

# MODUL PRAKTIKUM

MULTIMEDIA ONLINE

Versi 1.0

**Disusun Oleh:**

Farid Suryanto, MT

Habib Baharudin

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI TERAPAN

UNIVERSITAS AHMAD DAHLAN

2019

## Kata Pengantar

Modul praktikum ini merupakan panduan kegiatan praktikum mata kuliah Multimedia Online yang memuat serangkaian petunjuk teknis pembuatan aplikasi multimedia berbasis web. Secara spesifik, modul ini mengeksplorasi kanvas HTML untuk membuat dan menyusun elemen-elemen multimedia menjadi sebuah aplikasi berbasis web. Pada dunia industri teknologi, teknologi multimedia berbasis web sangat berperan untuk meningkatkan pengalaman pengguna. Teknologi multimedia berbasis web dapat digunakan untuk membuat aplikasi permainan (game), efek-efek animasi dalam aplikasi web, atau membuat simulasi multimedia berbasis web yang dapat diakses oleh pengguna melalui berbagai macam perangkat (cross-platform).

Setelah mengikuti kegiatan praktikum ini, diharapkan mahasiswa dapat menguasai penggunaan kanvas HTML untuk membuat aplikasi multimedia berbasis web. Selain itu, diharapkan mahasiswa juga dapat menguasai penggunaan beberapa library kanvas HTML yang tersedia misalnya konvaJS, fabricJS, paperJS, dll untuk membuat aplikasi multimedia berbasis web yang lebih menarik.

Terima kasih kepada tim asisten praktikum Multimedia Online Siti Kartika dan Habib Baharudin yang telah memberikan banyak kontribusi dalam proses penyusunan modul versi 1.0. Sebagai pelengkap, modul ini disertai dengan video tutorial yang dapat diakses melalui [playlist youtube](#).

Yogyakarta, Juni 2019

**Tim Penulis**

## Pra Praktikum

# Multimedia Online

## Prosedur Pra-Praktikum

1. Setiap mahasiswa wajib memiliki akun *github.com* dan melakukan konfigurasi pada akun masing-masing sedemikian rupa sehingga dapat membuat halaman untuk menampilkan hasil praktikum setiap pekan.
2. Sebelum mengikuti praktikum mahasiswa wajib membuat tampilan pada halaman github masing-masing yang berisi tampilan seperti dibawah ini:

### **Hasil Praktikum Multimedia**

Praktikum 1 - Canvas HTML

Praktikum 2 - Interaktifitas

..

..

3. List praktikum yang diikuti merupakan tautan yang nantinya akan mengarah pada hasil praktikum setiap pekan.

## Praktikum 1

# Canvas HTML

---

### Ringkasan

Bagian ini membahas tentang dasar penggunaan canvas HTML sebagai media untuk membuat berbagai macam elemen multimedia berbasis web.

### Topik

- Pembuatan elemen dan pengaturan canvas HTML
- Pembuatan responsive canvas
- Membuat dan memanipulasi objek vektor primitif: segi empat, garis, segitiga, dan lingkaran
- Layering

### Prosedur

1. Tutorial tentang dasar canvas HTML dapat di baca dan pahami pada prosedur di bawah ini.

Langkah-langkah membuat bentuk-bentuk dasar bangun ruang pada canvas HTML :

- a. langkah pertama adalah membuat canvas atau sebagai lembar kerja untuk membuat objek 2 Dimensi atau 3 Dimensi. Untuk mengatur warna kanvas bisa menggunakan `<style> css` dan untuk mengatur ukuran lebar kanvas bisa menggunakan syntax `<canvas width="">` dan untuk mengatur tinggi kanvas bisa menggunakan `<canvas height="">`

```

<!DOCTYPE html>
<html>

<head>
  <title>My first Canvas App</title>
  <style type="text/css">
    canvas {
      background-color: #ccc;
    };
  </style>
</head>
<body>
  <canvas width="800px" height="450px" id="canvas"></canvas>
  <script type="text/javascript">
    var c = document.getElementById('canvas');
    var ctx = c.getContext('2d');
  </script>
</body>
</html>

