



LEMBAR ASISTENSI  
PRAKTIKUM ALGORITMA DAN PEMROGRAMAN  
LABORATORIUM TEKNIK KOMPUTER  
JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG

Judul Praktikum                      HTML  
Praktikan (NPM)                    MUHAMMAD FITRA YUDHA    (2115061015)  
Asisten (NPM)                      Aldo Wijaya                      (2015061012)  
Kelompok                            PSTI A

No	Catatan	Tanggal	Paraf

Bandar Lampung, 4 november 2023

.....  
NPM.

## I. JUDUL PERCOBAAN

HTML

## II. TUJUAN PERCOBAAN

Adapun tujuan dari percobaan ini adalah sebagai berikut

1. Melakukan percobaan dasar HTML
2. Memahami dasar-dasar dari HTML

## III. TEORI DASAR

HTML5 memiliki struktur dasar yang sama dengan versi sebelumnya. Dokumen HTML5 dimulai dengan tag “<!DOCTYPE html>”, yang memberi tahu browser bahwa itu adalah dokumen HTML5. Selanjutnya, dokumen HTML5 terdiri dari elemen dasar seperti “<html> », « <kepala> » dan “<tubuh> ”. Elemen “<html>” » mengelilingi keseluruhan dokumen, “<head> » berisi informasi tentang dokumen, seperti judul dan parameter, dan “<content> » berisi konten yang akan ditampilkan pada halaman web.

Salah satu fitur utama HTML5 adalah penambahan tag dan elemen baru. Ini termasuk tag seperti “<header>“, yang digunakan untuk mengelompokkan elemen-elemen yang berisi judul atau header halaman. Ada juga tag “<nav>“, yang digunakan untuk mengelompokkan tautan navigasi. Tag “<article>“ berguna untuk mengelompokkan konten yang berdiri sendiri, seperti postingan blog. HTML5 juga mengenalkan tag “<video>“ dan “<audio>“, yang memungkinkan kita menyisipkan dan memutar video dan audio langsung di halaman web tanpa perlu plugin pihak ketiga seperti Flash. Selain itu, ada elemen ““yang memungkinkan pembuatan grafis dan gambar interaktif dengan menggunakan JavaScript. HTML5 juga

mendukung grafik vektor melalui tag “<svg>”. sangat berguna untuk menggambar gambar dan grafik responsif. Ada juga tag “<details>” dan “<summary>” yang memungkinkan kita membuat daftar detail yang dapat diperluas atau ditiadakan. Dalam hal pemrosesan formulir, HTML5 memperkenalkan tipe input baru seperti email, tanggal, dan rentang untuk mengatasi masalah validasi formulir dengan lebih baik. Atribut “contenteditable” memungkinkan kita membuat konten halaman web yang dapat diedit oleh pengguna. Atribut “draggable” memungkinkan elemen diseret dan dilepas. Aspek penting dari HTML5 adalah peningkatan semantik. Artinya HTML5 menawarkan banyak elemen yang memberikan makna lebih tepat pada konten halaman web. Hal ini membantu mesin pencari dan teknologi bantu memahami dan menafsirkan konten. Selain itu, HTML5 berfokus pada aksesibilitas web, sehingga pengguna penyandang disabilitas dapat mengakses situs web yang dibuat dengan HTML5 dengan lebih baik. HTML5 juga memungkinkan aplikasi web beroperasi secara offline dengan memanfaatkan penyimpanan lokal dan sesi.

Ini berarti user dapat mengakses sebagian besar konten web bahkan ketika mereka tidak terhubung ke internet. Selain itu, HTML5 menghadirkan API Geolocation, yang memungkinkan aplikasi web untuk mengakses lokasi user. Ini sangat berguna untuk layanan berbasis lokasi seperti peta interaktif dan aplikasi terkait lokasi. HTML5 adalah langkah besar dalam evolusi web. Dengan peningkatan dalam semantik, fitur multimedia, aksesibilitas, dan banyak lagi, HTML5 memungkinkan pengembang web untuk menciptakan pengalaman web yang lebih dinamis, interaktif, dan aksesibel. Dengan pemahaman dasar tentang HTML5, Anda dapat memulai perjalanan Anda untuk mengembangkan situs web yang modern dan responsif.

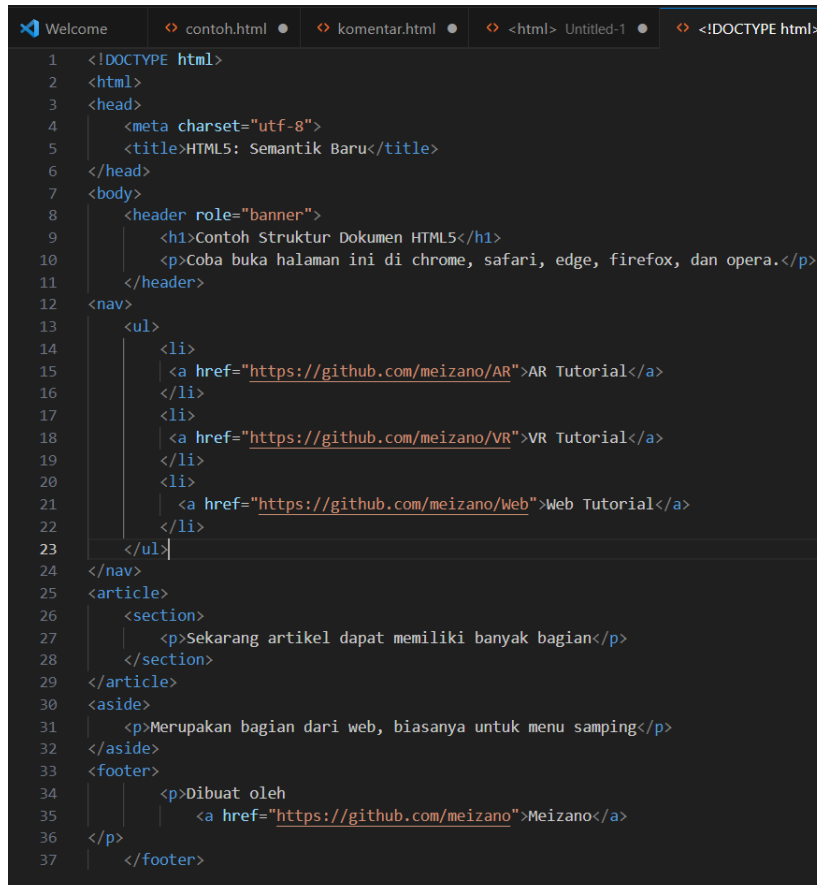
HTML5 memiliki WebSockets yang memungkinkan komunikasi real-time. Kita bisa membuat chat online atau game online yang responnya cepat. HTML5 juga mendukung grafik 3D menggunakan WebGL, ideal untuk game online dan visualisasi interaktif. Cache aplikasi membantu membuat situs web dapat diakses secara offline, dan Penyimpanan Lokal memungkinkan penyimpanan data sisi klien. HTML5 juga lebih mobile-friendly, dirancang untuk bekerja dengan baik di

perangkat seluler. Kita dapat menyuntikkan audio dan video langsung ke website tanpa memerlukan plugin seperti Flash dan HTML5 juga mendukung validasi form yang lebih baik. Terdapat berbagai fitur keamanan seperti Kebijakan Keamanan Konten (CSP) yang membantu melindungi situs web kami dari serangan berbahaya. Dengan pemahaman dasar tentang HTML5 dan fitur-fitur canggih yang ditawarkannya, Anda dapat membuat situs web yang lebih baik, lebih interaktif, dan lebih responsif. HTML5 adalah alat penting bagi pengembang web untuk menciptakan pengalaman web yang lebih baik di masa depan.

#### IV. PROSEDUR PERCOBAAN

Adapun prosedur untuk percobaan ini adalah sebagai berikut

##### 4.1 Percobaan 2-1 semantik HTML

A screenshot of a code editor with a dark theme. The editor has four tabs at the top: 'Welcome', 'contoh.html', 'komentar.html', and '<html> Untitled-1'. The active tab is '<html> Untitled-1'. The code is written in a light blue font on a dark background. It shows an HTML5 document structure with semantic elements. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>HTML5: Semantik Baru</title>
6 </head>
7 <body>
8   <header role="banner">
9     <h1>Contoh Struktur Dokumen HTML5</h1>
10    <p>Coba buka halaman ini di chrome, safari, edge, firefox, dan opera.</p>
11  </header>
12  <nav>
13    <ul>
14      <li>
15        <a href="https://github.com/meizano/AR">AR Tutorial</a>
16      </li>
17      <li>
18        <a href="https://github.com/meizano/VR">VR Tutorial</a>
19      </li>
20      <li>
21        <a href="https://github.com/meizano/Web">Web Tutorial</a>
22      </li>
23    </ul>
24  </nav>
25  <article>
26    <section>
27      <p>Sekarang artikel dapat memiliki banyak bagian</p>
28    </section>
29  </article>
30  <aside>
31    <p>Merupakan bagian dari web, biasanya untuk menu samping</p>
32  </aside>
33  <footer>
34    <p>Dibuat oleh
35      <a href="https://github.com/meizano">Meizano</a>
36    </p>
37  </footer>
```

Gambar Gambar 4.1 Percobaan 2-1 semantik HTML

## 4.2 Percobaan 2-2 Formulir 2.0

