung satu baris kotak.
  - Menetapkan media query untuk tampilan layar dengan lebar maximal 1080px (tablet). Lebar kotak ditingkatkan menjadi 45% sehingga dalam 1 baris nantinya hanya akan ada 2 kotak, dan 2 sisanya turun ke baris berikutnya. Margin kanan diubah menjadi 4% dan margin bawah 20px untuk jarak antar baris. Height pada nav juga disesuaikan menjadi 360px karena kotaknya menjadi 2 baris.
  - Menetapkan media query untuk tampilan layar dengan lebar maximal 640px(smartphone). Lebar kotak ditingkatkan menjadi 100% yang memaksa setiap kotak untuk mengisi 1 baris penuh sehingga susunan menjadi vertikal. Margin menjadi vertical dengan jarak bagian bawah 20px. Height pada nav juga disesuaikan menjadi 640px agar seluruh tampilan terlihat.
3. Menyimpan file dengan nama Latihan 2, dan menjalankan di web browser. (tampilan dapat dilihat pada Output Program)

## Latihan 2b:

1. Membuat file Latihan 2b
2. Menyalin scrip pada Latihan 2a, kemudian menyesuaikan beberapa bagian yang menggunakan float dan ubah menjadi flex seperti kode dibawah:

```
19  #nav{  
20      margin: auto;  
21      padding: 20px;  
22      border: 2px solid red;  
23      display: flex;  
24      flex-wrap: wrap;  
25      justify-content: space-between;  
26      align-items: center;  
27  }  
28  
29  .box{  
30      flex: 1 1 22%;  
31      margin: 10px;  
32      height: 120px;  
33      background: red;  
34      border: 2px solid red;  
35      box-sizing: border-box;  
36  }
```

- Pada container #nav aktifkan mode flexbox dengan mengubah display menjadi flex. Menghapus height karena flexbox akan menyesuaikan tinggi container secara otomatis. Dan menambahkan beberapa properti, seperti:
  - a. Menambahkan Flex-wrap:wrap agar item dapat turun ke baris baru saat ruang horizontal tidak cukup.
  - b. Menambahkan justify-content: space-between agar item diletakkan secara merata sepanjang baris dengan jarak diantara item.
- Pada container .box mengganti float: left dan width: 22% dengan flex: 1 1 22% yang merupakan ukuran dasar item (lebar awal box). Mengatur margin dengan ukuran 10px. Dan menambahkan:

Box-sizing: border-box; untuk memastikan padding dan border tidak menambah lebar elemen (cukup 22%).

```

44     @media only screen and (max-width: 1080px){
45         .box{
46             flex: 1 1 45%;
47         }
48     }
49
50     @media only screen and (max-width: 640px){
51         .box{
52             flex: 1 1 100%;
53         }
54     }

```

- Pada media query tablet (max 1080px) ubah flex menjadi 45% agar wrapping menjadi 2 kolom.
  - Pada media query smartphone (max 640px) ubah flex menjadi 100% untuk memaksa setiap kotak menempati satu baris secara penuh (stacking).
3. Menyimpan file dengan nama Latihan 2b, dan menjalankan di web browser. (tampilan dapat dilihat pada Output Program)

### Latihan 3a:

1. Masuk ke teks editor (VS Code)
2. Mengetikkan scrip HTML seperti pada gambar dibawah:

```

Latihan3.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Animasi Transisi CSS3</title>
5      <meta name=""viewport" content="width-device-width, initial-scale=0.1">
6
7      <style typ=""text/css">
8          .box{
9              width: 10%;
10             height: 100px;
11             background: #orange;
12             border: 2px solid #212121;
13
14             -webkit-transition: width 1s ease-in-out;
15             -o-transition: width 1s ease-in-out;
16             -moz-transition: width 1s ease-in-out;
17             transition: width 1s ease-in-out;
18         }
19
20         @media (orientation: landscape){
21             .box{
22                 width: 100%;
23             }
24         }
25     </style>
26 </head>
27 <body>
28     <h1>Transisi CSS3</h1>
29
30     <div class="box"></div>
31     <p>
32         <h1>ubah orientasi ke landscape/potrait dan sebaliknya untuk melihat transisi</h1>
33     </p>
34 </body>
35 </html>

```

- Mendefinisikan meta tag viewport dengan atribut content:width=device-width dan initial-scale=1.0 untuk

- memastikan lebar viewport menyesuaikan lebar device dan skala awal 1.0.
- Objek .box secara default (portrait) memiliki lebar 10% dan berwarna orange.
  - Mendefinisikan transition dengan prefix (-webkit-, -o-, -moz-) yang berisi width (yang dianimasikan adalah lebarnya), 1s (durasi transisinya adalah 1 detik), dan ease-in-out (objek bergerak lambat di awal dan lebih lambat lagi di akhir).
  - Media query orientasi untuk memicu transisi. Aturan didalamnya hanya akan aktif Ketika layar landscape. Dan lebar kotak atau box diubah menjadi 100%.
3. Menyimpan file dengan nama Latihan 3, dan menjalankan di web browser. (tampilan dapat dilihat pada Output Program)

### **Latihan 3b:**

1. Masuk ke teks editor (VS Code)
2. Mengetikkan scrip HTML seperti pada gambar dibawah:

```
transition:
    width 1s ease-in-out;
    background-color: 1s ease-in-out;
    border-radius: 1s ease-in-out;
    transform: 2s linear infinite;
}
```

- Menambahkan transisi 1 detik pada perubahan warna background agar dari oranye ke biru tidak tiba-tiba.
- Menambah transisi 1 detik juga pada transisi bentuk dari persegi Panjang menjadi oval melalui border-radius.
- Mengatur properti transform yang digunakan untuk rotasi untuk bertransisi selama 2 detik.

```
@keyframes rotate360 {
    from {
        transform: rotate(0deg);
    }
    to {
```

```
        transform: rotate(360deg);
    }
}
```

- Penambahan @keyframes untuk mendefinisikan Gerakan, rotate360 blueprint untuk mendefinisikan transform dari rotate(0deg) posisi awal ke rotate(360deg) satu putaran penuh.

```
@media (orientation: landscape){
    .box{
        width: 100%;
        background: blue;
        border-radius: 50%;
        animation: rotate360 2s linear infinite;
    }
}
```

- Mengubah warna background objek menjadi biru. Mengubah border radius menjadi 50% (oval). Dan memerintahkan bo untuk menjalankan animasi rotate360 selama 2 detik secara linear infinite.
3. Menyimpan file dengan nama Latihan 3b, dan menjalankan di web browser. (tampilan dapat dilihat pada Output Program).

## G. KESIMPULAN

Dari praktikum Desain Web ini dapat disimpulkan bahwa Responsive Web Design (RWD) berperan penting dalam menciptakan tampilan website yang adaptif terhadap berbagai ukuran layar, baik desktop, tablet, maupun smartphone. Dengan menerapkan meta tag viewport, media query, serta teknik Flexbox CSS, tampilan website menjadi lebih fleksibel, rapi, dan dapat menyesuaikan diri secara otomatis tanpa mengubah struktur utama HTML. Selain itu, penggunaan media query memungkinkan untuk menerapkan style yang berbeda sesuai resolusi layar, sementara flexbox membantu dalam pengaturan layout yang lebih fleksibel dan efisien dibandingkan dengan metode lama seperti float. Praktikum ini juga menunjukkan bagaimana transisi dan animasi CSS3 dapat mempercantik tampilan sekaligus meningkatkan interaktivitas halaman web.