```

- b. Langkah kedua adalah membuat objek bangun ruang pada canvas HTML, kode di bawah ini adalah untuk membuat objek lingkaran

```

ctx.beginPath();
ctx.arc(400, 225, 120, 0, 2 * Math.PI);
ctx.fillStyle = "#aa0000";
ctx.fill();
ctx.stroke();
ctx.beginPath();
ctx.arc(400, 225, 80, 0, 2 * Math.PI);
ctx.fillStyle = "#dab420";
ctx.fill();
ctx.stroke();

```

```
ctx.arc(x, y, r, sudutAwal, sudutAkhir, arah);
```

- x menentukan koordinat Objek yang akan di buat ke sumbu X
- y menentukan koordinat Objek yang akan di buat ke sumbu Y
- r menentukan jari-jari lingkaran yang akan di buat
- sudutAwal dengan satuan radian ( 0 radian berarti dimulai dari arah jam 3 )
- sudutAkhir dengan satuan radian

- arah dalam penggambaran lingkaran seperti di contoh arahnya adalah  $2 * \text{Math.PI}$  maka lingkaran yang akan di buat berbentuk 360 derajat atau lingkaran utuh.
- untuk pewarnaan bisa menggunakan syntax `fillStyle` ( menggunakan pewarnaan RGB )

c. langkah selanjutnya adalah membuat bentuk persegi, contohnya ada seperti kode di bawah ini :

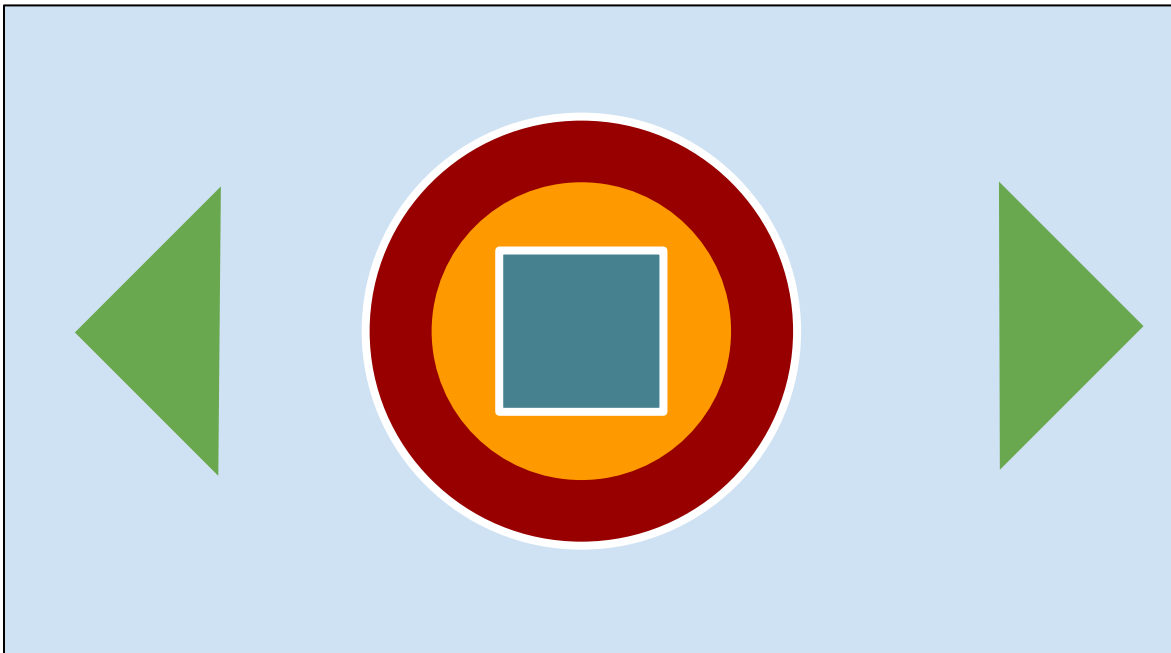
```
//persegi
ctx.beginPath();
ctx.lineWidth = "5";
ctx.strokeStyle = "white";
ctx.rect(350, 170, 100, 100);
ctx.fillStyle = "#6495ed";
ctx.fill();
ctx.stroke();
```