```
Welcome | contoh.html | komentar.html | <html> Untitled-1 | <!DOCTYPE html> Untitled-1
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>HTML5: Formulir 2.0</title>
6     <script>
7       function tampilkanHasil() {
8         url = document.forms["formulir"]["situs"].value;
9         document.forms["formulir"]["hasil"].value = url;
10      }
11    </script>
12  </head>
13  <body>
14    <form name="formulir">
15      Tanggal :
16      <input type="date" name="tanggal" required/>
17      <br/> Jam :
18      <input type="time" name="jam" />
19      <br/> Angka :
20      <input type="number" min="0" max="10" step "1" value="5" name="angka"
21      autofocus/>
22      <br/> email :
23      <input type="email" name="email" placeholder="email@example.com" />
24      <br/> URL :
25      <input type="url" name="situs" />
26      <br/>
27      <input type="submit" value="Hasil" onclick="tampilkanHasil();" />
28      <br/> <output name="hasil"></output>
29    </form>
30  </body>
31 </html>
```

Gambar 4.2 Percobaan 2-2 formulir 2.0

## 4.3 Percobaan 2-3 Web Storage

```
Welcome | sematik.html | formulir 2.0.html | <!DOCTYPE html> Untitled-1
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>HTML5: Web Storage</title>
6     <script>
7       if (localStorage.hits) {
8         localStorage.hits = Number(localStorage.hits) + 1;
9       } else {
10        localStorage.hits = 1;
11      }
12      document.write("Jumlah hit : " + localStorage.hits);
13    </script>
14  </head>
15  <body>
16    <script>
17      if (localStorage.hits) {
18        localStorage.hits = Number(localStorage.hits) + 1;
19      } else {
20        localStorage.hits = 1;
21      }
22      document.write("Jumlah hit : " + localStorage.hits);
23    </script>
24    <ol>
25      <li>Refresh halaman untuk meningkatkan jumlah hit.</li>
26      <li>Tutup window dan buka lagi untuk melihat hasilnya.</li>
27    </ol>
28  </body>
29 </html>
```

Gambar 4.3 Percobaan 2-3 garis dan komentar

#### 4.4 Percobaan 2-4 Canvas

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>HTML5: Canvas</title>
6   <script>
7     function animasi() {
8       setInterval(gambarCinta, 100);
9     }
10    function gambarCinta() {
11      // mendapatkan elemen canvas pada DOM
12      var canvas = document.getElementById('canvasku');
13      // Mengecek apakah canvas didukung
14      if (canvas.getContext) {
15        // menggunakan getContext untuk memakai canvas sebagai alat gambar
16        var ctx = canvas.getContext('2d');
17        // warna fill
18        ctx.fillStyle = 'rgba(0,253,206,0.4)';
19        // warna stroke
20        ctx.strokeStyle = 'rgba(0,153,55,0.4)';
21        ctx.save();
22        ctx.translate(200, 200);
23        var time = new Date();
24        ctx.rotate(((2 * Math.PI) / 6) * time.getSeconds() + ((2 * Math.PI) /
25        6000) * time.getMilliseconds());
26        ctx.translate(0, 28.5);
27        // Membuat simbol cinta
28        ctx.beginPath();
29        ctx.moveTo(75, 40);
30        ctx.bezierCurveTo(75, 37, 70, 25, 50, 25);
31        ctx.bezierCurveTo(20, 25, 20, 62.5, 20, 62.5);
32        ctx.bezierCurveTo(20, 80, 40, 102, 75, 120);
33        ctx.bezierCurveTo(110, 102, 130, 80, 130, 62.5);
34        ctx.bezierCurveTo(130, 62.5, 130, 25, 100, 25);
35        ctx.bezierCurveTo(85, 25, 75, 37, 75, 40);
36        ctx.fill();
37        ctx.restore();
38      }
39      16
40    }
41    } else {
42      alert('Membutuhkan browser yang mendukung canvas');
43    }
44  }
45 </script>
46 </head>
47 <body onload="animasi();">
48 <canvas id="canvasku" width="400" height="400"></canvas>
49 </body>
50 </html>
```

Gambar 4.4 Percobaan 2-4 canvas

#### 4.5 Percobaan 2-5 Audio dan Video

```
Welcome sematik.html formulir 2.0.html <!DOCTYPE html> Untitled-1
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML5: Audio & Video</title>
6 <script type="text/javascript">
7   function PlayVideo() {
8     var v = document.getElementsByTagName("video")[0];
9     v.play();
10  }
11 </script>
12 </head>
13 <body>
14 <audio controls autoplay>
15   <source src="asset/Indonesia-Raya.mp3" type="audio/mp3" />
16   <source src="asset/Indonesia-Raya.ogg" type="audio/ogg" />
17   <source src="asset/Indonesia-Raya.wav" type="audio/wav" /> browser tidak
18   mendukung elemen audio.
19 </audio>
20 <form>
21 <video width="300" height="200" src="asset/Indonesia-Raya.mp4">
22   browser tidak mendukung elemen video.
23 </video>
24 <br />
25 <input type="button" onclick="PlayVideo();" value="Play" />
26 </form>
27 </body>
28 </html>
```

Gambar 4.5 Percobaan 2-5 Audio dan Video

#### 4.6 Percobaan 2-6 Geolocation

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset = "utf-8">
5 <title>HTML5: Geolocation</title>
6 <script>
7   function showLocation(position) {
8     var latitude = position.coords.latitude;
9     var longitude = position.coords.longitude;
10    alert("Latitude : " + latitude + " Longitude: " + longitude);
11  }
12  function errorHandler(err) {
13    if(err.code == 1) {
14      alert("Error: Akses ditolak!");
15    } else if( err.code == 2) {
16      alert("Error: Posisi tidak tersedia!");
17    }
18  }
19  function getLocation() {
20    if(navigator.geolocation) {
21      // timeout at 60000 milliseconds (60 seconds)
22      var options = {timeout:60000};
23      navigator.geolocation.getCurrentPosition(showLocation,
24        errorHandler, options);
25    } else {
26      alert("Maaf, browser tidak mendukung geolocation!");
27    }
28  }
29 </script>
30 </head>
31 <body>
32 <form>
33   <input type = "button" onclick = "getLocation();" value = "Dapatkan
34   lokasi"/>
35 </form>
36 </body>
37 </html>
```

Gambar 4.6 Percobaan 2-6 geolocation



## 4.7 Percobaan 2-7 Microdata

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML5: Microdata</title>
6 </head>
7 <body>
8 <div itemscope>
9 <section itemscope itemtype="http://data-vocabulary.org/Person">
10 <h1 itemprop="name">Meizano Ardhi Muhammad</h1>
11 <a itemprop="url" href="http://meizano.github.io">Github</a>
12 </section>
13 </div>
14
15 <p>Uji dokumen html di <a href="https://search.google.com/structured-data/testing-
16 tool" target="_blank">Google Rich Snippets Testing Tool</a></a></p>
17
18 </body>
19 </html>
```

Gambar 4.7 Percobaan 2-7 Microdata

#### 4.6 Percobaan 2-8 Drag dan Drop

```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML5: Drag & Drop</title>
6  <style>
7  #boxA,
8  #boxB {
9  float: left;
10 padding: 10px;
11 margin: 10px;
12 -moz-user-select: none;
13 }
14 #boxA {
15 background-color: #rgb(255, 160, 51);
16 width: 75px;
17 height: 75px;
18 }
19 #boxB {
20 background-color: #rgb(102, 255, 127);
21 width: 150px;
22 height: 150px;
23 }
24 }
25 </style>
26 <script type="text/javascript">
27 function dragStart(ev) {
28 ev.dataTransfer.effectAllowed = 'move';
29 ev.dataTransfer.setData("Text", ev.target.getAttribute('id'));
30 ev.dataTransfer.setDragImage(ev.target, 0, 0);
31 return true;
32 }
33 function dragEnter(ev) {
34 event.preventDefault();
35 return true;
36 }
37 function dragOver(ev) {
38 return false;
39 }
40 function dragDrop(ev) {
41 var src = ev.dataTransfer.getData("Text");
42 ev.target.appendChild(document.getElementById(src));
43 ev.stopPropagation();
44 return false;
45 }
46 </script>
47 </head>
48 <body>
49 <center>
50 <div>Coba gerakkan kotak jingga ke kotak hijau.</div>
51 <div id="boxA" draggable="true" ondragstart="return dragStart(event)">
52 <p>Geser saya</p>
53 </div>
54 <div id="boxB" ondragenter="return dragEnter(event)" ondrop="return
55 dragDrop(event)" ondragover="return dragOver(event)">Brankas</div>
56 </center>
57 </body>
58 </html>
```