- `lineWidth` untuk menentukan tebal tipisnya outline dari persegi yang akan di buat
- `strokeStyle` untuk menentukan warna dari outlinenya ( menggunakan pewarnaan RGB )
- `ctx.rect` (sumbuX, sumbuY, lebar, tinggi)
- `fillStyle` untuk menentukan warna dari persegi yang akan di buat

d. Langkah terakhir adalah membuat bentuk segitiga untuk contoh kodenya ada di bawah ini :

```
ctx.beginPath();
ctx.moveTo(650, 130);
ctx.lineTo(650, 320);
ctx.lineTo(750, 230);
ctx.closePath();
ctx.fillStyle = "#aab001";
ctx.fill();
ctx.beginPath();
ctx.moveTo(150, 130);
ctx.lineTo(150, 320);
ctx.lineTo(50, 230);
ctx.closePath();
ctx.fillStyle = "#aab001";
ctx.fill();
```

- `ctx.moveTo(650, 130);`  
`ctx.lineTo(650, 320);`  
Untuk menggambar garis dari titik awal ke titik ke-dua,
- `ctx.lineTo(750, 230);`  
Untuk menggambar garis dari titik kedua ke titik ketiga
- `ctx.closePath();`  
Untuk menghubungkan garis dari titik awal ke titik akhir.



**Gambar 1** Tampilan hasil akhir

Kode Lengkapnya ada di bawah ini :

```
<!DOCTYPE html>
<html>

<head>
  <title>My first Canvas App</title>
  <style type="text/css">
    canvas {
      background-color: #ccc;
    }
  ;
```

```

</style>
</head>

<body>
  <canvas width="800px" height="450px" id="canvas"></canvas>
  <script type="text/javascript">
    var c = document.getElementById('canvas');
    var ctx = c.getContext('2d');

    //lingkaran
    ctx.beginPath();
    ctx.arc(400, 225, 120, 0, 2 * Math.PI);
    ctx.fillStyle = "#aa0000";
    ctx.fill();
    ctx.stroke();

    ctx.beginPath();
    ctx.arc(400, 225, 80, 0, 2 * Math.PI);
    ctx.fillStyle = "#dab420";
    ctx.fill();
    ctx.stroke();

    //persegi
    ctx.beginPath();
    ctx.lineWidth = "5";
    ctx.strokeStyle = "white";
    ctx.rect(350, 170, 100, 100);
    ctx.fillStyle = "#6495ed";
    ctx.fill();
    ctx.stroke();

    ctx.beginPath();
    ctx.moveTo(650, 130);
    ctx.lineTo(650, 320);
    ctx.lineTo(750, 230);
    ctx.closePath();
    ctx.fillStyle = "#aab001";
    ctx.fill();

    ctx.beginPath();

```



```
    ctx.moveTo(150, 130);  
    ctx.lineTo(150, 320);  
    ctx.lineTo(50, 230);  
    ctx.closePath();  
    ctx.fillStyle = "#aab001";  
    ctx.fill();  
  
    </script>  
</body>  
  
</html>
```

2. Simpan file dengan praktikum\_1\_NIM.html
3. Unggah file pada repositori github masing-masing dan masukkan tautan file tersebut pada halaman utama github.