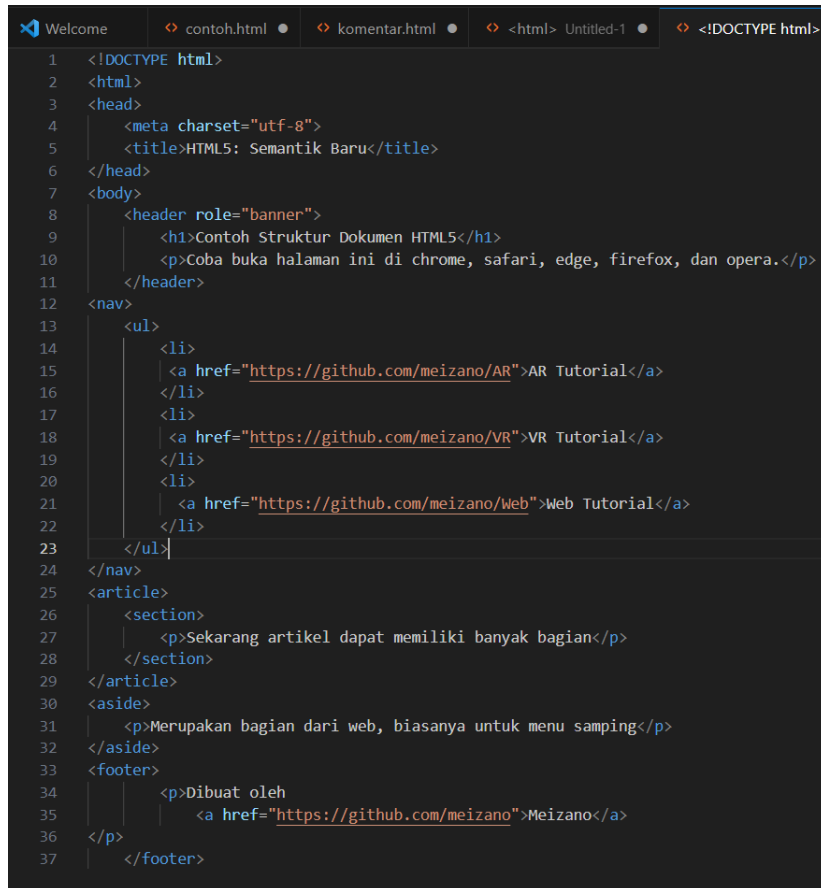
Gambar 4.8 Percobaan 2-8 Drag and Drop

## V. PEMBAHASAN

Adapun pembahasan untuk percobaan ini adalah sebagai berikut

### 5.1 Percobaan 2-1 Semantik HTML

#### 5.1.a Source Code Percobaan 2-1 Semantik HTML



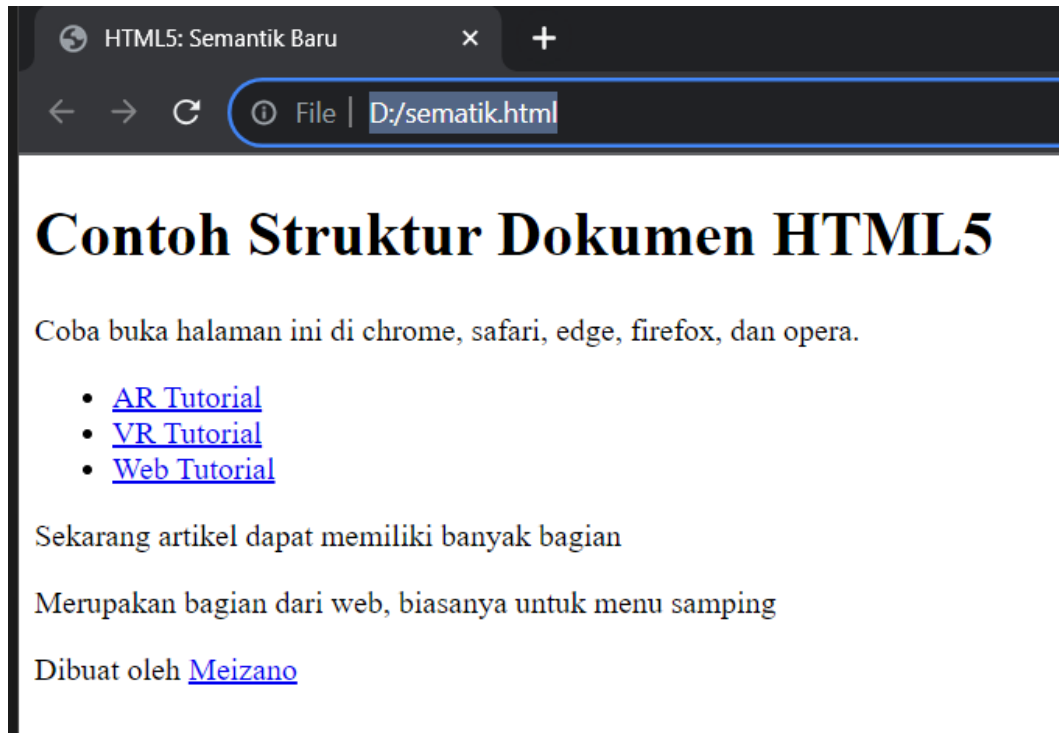
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>HTML5: Semantik Baru</title>
6 </head>
7 <body>
8   <header role="banner">
9     <h1>Contoh Struktur Dokumen HTML5</h1>
10    <p>Coba buka halaman ini di chrome, safari, edge, firefox, dan opera.</p>
11  </header>
12  <nav>
13    <ul>
14      <li>
15        <a href="https://github.com/meizano/AR">AR Tutorial</a>
16      </li>
17      <li>
18        <a href="https://github.com/meizano/VR">VR Tutorial</a>
19      </li>
20      <li>
21        <a href="https://github.com/meizano/Web">Web Tutorial</a>
22      </li>
23    </ul>
24  </nav>
25  <article>
26    <section>
27      <p>Sekarang artikel dapat memiliki banyak bagian</p>
28    </section>
29  </article>
30  <aside>
31    <p>Merupakan bagian dari web, biasanya untuk menu samping</p>
32  </aside>
33  <footer>
34    <p>Dibuat oleh
35      <a href="https://github.com/meizano">Meizano</a>
36    </p>
37  </footer>
```

Gambar 5.1.a Source Code Percobaan 2-1 semantik HTML

Berdasarkan gambar diatas, dapat dilihat bahwa Pada baris ke-1, terdapat “<!DOCTYPE html>” untuk mendeklarasikan jenis dokumen HTML5. Pada baris ke-2, terdapat “<html>”, yang merupakan elemen utama dalam dokumen HTML. Pada baris ke-3, terdapat “<head>”, berisi informasi tambahan tentang dokumen, seperti karakter set dan judul halaman. Pada baris ke-4, terdapat “<meta charset="utf-8">”, berfungsi untuk menentukan karakter set UTF-8 untuk karakter khusus dalam dokumen. Pada baris ke-5, terdapat “<title>HTML5 Semantik Baru</title>”, berfungsi untuk menentukan judul halaman yang akan ditampilkan. Pada baris ke-7, terdapat “<body>”, berfungsi sebagai wadah . Pada baris ke-8,

terdapat “<header role="banner">“, yang merupakan elemen header dengan atribut “role="banner"“, bagian atas halaman yang berisi elemen seperti judul halaman. Pada baris ke-9, terdapat “<h1>Contoh Struktur Dokumen HTML5</h1>“, berisi judul utama halaman yang terletak dalam header. Pada baris ke-10, terdapat “<p>Coba buka halaman ini di chrome, safari, edge, firefox, dan opera.</p>“, berfungsi untuk menampilkan sebuah paragraf pada header dan memberikan informasi tambahan. Pada baris ke-12, terdapat “<nav>“, yang berfungsi sebagai elemen navigasi, digunakan untuk menyimpan tautan menu. Pada baris ke-13, terdapat “<ul>“, merupakan daftar yang berisi tautan-tautan menu. Pada baris ke-14, terdapat “<li>“, merupakan elemen daftar dalam daftar tak-terurut yang berisi tautan ke tutorial .Pada baris ke-15, terdapat “<a href="https://github.com/meizano/AR">AR Tutorial</a>“, berfungsi sebagai tautan ke tutorial . Pada baris ke-25, terdapat “<article>“, yang merupakan elemen artikel yang digunakan untuk mengelompokkan konten yang independen. Pada baris ke-26, terdapat “<section>“, yang merupakan elemen bagian dalam artikel yang menunjukkan pemisahan konten dalam artikel. Pada baris ke-20, terdapat “<p>Sekarang artikel dapat memiliki banyak bagian</p>“, berfungsi untuk menampilkan paragraf dalam bagian tersebut. Pada baris ke-30, terdapat “<aside>“, merupakan elemen aside yang berisi konten terkait dengan halaman web, seperti menu samping. Pada baris ke-31, terdapat “<p>Merupakan bagian dari web, biasanya untuk menu samping</p>“, berfungsi untuk menampilkan paragraf dalam elemen aside. Pada baris ke-33, terdapat “<footer>“, yang merupakan elemen footer. Pada baris ke-34-35, terdapat “<p>Dibuat oleh <a href="https://github.com/meizano">Meizano</a></p>“, yang berfungsi untuk menampilkan tautan pembuat ke github.

### 5.1.b Output Percobaan 2-1 Semantik HTML



Gambar 5.1.b Output Percobaan 2-1 semantik HTML

Berdasarkan gambar di atas, Dapat dilihat bahwa output menampilkan halaman web. Hasil tersebut didapat dari source code pada baris ke-9 hingga ke-23. Baris ke-8 hingga ke-11 mendefinisikan elemen header yang berisi judul utama halaman dan paragraf. Baris ke-12 hingga ke-24 mengandung elemen navigasi dengan tautan-tautan menu ke tutorial. Baris ke-25 hingga ke-29 menggunakan elemen article dan section untuk mengelompokkan konten artikel ke dalam bagian-bagian yang terorganisir. Kemudian, pada baris ke-30 hingga ke-32, terdapat elemen aside yang berisi informasi terkait.

### 5.1.c Jawaban Soal Percobaan 2-1 Semantik HTML

#### 1. Jelaskan guna semantik header, nav, article, section, aside, dan footer!

- Header adalah bagian atas dari halaman web. Biasanya berisi judul situs web, serta elemen lain yang memberikan informasi penting tentang situs Kita.
- Nav digunakan untuk membuat menu navigasi. Kita menempatkan tautan pada situs Kita.
- Article untuk mengelompokkan konten. Setiap artikel memiliki judul dan berdiri sendiri tanpa memerlukan konteks tambahan.
- Section digunakan untuk mengelompokkan konten yang memiliki makna terkait. Ini membantu Kita mengorganisasi halaman web menjadi bagian-bagian yang lebih mudah dimengerti. Setiap bagian biasanya memiliki judul yang menjelaskan isinya.
- Aside digunakan untuk menambahkan konten tambahan yang tidak terlalu terkait dengan konten utama halaman web.
- Footer adalah bagian bawah dari halaman web atau bagian tertentu.

## 5.2 Percobaan 2-2 Formulir 2.0

### 5.2.a Source Code Percobaan 2-2 Formulir 2.0

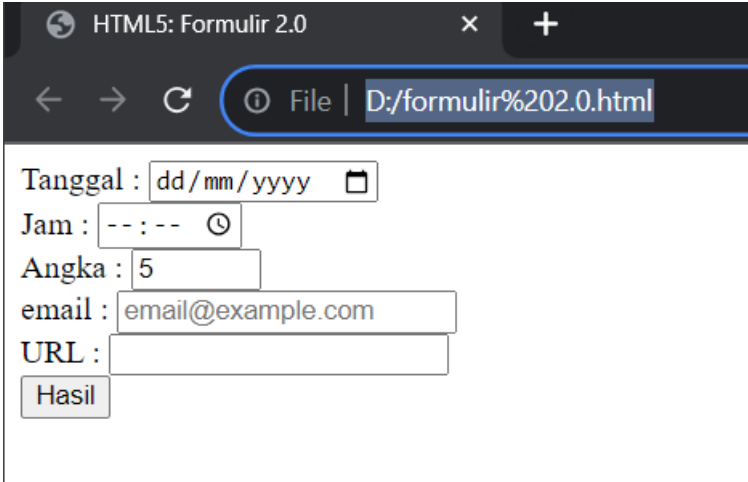
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>HTML5: Formulir 2.0</title>
6     <script>
7       function tampilkanHasil() {
8         url = document.forms["formulir"]["situs"].value;
9         document.forms["formulir"]["hasil"].value = url;
10      }
11    </script>
12  </head>
13  <body>
14    <form name="formulir">
15      Tanggal :
16      <input type="date" name="tanggal" required/>
17      <br/> Jam :
18      <input type="time" name="jam" />
19      <br/> Angka :
20      <input type="number" min="0" max="10" step="1" value="5" name="angka"
21      autofocus/>
22      <br/> email :
23      <input type="email" name="email" placeholder="email@example.com" />
24      <br/> URL :
25      <input type="url" name="situs" />
26      <br/>
27      <input type="submit" value="Hasil" onclick="tampilkanHasil();" />
28      <br/> <output name="hasil"></output>
29    </form>
30  </body>
31 </html>
```

Gambar 5.2.a Source Code Percobaan 2-2 Formulir 2.0

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris 1 berisi deklarasi tipe dokumen HTML. Baris 2-3 berisi elemen-elemen dasar dokumen HTML, seperti elemen “<html>“, “<head>“, dan pengaturan karakter set. Pada baris 11, terdapat elemen “<script>“, untuk mengolah data formulir. Baris 14-38 berisi elemen yang membentuk formulir dan memiliki atribut "name" dengan nilai "formulir" untuk menyisipkan script pada client. Baris 16, adalah Input tanggal dengan atribut "type=date" untuk memasukan tanggal, dan atribut "required" membuatnya wajib diisi dan tidak boleh kosong. Baris 18 Input waktu dengan atribut "type=time" untuk memasukkan jam. Baris 20 Input angka dengan atribut "type=number". Dengan nilai minimum adalah 0, nilai maksimum adalah 10, dan step adalah 1. Nilai awalnya adalah 5, dan pada baris 21 atribut "autofocus" membuat input agar fokus saat halaman dimuat. Baris 23 Input email dengan atribut "type=email" dan atribut "placeholder" yang memberikan contoh format email. Baris 25 Input URL dengan atribut "type=url" untuk memasukkan alamat situs pada

web. Baris 27 Tombol "Hasil" dengan atribut "type=submit". Ketika tombol ini ditekan, maka akan memanggil fungsi JavaScript "tampilkanHasil()". Baris 28 Output hasil yang akan menampilkan URL yang dimasukkan oleh user. Baris 30 - 31 , merupakan penutup pada dokumen HTML yang menutup elemen "html" dan "body".

#### 5.2.b Output Percobaan 2-2 Formulir 2.0



The screenshot shows a web browser window with the title "HTML5: Formulir 2.0". The address bar displays "File | D:/formulir%202.0.html". The form contains the following elements:

- Tanggal :  (with a calendar icon)
- Jam :  (with a clock icon)
- Angka :
- email :
- URL :
- Hasil (submit button)

Gambar 5.2.b Output Percobaan 2-2 Formulir 2.0

Berdasarkan gambar di atas, Dapat dilihat bahwa output menampilkan formulir dengan berbagai jenis input, seperti tanggal, jam , angka, alamat email, dan URL. Setelah mengisi formulir, user dapat mengklik tombol "Hasil". Ketika tombol "Hasil" ditekan, fungsi JavaScript “tampilkanHasil()”dijalankan. Fungsi ini akan mengambil nilai yang dimasukkan oleh user ke dalam input "URL". Setelah mengambil nilai URL tersebut, maka hasilnya akan ditampilkan di bawah tombol "Hasil" dalam elemen output.



### 5.2.c Jawaban Soal Percobaan 2-2 Formulir 2.0

#### 1. Jelaskan tipe input date, time, number, email, dan url!

- Tipe "date", memungkinkan user memilih tanggal dari kalender tanpa harus mengetikkannya.
- Tipe "time", Digunakan untuk memasukkan waktu, seperti jam, menit, dan detik.
- Tipe "number", digunakan untuk memasukkan angka numerik, seperti usia.
- Tipe "email", digunakan untuk mengumpulkan alamat email.
- Tipe "url", digunakan untuk memasukkan URL, seperti alamat situs web. Browser akan memeriksa apakah URL yang dimasukkan benar.

#### 2. Untuk apa step pada tipe input number?

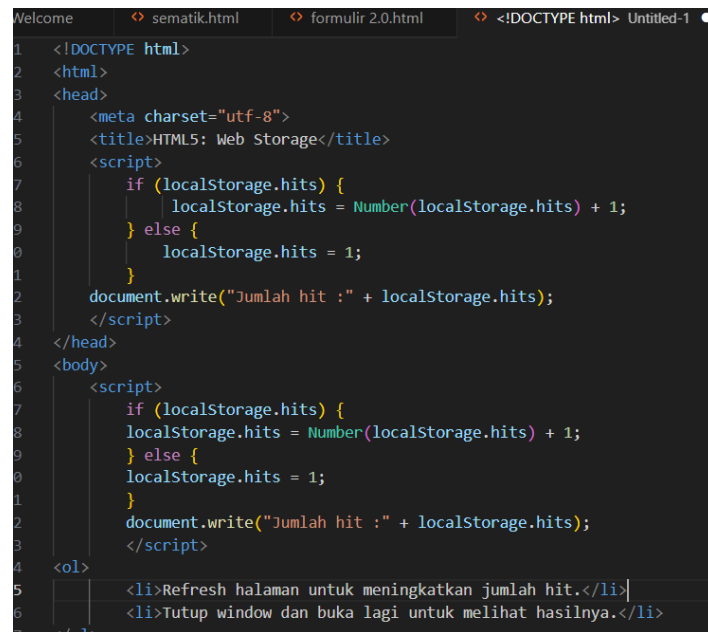
- Atribut step pada tipe input number sebagai pengatur perubahan nilai saat user menekan tombol naik atau turun. Contoh, jika "step" diatur ke "5," maka user hanya dapat mengubah nilai dalam kelipatan 5.

#### 3. Untuk apa autofocus dan required?

- Autofocus digunakan untuk memberikan fokus otomatis ke elemen input ketika halaman dimuat dan mempermudah user untuk langsung memasukkan data.
- Required digunakan agar input menjadi wajib diisi sebelum formulir dikirimkan. Jika inputan kosong, browser mencegah pengiriman formulir dan memberikan pesan kesalahan.