## Praktikum 2

# Vector Pattern

---

### Ringkasan

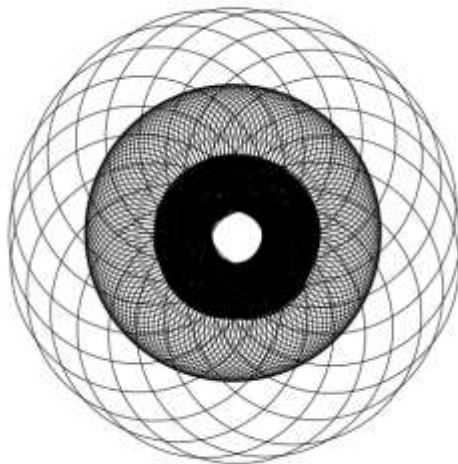
Multimedia berbasis web dapat digunakan untuk berbagai macam keperluan. Salah satunya adalah untuk membuat konten digital art. Bagian ini melatih mahasiswa untuk membuat konten digital art menggunakan vector canvas yang dibuat programmatically menggunakan konsep looping (perulangan) pada javascript.

### Topik

- Membuat drawing function yang reusable pada javascript
- Menentukan formula matematis untuk membuat pola grafis

### Prosedur

1. Simak prosedur cara membuat vector pattern pada modul berikut ini.
2. Membuat pola vektor menggunakan looping sehingga hasil mendekati tampilan dibawah.



3. Langkah-langkah untuk membuat pola vector pattern dengan looping
  - a. langkah pertama adalah membuat 3 canvas yang akan di buat secara berlayer atau bertumpuk mulai dari layer 1, layer 2, dan layer 3 dengan ukuran yang sama.

```
<html><head>
  <meta charset="UTF-8">
  <title>Canvas Layer Test</title>
  <style type="text/css">
    body {
      position: relative;
      margin: 0px;
    }
    #layer1 {
      z-index: 1;
      position: absolute;
      width: 100%;
    }
    #layer2 {
      z-index: 2;
      position: absolute;
      width: 100%;
    }
    #layer3 {
      z-index: 3;
      position: absolute;
      width: 100%;
    }
  </style>
</head>
<body onresize="init()">
  <canvas id="layer1" width="811" height="657"></canvas>
  <canvas id="layer2" width="811" height="657"></canvas>
  <canvas id="layer3" width="811" height="657"></canvas>
  <script type="text/javascript">
    var layer1, layer2, layer3, ctx1, ctx2, ctx3, screenW, screenH;

    function init() {
      layer1 = document.getElementById('layer1');
      layer2 = document.getElementById('layer2');
      layer3 = document.getElementById('layer3');
      ctx1 = layer1.getContext('2d');
      ctx2 = layer2.getContext('2d');
```

```

    ctx3 = layer3.getContext('2d');

    screenW = window.innerWidth;
    screenH = window.innerHeight;
    layer1.width = screenW;
    layer1.height = screenH;
    layer2.width = screenW;
    layer2.height = screenH;
    layer3.width = screenW;
    layer3.height = screenH;
    draw(ctx3, 50, 55, 10, 300);
    draw(ctx2, 200, 130, 5, 20);
    draw(ctx1, 200, 50, 0, 120);
}

```

lalu pada function init () adalah fungsi yang nantinya akan di gunakan untuk memanggil canvas yang sudah di buat.

screenW dan screenH berfungsi untuk membuat canvas menjadi responsif

b. Langkah kedua membuat fungsi lingkaran kode programnya ada di bawah ini :

```

function createCircle(context, x, y, size) {
    context.beginPath();
    context.arc(x, y, size, 0, 2*Math.PI);
    context.stroke();
}

```

c. Setelah fungsi lingkaran di buat makan selanjutnya adalah memanggil fungsi tersebut lalu di panggil secara berulang atau looping dengan fungsi berikut ini :

```

function draw(context, radius, circleSize, randomSize, totalObject) {
    context.clearRect(0, 0, screenW, screenH);
    var total_dot = totalObject;
    var radius = radius;
    var dots = [];
    for (var i=1; i<=total_dot; i++) {
        sudut = 360/total_dot*i;
        posx = Math.cos(sudut*(Math.PI/180))*radius/2;
        posy = Math.sin(sudut*(Math.PI/180))*radius/2;
        var newPosx = posx + (screenW/2);
        var newPosy = posy + (screenH/2);
        var randSize = randomSize;
        var random1 = Math.floor((Math.random() * randSize));
    }
}