### 5.3 Percobaan 2-3 Web Storage

#### 5.3.a Source Code Percobaan 2-3 Web Storage

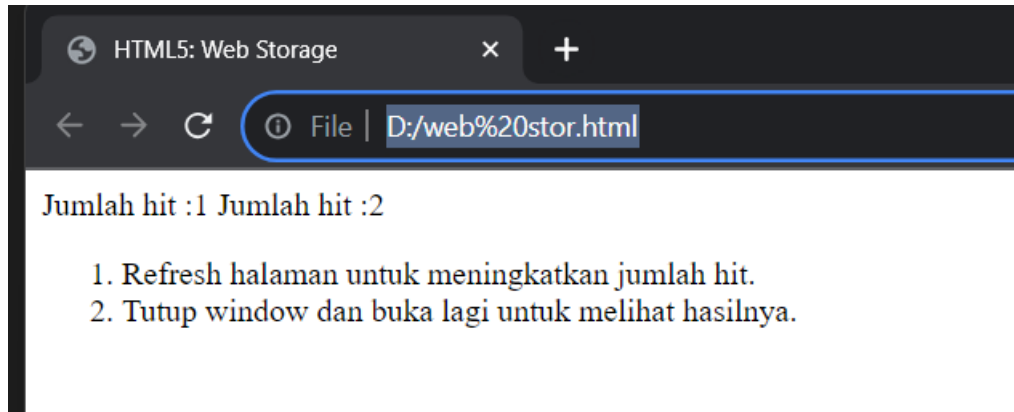


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>HTML5: Web Storage</title>
6   <script>
7     if (localStorage.hits) {
8       localStorage.hits = Number(localStorage.hits) + 1;
9     } else {
10      localStorage.hits = 1;
11    }
12    document.write("Jumlah hit : " + localStorage.hits);
13  </script>
14 </head>
15 <body>
16   <script>
17     if (localStorage.hits) {
18       localStorage.hits = Number(localStorage.hits) + 1;
19     } else {
20       localStorage.hits = 1;
21     }
22     document.write("Jumlah hit : " + localStorage.hits);
23   </script>
24   <ol>
25     <li>Refresh halaman untuk meningkatkan jumlah hit.</li>
26     <li>Tutup window dan buka lagi untuk melihat hasilnya.</li>
27   </ol>
```

Gambar 5.3.a Source Code Percobaan 2-3 Web Storage

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris 1-2 , merupakan deklarasi dokumen HTML. Baris 3-5, pada bagian “<html>” mencakup deklarasi dokumen dengan “<head>” sebagai kepala dokumen. Baris 5-6. pada “<head>”, menentukan judul halaman web dengan “<title>”, yang akan ditampilkan di browser. Baris 6-16, terdapat dua blok skrip JavaScript. Baris 7-12 merupakan blok JavaScript pertama digunakan untuk menghitung jumlah membuka halaman web. Jika Local Storage memiliki nilai "hits," maka nilai akan ditambahkan 1, dan kemudian disimpan kembali di Local Storage. Jika "hits" belum ada, maka akan diinisialisasi dengan nilai 1. Baris 12, terdapat code “document.write()” untuk menampilkan jumlah hit yang telah dihitung. Hasil ini akan muncul pada laman web sebagai "Jumlah hit " diikuti oleh jumlah hit yang diambil dari Local Storage. Baris 17-20 digunakan untuk menghitung jumlah hit dan menampilkannya dalam elemen HTML. Baris 24 adalah elemen “<ol>” yang berisi panduan tentang cara meningkatkan jumlah hit pada halaman web.

### 5.3.b Output Percobaan 2-3 Web Storage



Gambar 5.3.b Output Percobaan 2-3 Web Storage

Berdasarkan gambar di atas, Dapat dilihat bahwa output yang dihasilkan berupa teks "Jumlah hit 1" pada halaman, yang berarti kunjungan pertama ke laman tersebut. Setiap user merefresh halaman, jumlah hit akan bertambah 1. Contoh, ketika user merefresh halaman, user akan melihat teks "Jumlah hit 4" di halaman. Jika user menutup browser dan membuka halaman web kembali, jumlah hit akan tetap disimpan dan ditingkatkan dari jumlah terakhir saat jendela ditutup

### 5.3.c Jawaban Soal Percobaan 2-3 Web Storage

1. Jelaskan kegunaan obyek localStorage!

localStorage digunakan untuk menyimpan data di komputer user ketika mereka mengunjungi sebuah web. Data dalam "localStorage" tetap ada bahkan setelah kita menutup dan membuka kembali halaman web.

2. Jelaskan kegunaan document.write!

document.write digunakan untuk menambahkan menambahkan teks ke halaman.

3. Mengapa refresh menambah jumlah hit?

saat merefresh halaman web, browser akan meminta kembali semua elemen halaman dari server. Setiap permintaan ini dari server disebut "hit." Jadi, setiap merefresh, penyimpanan server akan mencatat permintaan baru, dan akan meningkatkan jumlah hit.

4. Apa yang berbeda setelah window ditutup?

Setelah Kita menutup window, semua data dan tindakan akan dihapus, data sesi hilang, dan notice penutupan jendela akan muncul. Jadi, semua yang pada window tidak akan berlaku lagi setelah ditutup.

## 5.4 Percobaan 2-4 Canvas

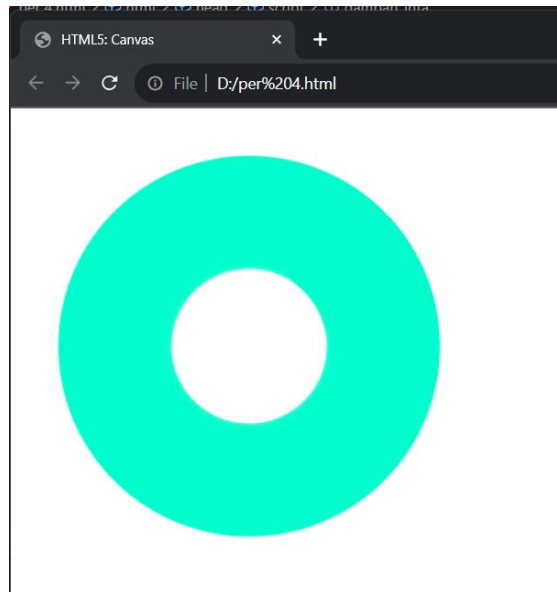
### 5.4.a Source Code Percobaan 2-4 Canvas

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>HTML5: Canvas</title>
6      <script>
7          function animasi() {
8              setInterval(gambarCinta, 100);
9          }
10         function gambarCinta() {
11             // mendapatkan elemen canvas pada DOM
12             var canvas = document.getElementById('canskaku');
13             // Mengecek apakah canvas didukung
14             if (canvas.getContext) {
15                 // menggunakan getContext untuk memakai canvas sebagai alat gambar
16                 var ctx = canvas.getContext('2d');
17                 // warna fill
18                 ctx.fillStyle = 'rgba(0,253,206,0.4)';
19                 // warna stroke
20                 ctx.strokeStyle = 'rgba(0,153,55,0.4)';
21                 ctx.save();
22                 ctx.translate(200, 200);
23                 var time = new Date();
24                 ctx.rotate(((2 * Math.PI) / 6) * time.getSeconds() + ((2 * Math.PI) /
25                     6000) * time.getMilliseconds());
26                 ctx.translate(0, 28.5);
27                 // Membuat simbol cinta
28                 ctx.beginPath();
29                 ctx.moveTo(75, 40);
30                 ctx.bezierCurveTo(75, 37, 70, 25, 50, 25);
31                 ctx.bezierCurveTo(20, 25, 20, 62.5, 20, 62.5);
32                 ctx.bezierCurveTo(20, 80, 40, 102, 75, 120);
33                 ctx.bezierCurveTo(110, 102, 130, 80, 130, 62.5);
34                 ctx.bezierCurveTo(130, 62.5, 130, 25, 100, 25);
35                 ctx.bezierCurveTo(85, 25, 75, 37, 75, 40);
36                 ctx.fill();
37                 ctx.restore();
38
39         16
40
41     } else {
42         alert('Membutuhkan browser yang mendukung canvas');
43     }
44 }
45 </script>
46 </head>
47 <body onload="animasi();">
48 <canvas id="canskaku" width="400" height="400"></canvas>
49 </body>
50 </html>
```

Gambar 5.4.a Source Code Percobaan 2-4 Canvas

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris pertama “<!DOCTYPE html>” mendeklarasi dokumen HTML5. Baris kedua “<html>” merupakan elemen awal dari HTML. Baris 3 terdapat “<html>” yang mencakup deklarasi karakter set dokumen dengan “<head>” sebagai elemen kepala dokumen. Baris 4 pada “<title>”, menentukan judul halaman web yang ditampilkan di tab browser . Baris 7-38 ada Fungsi “animasi()” ,baris 8 akan memanggil fungsi “gambarCinta()” setiap 100 milidetik menggunakan “setInterval()”. Fungsi “gambarCinta()” pada baris 10-38 menggambar animasi simbol cinta pada elemen “<canvas>”. fungsi ini akan mengakses elemen “<canvas>” dengan id 'cavasku' di baris 12 dan memeriksa apakah si browser mendukung “<canvas>” pada baris 14-19 . Jika “<canvas>” didukung, maka broser menggambar simbol cinta dengan fungsi “ctx.beginPath()” dan “ctx.bezierCurveTo()” untuk mengendalikan jalur simbol cinta. Kemudian, simbol cinta di isi dengan warna. Jika browser tidak mendukung “<canvas>”, user akan menerima pesan peringatan pada baris 41-42. Baris 47 Ini merupakan elemen “<body>” yang akan mengaktifkan fungsi “animasi()” saat halaman web dimuat menggunakan atribut “onload”. Baris 48 merupakan elemen “<canvas>” dengan id 'cavasku' yang digunakan untuk menggambar animasi simbol cinta. Lebar dan tinggi canvas akan ditentukan oleh atribut “width” dan “height”.

#### 5.4.b Output Percobaan 2-4 Canvas



Gambar 5.4.b Output Percobaan 2-4 Canvas

Berdasarkan gambar di atas, Dapat dilihat bahwa output yang dihasilkan menampilkan animasi simbol cinta pada elemen “<canvas>”. Saat web dimuat, animasi akan dimulai dan user akan melihat simbol cinta yang berputar membentuk lingkaran di dalam elemen “<canvas>”. Simbol cinta ini akan terus berputar setiap 100 milidetik karena fungsi “animasi()” memanggil “gambarCinta()” dengan interval tersebut. Jika browser user tidak mendukung elemen “<canvas>”, user akan diberi pesan peringatan yang muncul sebagai “alert” dengan teks "Membutuhkan browser yang mendukung canvas."

#### 5.4.c Jawaban Soal Percobaan 2-4 Canvas

##### 1. Kenapa DOM harus diakses?

- Untuk mendapatkan akses ke elemen canvas , codingan perlu mengakses elemen “<canvas>“ di dalam dokumen HTML kemudian menggambar di atasnya. Dengan menggunakan “document.getElementById('cankasku')”, kode mendapat referensi ke elemen canvas dengan ID "cankasku" agar dapat mengakses elemen tersebut.
- Untuk memeriksa dukungan canvas, Sebelum menggambar, codingan perlu memastikan bahwa browser yang digunakan mendukung elemen canvas.

##### 2. Jelaskan bagian kode membuat simbol cinta!

Dalam kode di atas, bagian yang membuat simbol cinta adalah :

- “ctx.beginPath();” digunakan untuk memulai penciptaan bagian baru di elemen canvas. Semua operasi akan dilakukan di dalam bagian ini.
- “ctx.moveTo(75, 40);” Perintah ini memindahkan pensil gambar ke koordinat (75, 40) pada elemen canvas. Dan sebagai titik awal dari gambar simbol cinta.
- “ctx.bezierCurveTo(...)” digunakan untuk membuat lengkungan yang akan membentuk simbol cinta ini merupakan kurva yang dapat diatur dengan titik kendali,
- “ctx.fill();”digunakan untuk mengisi area yang dipilih dengan warna yang di inginkan menggunakan “ctx.fillStyle”. warna yang dipilih pada code adalah 'rgba(0,253,206,0.4)'.



## 5.5 Percobaan 2-5 Audio dan Video

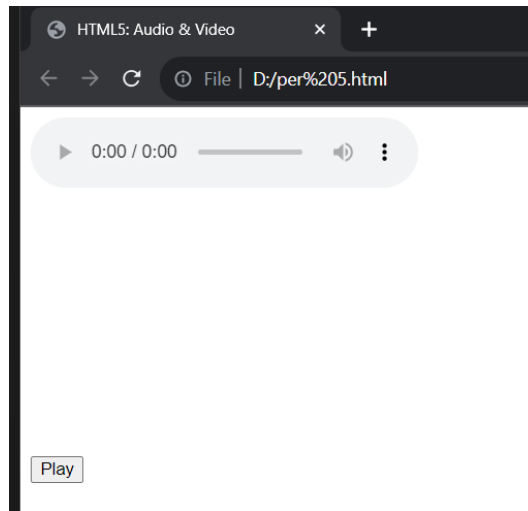
### 5.5.a Source Code Percobaan 2-5 Audio dan Video

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML5: Audio & Video</title>
6  <script type="text/javascript">
7  function PlayVideo() {
8  var v = document.getElementsByTagName("video")[0];
9  v.play();
10 }
11 </script>
12 </head>
13 <body>
14 <audio controls autoplay>
15 <source src="asset/Indonesia-Raya.mp3" type="audio/mp3" />
16 <source src="asset/Indonesia-Raya.ogg" type="audio/ogg" />
17 <source src="asset/Indonesia-Raya.wav" type="audio/wav" /> browser tidak
18 mendukung elemen audio.
19 </audio>
20 <form>
21 <video width="300" height="200" src="asset/Indonesia-Raya.mp4">
22 browser tidak mendukung elemen video.
23 </video>
24 <br />
25 <input type="button" onclick="PlayVideo();" value="Play" />
26 </form>
27 </body>
28 </html>
```

Gambar 5.5.a Source Code Percobaan 2-5 Audio dan Video

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris ke-1 terdapat “<!DOCTYPE html>” yang mendeklarasikan dokumen HTML5. Baris ke-2 untuk membuka elemen html . pada baris ke-3 “<html>”, digunakan untuk sebagai struktur dokumen dan dimulai dari elemen “<head>”. Di baris ke-5, mengatur judul halaman yang akan ditampilkan pada browser. Baris ke-6 hingga ke-11 merupakan blok JavaScript yang dimulai dari “<script>”. Fungsi “PlayVideo()” didefinisikan pada baris ke-7 untuk memutar video. Pada baris ke-13 hingga ke-27, elemen “<body>”, mendefinisikan elemen “<audio>” untuk memutar audio. Kontrol audio ditampilkan dengan atribut “controls”, audio akan diputar otomatis dengan atribut “autoplay”. Baris ke-20 hingga ke-26, memiliki elemen “<form>” yang terdapat elemen “<video>”. Lebar dan tinggi video akan ditentukan oleh atribut “width” dan “height”. Video akan ditampilkan dengan atribut “src”. Pada baris ke-25, terdapat tombol "Play" ketika user menekan, akan memanggil fungsi “PlayVideo()” untuk memutar video.

### 5.5.b Output Percobaan 2-5 Audio dan Video



Gambar 5.5.b Output Percobaan 2-5 Audio dan Video

Berdasarkan gambar di atas, Dapat dilihat bahwa output yang dihasilkan adalah sebuah audio dan video, Jika browser mendukung elemen audio dan user memilih untuk memutar audio, maka audio lagu "Indonesia-Raya" akan diputar dihasilkan dari Bagian audio “<audio>” dengan tiga source audio yang berbeda “<source>” serta akan memutar otomatis “autoplay”. Jika browser tidak mendukung elemen audio, pesan "browser tidak mendukung elemen audio" akan ditampilkan. Jika user mengklik tombol "Play" di bawah pemutar video dan browser mendukung pemutaran video, maka video "Indonesia-Raya" akan diputar. User dapat mengontrol pemutaran video dengan tombol pemutaran dengan tombol "Play" yang memanggil fungsi JavaScript “PlayVideo()” untuk memulai pemutaran videoyang terdapat pada baris 15.

### 5.5.c Jawaban Soal Percobaan 2-5 Audio dan Video

1. Jelaskan mengenai kegunaan atribut controls dan autoplay!

- Atribut control digunakan untuk menambahkan tombol control . pada code diatas yaitu tombol play ke pemutar audio atau video
- Atribut autoplay membuat audio atau video akan diputar secara otomatis ketika halaman dimuat.

2. Kenapa source pada audio ada banyak?

Atribut source digunakan untuk menyediakan beberapa file audio dengan format yang berbeda. karena beberapa browser mendukung format audio yang berbeda. Dengan menyediakan beberapa sumber audio, dapat membuat user mendengarkan audio tanpa masalah.

3. Apa pengaruh dari width dan height pada video?

Atribut “width” dan “height” mengatur ukuran visual dari area pemutaran video. Dengan mengaturnya kita dapat mengontrol besar video yang akan ditampilkan. Misalnya, kita atur “width=“300”“ dan “height=“200”“, maka lebar video 300 piksel dan tinggi 200 piksel ketika ditampilkan pada web.

## 5.6 Percobaan 2-6 Geolocation

### 5.6.a Source Code Percobaan 2-6 Geolocation

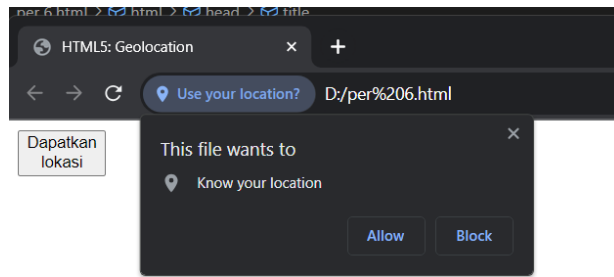
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset = "utf-8">
5  <title>HTML5: Geolocation</title>
6  <script>
7  function showLocation(position) {
8  var latitude = position.coords.latitude;
9  var longitude = position.coords.longitude;
10 alert("Latitude : " + latitude + " Longitude: " + longitude);
11 }
12 function errorHandler(err) {
13 if(err.code == 1) {
14 alert("Error: Akses ditolak!");
15 } else if( err.code == 2) {
16 alert("Error: Posisi tidak tersedia!");
17 }
18 }
19 function getLocation() {
20 if(navigator.geolocation) {
21 // timeout at 60000 milliseconds (60 seconds)
22 var options = {timeout:60000};
23 navigator.geolocation.getCurrentPosition(showLocation,
24 errorHandler, options);
25 } else {
26 alert("Maaf, browser tidak mendukung geolocation!");
27 }
28 }
29 </script>
30 </head>
31 <body>
32 <form>
33 <input type = "button" onclick = "getLocation();" value = "Dapatkan
34 lokasi"/>
35 </form>
36 </body>
37 </html>
```

Gambar 5.6.a Source Code Percobaan 2-6 Geolocation

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris ke-1 hingga ke-3, deklarasi dokumen HTML (“<!DOCTYPE html>”). Baris ke-3 merupakan elemen “<head>” sebagai kepala html. Di baris ke-5, kita mengatur judul halaman yang akan ditampilkan di tab browser. Baris ke-4 “charset =utf-8” memanggil karakter sesuai standar utf-8 . Baris ke-8 hingga ke-29 terdapat “<script>”. Di sini, juga mendefinisikan fungsi “showLocation(position)”, digunakan untuk menampilkan lokasi user. Itu akan menampilkan latitude dan longitude user. “errorHandler(err)” digunakan Ketika terjadi kesalahan saat mencoba mendapatkan lokasi user. akan menampilkan pesan kesalahan/ alert berdasarkan kode kesalahan. Pada baris-19 terdapat ”getLocation()” digunakan untuk mendapatkan lokasi user dengan menggunakan API geolocation yang tersedia dibrowser. Jika geolocation

didukung, maka “`getCurrentPosition`” akan digunakan untuk mendapatkan lokasi, dan jika tidak didukung, pesan kesalahan akan ditampilkan. Pada baris ke-33 terdapat elemen “`button`”, digunakan untuk memicu fungsi “`getLocation()`” saat diklik.

#### 5.6.b Output Percobaan 2-6 Geolocation



Gambar 5.6.b Output Percobaan 2-6 Geolocation

Berdasarkan gambar di atas, Dapat dilihat bahwa output yang dihasilkan adalah ketika user mengklik tombol "Dapatkan lokasi," fungsi “`getLocation()`” akan dipanggil berdasarkan source code baris 19. Jika browser mendukung geolocation, maka akan mencari lokasi user. Jika berhasil mendapatkan lokasi, maka fungsi “`showLocation(position)`” akan dipanggil pada baris 7. Output akan menampilkan latitude dan longitude user. Jika terjadi kesalahan dalam mendapatkan lokasi user, maka fungsi “`errorHandler(err)`” akan dipanggil berdasarkan baris 12. Jika kode kesalahan adalah 1 (`err.code == 1`), maka akan muncul pesan "Error Akses ditolak!". Ini terjadi ketika user menolak permintaan akses lokasi. Jika lokasi tidak ditemukan maka pesan "Error Posisi tidak tersedia!" akan ditampilkan.

### 5.6.c Jawaban Soal Percobaan 2-6 Geolocation

1. Apakah latitude dan longitude?

Latitude dan Longitude merupakan nilai yang digunakan untuk menunjukkan di mana lokasi kita berada di Bumi. Latitude (garis lintang) mengukur sejauh mana kita berada di utara atau selatan, sementara Longitude (garis bujur) mengukur sejauh mana kita berada di timur atau barat meridian. Kode ini mengambil informasi dari perangkat user dan menampilkannya dalam pesan.