```

```

        var random2 = Math.floor((Math.random() * randSize));
        createCircle(contex, newPosx+random1, newPosy+random2,
circleSize);
    }
}

```

- d. Langkah terakhir adalah memanggil fungsi yang sudah di buat tadi ke dalam fungsi `init()`.

maka dalam fungsi `init ()` fungsi yang sudah di panggil akan di masukkan dalam canvas yang ada pada fungsi `init ()`.

```
init();
```

Kode program lengkapnya ada di bawah ini :

```

<html>
<head>
  <meta charset="UTF-8">
  <title>Canvas Layer Test</title>
  <style type="text/css">
    body {
      position: relative;
      margin: 0px;
    }
    #layer1 {
      z-index: 1;
      position: absolute;
      width: 100%;
    }
    #layer2 {
      z-index: 2;
      position: absolute;
      width: 100%;
    }
    #layer3 {
      z-index: 3;
      position: absolute;
      width: 100%;
    }
  </style>

```

```

</style>
</head>
<body onresize="init()">
  <canvas id="layer1" width="811" height="657"></canvas>
  <canvas id="layer2" width="811" height="657"></canvas>
  <canvas id="layer3" width="811" height="657"></canvas>
  <script type="text/javascript">
    var layer1, layer2, layer3, ctx1, ctx2, ctx3, screenW, screenH;

    function init() {
      layer1 = document.getElementById('layer1');
      layer2 = document.getElementById('layer2');
      layer3 = document.getElementById('layer3');
      ctx1 = layer1.getContext('2d');
      ctx2 = layer2.getContext('2d');
      ctx3 = layer3.getContext('2d');

      screenW = window.innerWidth;
      screenH = window.innerHeight;
      layer1.width = screenW;
      layer1.height = screenH;
      layer2.width = screenW;
      layer2.height = screenH;
      layer3.width = screenW;
      layer3.height = screenH;

      draw(ctx3, 50, 55, 10, 300);
      draw(ctx2, 200, 130, 5, 20);
      draw(ctx1, 200, 50, 0, 120);
    }

    function draw(context, radius, circleSize, randomSize,
totalObject) {
      context.clearRect(0, 0, screenW, screenH);
      var total_dot = totalObject;
      var radius = radius;
      var dots = [];

      for (var i=1; i<=total_dot; i++) {
        sudut = 360/total_dot*i;
        posx = Math.cos(sudut*(Math.PI/180))*radius/2;
        posy = Math.sin(sudut*(Math.PI/180))*radius/2;

```

```
        var newPosx = posX + (screenW/2);
        var newPosy = posy + (screenH/2);

        var randSize = randomSize;
        var random1 = Math.floor(Math.random() * randSize);
        var random2 = Math.floor(Math.random() * randSize);

        createCircle(context, newPosx+random1, newPosy+random2,
circleSize);
    }
}

function createCircle(context, x, y, size) {
    context.beginPath();
    context.arc(x, y, size, 0, 2*Math.PI);
    context.stroke();
}

init();
</script>

</body></html>
```

4. Simpan file dengan praktikum\_2\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github.

## Praktikum 3

# Interaktivitas

---

### Ringkasan

Bagian ini mempelajari tentang interaktivitas pada aplikasi web menggunakan pointer. Diharapkan mahasiswa dapat memahami event menggunakan mouse dan keyboard serta mengambil nilai masukan dari kedua perangkat untuk dikembangkan menjadi aplikasi menggambar sederhana.