2. Untuk apa `getCurrentPosition`?

`getCurrentPosition` digunakan untuk meminta lokasi geografis kita saat ini. Saat kita mengeklik tombol "Dapatkan lokasi,".

3. Apa yang terjadi jika kode dijalankan di perangkat bergerak seperti smartphone?

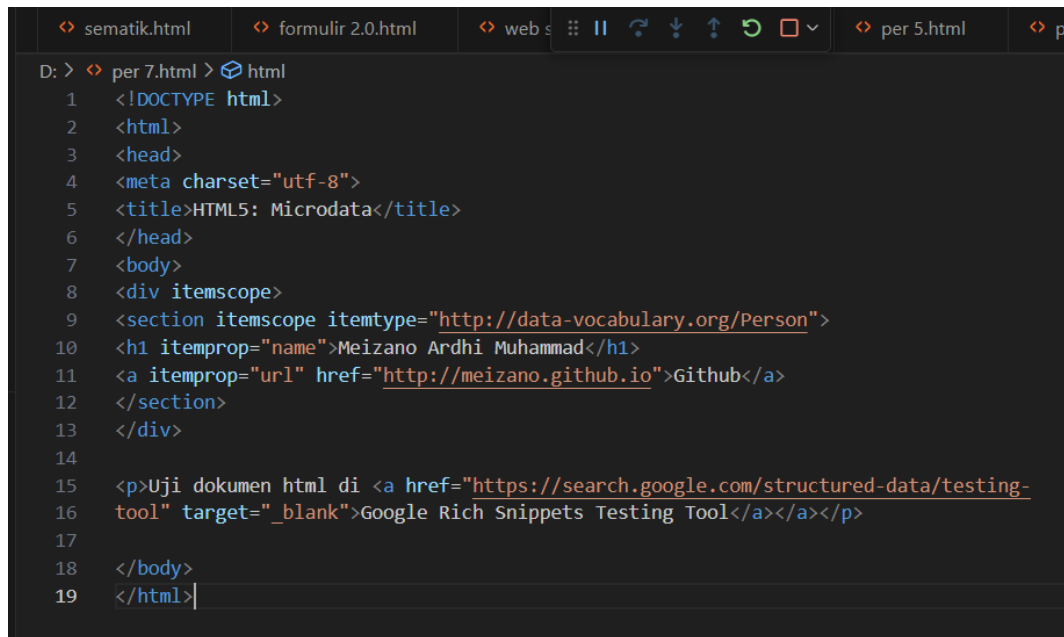
Jika dijalankan pada smartphone, maka saat mengeklik tombol "Dapatkan lokasi," browser akan meminta izin untuk menggunakan GPS atau lokasi terkini perangkat.

4. Apa yang terjadi jika kode dijalankan di desktop?

Jika dijalankan di desktop, kita masih dapat mengeklik tombol "Dapatkan lokasi," tetapi biasanya, perangkat desktop tidak memiliki GPS. Dalam hal ini, kode akan mencoba mendapatkan lokasi perkiraan dengan menggunakan informasi jaringan atau alamat IP komputer kita.

## 5.7 Percobaan 2-7 Microdata

### 5.7.a Source Code Percobaan 2-7 Microdata

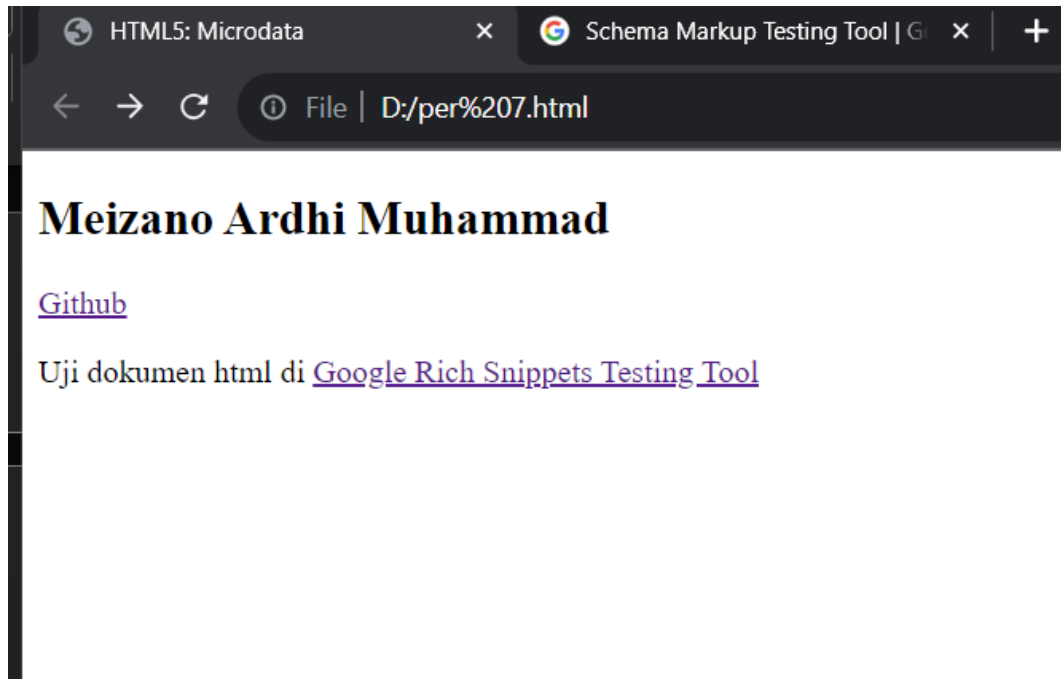
The image shows a code editor window with a dark background. At the top, there are several tabs: 'sematik.html', 'formulir 2.0.html', 'web s...', 'per 5.html', and 'p...'. The active tab is 'per 7.html'. The code is written in HTML and is as follows:

```
D: > > per 7.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML5: Microdata</title>
6  </head>
7  <body>
8  <div itemscope>
9  <section itemscope itemtype="http://data-vocabulary.org/Person">
10 <h1 itemprop="name">Meizano Ardhi Muhammad</h1>
11 <a itemprop="url" href="http://meizano.github.io">Github</a>
12 </section>
13 </div>
14
15 <p>Uji dokumen html di <a href="https://search.google.com/structured-data/testing-
16 tool" target="_blank">Google Rich Snippets Testing Tool</a></p>
17
18 </body>
19 </html>
```

Gambar 5.7.a Source Code Percobaan 2-7 Microdata

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris pertama, “<!DOCTYPE html>“, berfungsi untuk mendeklarasikan jenis dokumen HTML yang akan digunakan. Baris ke-3, “<head>“, menandakan setruktur dokumen bagian kepala dokumen kepala dokumen HTML. “<meta charset="utf-8">“ digunakan untuk menentukan karakter set dokumen sebagai UTF-8 sebagai standarisasi. “<title>“ digunakan untuk menentukan judul html di browser. Baris ke-7, “<body>“, sebagai awal dari isi dokumen HTML yang akan ditampilkan browser. Baris ke-8 hingga ke-16 mendefinisikan struktur halaman. elemen HTML, seperti “<div>“, “<section>“, “<h1>“, “<a>“, yang memiliki atribut “itemscope”, “itemtype”, dan “itemprop”. digunakan untuk tautan ke profil GitHub.

### 5.7.b Output Percobaan 2-7 Microdata



Gambar 5.7.b Output Percobaan 2-7 Microdata

Berdasarkan gambar di atas, Dapat dilihat bahwa output yang dihasilkan adalah halaman web yang menampilkan informasi "Meizano Ardhi Muhammad" dan tautan ke profil GitHub-nya. Kode seperti "itemscope", "itemtype", dan "itemprop" akan mendefinisikan "Meizano Ardhi Muhammad" dan tautan ke profil GitHubnya. dan berisi tautan ke "Google Rich Snippets Testing Tool." Tautan ini memungkinkan user menguji penggunaan microdata dalam dokumen HTML ini.



### 5.7.c Jawaban Soal Percobaan 2-7 Microdata

#### 1. Jelaskan tentang itemscope, itemtype, dan itemprop!

- itemscope adalah atribut yang digunakan dalam HTML untuk menandai elemen sebagai kontainer untuk data terstruktur, elemen yang memiliki atribut “itemscope” di dalamnya terdapat informasi dalam format terstruktur.
- itemtype digunakan untuk mendefinisikan jenis dari data terstruktur yang terkandung dalam elemen yang memiliki “itemscope”. Misalnya, pada code diatas, “itemtype="http://data-vocabulary.org/Person"” mengindikasikan bahwa data terstruktur dalam elemen “<section>” merupakan tipe "Person" dan mengikuti aturan http://data-vocabulary.org.
- itemprop digunakan untuk menandai elemen HTML sebagai atribut dari data terstruktur. Misal, “<h1 itemprop="name">Meizano Ardhi Muhammad</h1>” menunjukkan bahwa teks "Meizano Ardhi Muhammad" adalah properti "name" dari entitas "Person". Sementara, “<a itemprop="url" href="http://meizano.github.io">Github</a>” mengindikasikan bahwa tautan tersebut adalah properti "url" dari entitas yang sama, yaitu "Person."

#### 2. Jelaskan mengenai hasil dari Google Rich Snippets Testing Tool!

Google Rich Snippets Testing Tool merupakan alat yang digunakan untuk memeriksa dan memvalidasi penggunaan microdatadata terstruktur dalam halaman web. Dengan alat ini, Kita dapat menentukan apakah data terstruktur diinterpretasikan dengan benar oleh browser dan apakah tidak ada masalah dalam penggunaannya.

Hasilnya mencakup

- Status keberhasilan atau kegagalan pengujian data terstruktur.
- Informasi tentang jenis data terstruktur yang teridentifikasi
- Detail tentang atribut “itemprop” dan nilainya diinterpretasikan.
- Tautan yang menunjukkan bagaimana data terstruktur mungkin akan ditampilkan pada broser.

## 5.8 Percobaan 2-8 Drag dan Drop

### 5.8.a Source Code Percobaan 2-8 Drag dan Drop

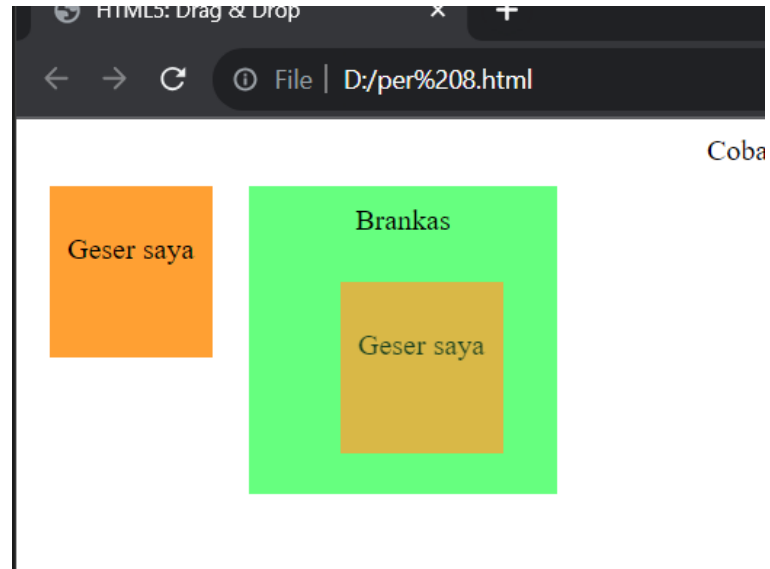
```
D: > per 8.html > html > head > style > height
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML5: Drag & Drop</title>
6 <style>
7 #boxA,
8 #boxB {
9 float: left;
10 padding: 10px;
11 margin: 10px;
12 -moz-user-select: none;
13 }
14 #boxA {
15 background-color: rgb(255, 160, 51);
16 }
17 width: 75px;
18 height: 75px;
19 }
20 #boxB {
21 background-color: rgb(102, 255, 127);
22 width: 150px;
23 height: 150px;
24 }
25 }
26 </style>
27 <script type="text/javascript">
28 function dragStart(ev) {
29 ev.dataTransfer.effectAllowed = 'move';
30 ev.dataTransfer.setData("Text", ev.target.getAttribute('id'));
31 ev.dataTransfer.setDragImage(ev.target, 0, 0);
32 return true;
33 }
34 function dragEnter(ev) {
35 event.preventDefault();
36 return true;
37 }
38 function dragOver(ev) {
39 return false;
40 }
41 function dragDrop(ev) {
42 var src = ev.dataTransfer.getData("Text");
43 ev.target.appendChild(document.getElementById(src));
44 ev.stopPropagation();
45 return false;
46 }
47 </script>
48 </head>
49 <body>
50 <center>
51 <div>Coba gerakan kotak jingga ke kotak hijau.</div>
52 <div id="boxA" draggable="true" ondragstart="return dragStart(event)">
53 <p>Geser saya</p>
54 </div>
55 <div id="boxB" ondragenter="return dragEnter(event)" ondrop="return
56 dragDrop(event)" ondragover="return dragOver(event)">Brankas</div>
57 </center>
58 </body>
59 </html>
```

Gambar 5.8.a Source Code Percobaan 2-8 Drag dan Drop

Berdasarkan gambar diatas, dapat dilihat bahwa pada baris ke-1-5 merupakan deklarasi dokumen HTML5 dengan karakter set UTF-8 dan judul halaman "HTML5 Drag & Drop. Baris 6-20 Bagian ini terdapat elemen "<style>"

yang digunakan untuk mendefinisikan gaya (styling) elemen pada halaman. “#boxA” memiliki warna oranye dengan lebar dan tinggi 75px.”#boxB” memiliki warna hijau dengan lebar dan tinggi 150px. Baris 27-47 merupakan blok JavaScript dimulai dengan tag “<script>“. Di dalamnya, terdapat fungsi “dragStart(ev)”, Fungsi ini akan dipanggil saat elemen "boxA" mulai diseret menggunakan mouse. kemudian “dragEnter(ev)”, Fungsi ini dipanggil saat elemen yang diseret memasuki area "boxB." Fungsi ini mencegah tindakan yang biasanya terjadi saat elemen memasuki area lain. selanjutnya ada “dragOver(ev)”, Fungsi ini dipanggil saat elemen yang sedang diseret berada di atas "boxB." Fungsi ini mengembalikan “false” untuk menghentikan tindakan default yang terjadi.”dragDrop(ev)” Fungsi ini dipanggil saat elemen yang sedang diseret dilepaskan ke dalam "boxB." Ini mengambil ID elemen yang diseret, dan memindahkan elemen tersebut ke dalam "boxB,“. Baris 49-58 pada “<body>“. Didalamnya terdapat instruksi untuk mencoba menyeret kotak jingga ke dalam kotak hijau. Kemudian, terdapat dua div dengan ID "boxA" dan "boxB." "boxA" dapat diseret “draggable="true"” dan memiliki teks "Geser saya," sementara "boxB" memiliki fungsi event untuk operasi drag-and-drop.

### 5.8.b Output Percobaan 2-8 Drag dan Drop



Gambar 5.8.b Output Percobaan 2-8 Drag dan Drop

Berdasarkan gambar di atas, Dapat dilihat bahwa output menampilkan kotak jingga dan kotak hijau yang memungkinkan menggunakan operasi drag-and-drop. Menggerakkan kotak jingga (“boxA”) ke dalam kotak hijau (“boxB”) menggunakan drag-and-drop. Ketika kotak jingga ditarik dan dilepaskan di dalam kotak hijau, kotak jingga akan berpindah ke dalam kotak hijau.

### 5.8.c Jawaban Soal Percobaan 2-8 Drag and Drop

1. Jelaskan atribut draggable, ondragstart, ondragenter, ondrop, dan ondragover!
  - draggable Menentukan apakah elemen dapat di-drag atau tidak.
  - ondragstart Digunakan untuk menentukan tindakan saat elemen di-drag, untuk mengatur data yang akan dipindahkan.
  - ondragenter saat elemen lain masuk ke dalam elemen yang sedang di-drag.
  - ondrop Digunakan untuk menentukan tindakan saat elemen di-drop dan mengambil data menggunakan “getData”.
  - ondragover Digunakan untuk mengatur tindakan saat elemen di atas elemen target selama operasi drag.
2. Jelaskan tentang method effectAllowed, setData, setDragImage, dan getData pada dataTransfer!
  - effectAllowed Mengatur jenis efek yang diinginkan selama operasi drag-and-drop.
  - setData Menyimpan data yang akan dipindahkan ketika operasi drag.
  - setDragImage Menentukan gambar ikon selama drag-and-drop.
  - getData Mengambil data yang telah disimpan selama operasi drag-and-drop.
3. Apakah kegunaan stopPropagation!

stopPropagation digunakan untuk menghentikan penyebaran event ke elemen lain pada DOM. Ini membantu mengendalikan bagaimana event diproses dan memungkinkan untuk mengisolasi tindakan pada elemen tertentu.

## VI. KESIMPULAN

Adapun kesimpulan percobaan ini adalah sebagai berikut

1. Berdasarkan Percobaan 2-2 Formulir 2.0-1 semantik HTML dapat disimpulkan bahwa pada baris ke-9 hingga ke-23, codingan ini menggunakan elemen-elemen HTML5 seperti “<header>“, “<nav>“, “<article>“, “<aside>“, dan “<footer>“ untuk mengorganisasi konten halaman dengan semantik yang kuat. Hal ini mempermudah pemahaman struktur halaman web oleh pembaca dan mesin pencari.
2. Berdasarkan Percobaan 2-4 Canvas dapat disimpulkan bahwa Kode fungsi “animasi()” memanggil “gambarCinta()” pada HTML ini menciptakan halaman web dengan animasi simbol cinta yang berputar pada elemen “<canvas>“.
3. Berdasarkan Percobaan 2-5 Audio dan Video dapat disimpulkan bahwa juga memuat elemen “<video>” yang memungkinkan pemutaran video. Video dalam format MP4 ditentukan dalam elemen video. Tombol "Play" digunakan untuk memulai pemutaran video. Jika browser tidak mendukung elemen video, pesan alternatif akan ditampilkan.
4. Berdasarkan Percobaan 2-7 Microdata dapat disimpulkan bahwa fungsi “showLocation” dipanggil untuk menampilkan latitude dan longitude user. Sedangkan fungsi “errorHandler” menangani pesan kesalahan jika terjadi masalah.
5. Berdasarkan Percobaan 2-8 Drag dan Drop dapat disimpulkan bahwa implementasi fungsi drag-and-drop dengan pengaturan tampilan khusus. Kode ini menciptakan dua kotak, "boxA" yang dapat dijatuhkan (drag) dan "boxB" sebagai area target. Fungsi JavaScript dalam kode mengendalikan perilaku selama operasi drag-and-drop, mulai dari saat "boxA" mulai ditarik, masuk ke dalam "boxB," hingga dilepaskan di dalam "boxB." Ini memberikan user pengalaman interaktif untuk mencoba operasi drag-and-drop di antara dua kotak.

## DAFTAR PUSTAKA

- S. M. D. I. S. M. D. I. F. S. S. M. I. M. F. S. M. P. W. G. S. M. I. A. S. M. H. E. S. M. Iqbal  
Ramadhani Mukhlis, BUKU AJAR PEMROGRAMAN WEB 1, jambi PT. Sonpedia  
Publishing Indonesia, 2023 .*
- Y. Juandy, HTML 5.0 MULTIMEDIA WEBSTANDAR, Bandung STMIK LIKMI, 2010.*