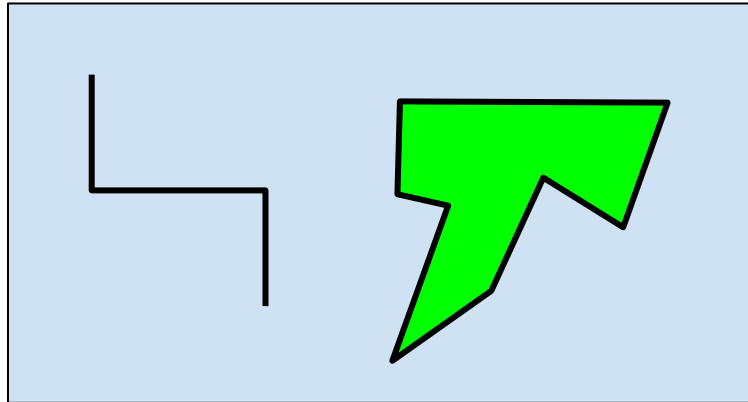
### Topik

- mouse event
- keyboard event

### Prosedur

1. Simak dan baca tutorial pada modul berikut ini
2. Mem aplikasi web sederhana dengan ketentuan:
  - a. Saat pointer klik maka aplikasi membuat garis sesuai dengan titik yang terdeteksi oleh pointer
  - b. Jika ditekan enter pada keyboard maka titik titik yang dihubungkan oleh garis tersebut berubah menjadi poligon yang memiliki fill (berwarna pada area dalam)
  - c. Ilustrasi:





- d. Untuk membuat seperti contoh diatas dalam canvas HTML kita memerlukan suatu event yaitu mouseEvent. MouseEvent sendiri memiliki beberapa tipe peristiwa atau event dapat dilihat pada tabel berikut ini :

**Tabel 1.1** Tipe mouse event

Event	Deskripsi
onclick	Peristiwa saat mouse di klik di atas elemen
ondblclick	Peristiwa saat mouse di klik dua kali (double click) diatas elemen
onmousedown	Peristiwa saat tombol mouse ditekan di atas elemen
onmouseenter	Peristiwa saat kursor bergerak masuk diatas elemen
onmouseleave	Peristiwa saat kursor bergerak keluar dari atas elemen
onmousemove	Peristiwa terjadi ketika pointer bergerak saat berada di atas elemen
onmouseout	Peristiwa terjadi ketika pengguna menggerakkan penunjuk kursor dari suatu elemen termasuk elemen yang ada didalamnya (children element)
onmouseover	Peristiwa terjadi ketika pointer dipindahkan ke atas elemen, atau ke salah satu dari elemen didalamnya (children element)
onmouseup	Peristiwa terjadi ketika pengguna melepaskan tombol mouse di atas elemen

3. Langkah-langkah untuk membuatnya dapat mengikuti prosedur dibawah ini :
- Langkah pertama adalah membuat canvas atau lembar kerjanya untuk pembuatannya dilihat pada kode program dibawah ini :

```

<!DOCTYPE html>
<html>
<head>
  <title>Click mouse and draw line</title>
  <style type="text/css">
    canvas{
      background-color: #ccc;
    }
  </style>
</head>
<body>
<canvas height="450" width="800" id="canvas"></canvas>
<script type="text/javascript">
  var c=document.getElementById('canvas');
  var ctx=c.getContext('2d');

```

- b. Setelah itu buat variable untuk menampung hasil dari setiap mouse clicknya dalam bentuk array

```
var lines=[];
```

- c. Langkah selanjutnya adalah membuat kondisi dimana ketika mousedown akan menjalankan fungsi drawline.

```
document.body.addEventListener('mousedown',drawline);
```

- d. Selanjutnya membuat fungsi drawline

```

function drawline(e)
{
  var x=e.offsetX;
  var y=e.offsetY;
  if(lines.length==0)
  {
    ctx.moveTo(x,y);
  }
  ctx.lineTo(x,y);
  ctx.lineWidth=2;
  ctx.strokeStyle='black';
  ctx.stroke();
  lines.push([x,y]);
  var sprite={
    x:lines[0][0],
    y:lines[0][1]

```

```

    };
    var target={
        x:lines[lines.length-1][0],
        y:lines[lines.length-1][1]
    };
    isCloseTo(sprite,target);
}

```

dari setiap line yang dibuat oleh setiap mousedown akan dimasukkan kedalam fungsi isCloseTo.

- e. Langkah terakhir adalah membuat fungsi isCloseTo fungsi ini akan berjalan ketika titik awal mousedown bertemu dengan titik akhir mousedown maka akan membuat objek baru

```

function isCloseTo(sprite,target)
{
    var dist=Math.sqrt(((sprite.x-target.x)*(sprite.x-
target.x))+((sprite.y-target.y)*(sprite.y-target.y)));
    console.log(dist);
    if(lines.length>1 && dist<6)
    {
        console.log('make polygon');
        ctx.lineTo(target.x,target.y);
        ctx.fillStyle = 'red';
        ctx.fill();
        lines=[];
    }
}

</script>
</body>
</html>

```

Kode lengkapnya ada di bawah ini :

```

<!DOCTYPE html>
<html>

```

```

<head>
  <title>Click mouse and draw line</title>
  <style type="text/css">
    canvas{
      background-color: #ccc;
    }
  </style>
</head>
<body>
<canvas height="450" width="800" id="canvas"></canvas>
<script type="text/javascript">
  var c=document.getElementById('canvas');
  var ctx=c.getContext('2d');

  var lines=[];

  document.body.addEventListener('mousedown',drawline);
  function drawline(e)
  {
    var x=e.offsetX;
    var y=e.offsetY;
    if(lines.length==0)
    {
      ctx.moveTo(x,y);
    }
    ctx.lineTo(x,y);
    ctx.lineWidth=2;
    ctx.strokeStyle='black';
    ctx.stroke();
    lines.push([x,y]);
    var sprite={
      x:lines[0][0],
      y:lines[0][1]
    };
    var target={
      x:lines[lines.length-1][0],
      y:lines[lines.length-1][1]
    };
    isCloseTo(sprite,target);
  }

  function isCloseTo(sprite,target)
  {

```

```
        var dist=Math.sqrt(((sprite.x-target.x)*(sprite.x-
target.x))+((sprite.y-target.y)*(sprite.y-target.y)));
        console.log(dist);
        if(lines.length>1 && dist<6)
        {
            console.log('make polygon');
            ctx.lineTo(target.x,target.y);
            ctx.fillStyle = 'red';
            ctx.fill();
            lines=[];
        }
    }
</script>
</body>
</html>
```

4. Simpan file dengan praktikum\_3\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github.

## Praktikum 4

# Puzzle Project

---

### Ringkasan

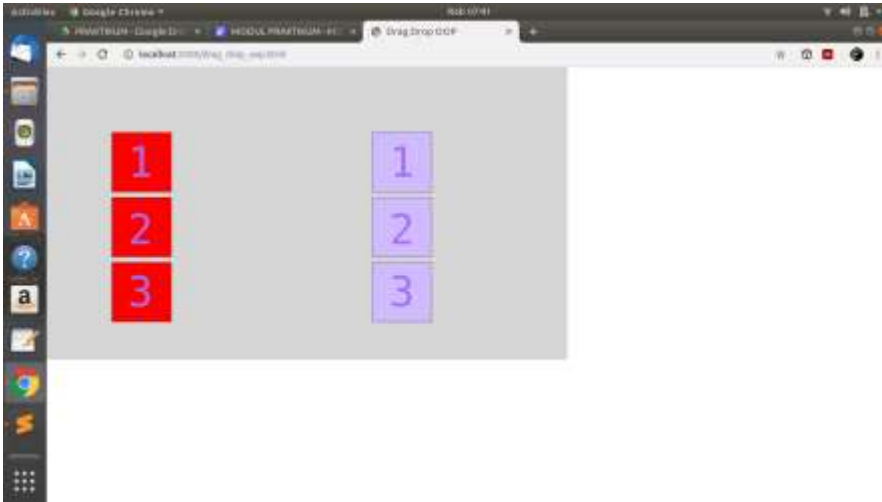
Salah satu interaktivitas menggunakan pointer event yang dapat dilakukan pengguna adalah melakukan drag & drop. Aktivitas ini dapat diterapkan untuk berbagai macam kebutuhan misalnya melakukan editing objek pada graphic editor atau menyusun objek-objek seperti pada permainan puzzle. Bagian ini melatih mahasiswa untuk menerapkan konsep interaktivitas untuk membuat aplikasi puzzle sederhana.

### Topik

- Mengambil gambar (svg, png, jpeg) dari sumber daya luar ke dalam halaman web
- Menerapkan interaktivitas pada gambar
- Menghitung jarak antara objek dengan memanfaatkan mouseEvent sebagai trigger

### Prosedur

1. Simak dan pahami prosedur pembuatannya pada modul dibawah ini
2. Membuat aplikasi puzzle sederhana berbasis web. Gambaran aplikasi tersebut adalah sebagai berikut:



Objek puzzle adalah pasangan gambar yang jika puzzle di drag dan di drop dengan jarak tertentu terhadap target maka puzzle tersebut akan menempel pada target yang benar.

3. Langkah-langkah dalam membuatnya dapat diikuti pada prosedur di bawah ini :
  - a. Langkah pertama adalah membuat fungsi startGame yang digunakan sebagai program utama dalam menjalankan game puzzle,

```
<!DOCTYPE html>
<html>
  <head>
    <title>Drag Drop OOP</title>
  </head>
  <body>
    <script>
      var sprites=[], targets=[];
      function startGame() {
        myApp.start();
        for(var i=1; i<4; i++)
        {
          sprites[i]=new sprite('assets/obj_'+i+'.png',100,i*100);
        }
      }
    </script>
  </body>
</html>
```

```

        targets[i]=new sprite('assets/target_'+i+'.png',500,i*100);
        draggable(sprites[i],targets[i]);
    }
}

```

var sprites= [], targets=[] berfungsi untuk menyimpan objek-objek yang dipanggil dari aset dalam bentuk array.

- b. Langkah selanjutnya adalah membuat kanvas atau lembar kerja untuk meletakkan objek-objek yang akan digunakan sebagai puzzle. Kode programnya dapat dilihat dibawah ini :

```

var myApp = {
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 800;
        this.canvas.height = 450;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
        this.interval = setInterval(appLoop, 20);
    },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
    }
}

```

- c. Setelah membuat canvasnya maka selanjutnya adalah meletakkan objek yang sudah di panggil pada fungsi pertama tadi ke dalam canvas.

```

function sprite(url,x,y)
{
    this.x=x;
    this.y=y;
    this.width=100;
    this.height=100;
    var ctx=myApp.context;
    var imgObject= new Image();
    imgObject.src = url;
    ctx.drawImage(imgObject,this.x,this.y);
    this.update=function()

```



```

    {
        ctx.drawImage(imgObject,this.x,this.y);
    }
}
function appLoop()
{
    myApp.clear();
    for(var i=1; i<4;i++)
    {
        targets[i].update();
        //sprites[i].update();
    }
    for(var i=1; i<4;i++)
    {
        //targets[i].update();
        sprites[i].update();
    } }

```

- d. Langkah terakhir adalah membuat fungsi untuk membuat objek gambar yang ada di kanvas bisa di drag and drop.

```

function draggable(sprite,target)
{
    var offsetX, offsetY;
    document.body.addEventListener('mousedown',function(e)
    {
        offsetX=e.clientX-sprite.x;
        offsetY=e.clientY-sprite.y;
        //mendeteksi pointer klik diatas objek lewat fungsi mouseIsOver
        if(mouseIsOver(e.clientX,e.clientY,sprite))
        {
            //jika klik diatas objek, deteksi pergerakan mouse dan kondisi
            saat mouse lepas
            //panggil fungsi mouseMove saat pointer bergerak
            document.body.addEventListener('mousemove',mouseMove);
            //panggil fungsi mouseUp saat pointer lepas
            document.body.addEventListener('mouseup',mouseUp);
        }
    });
    function mouseMove(e)
    {

```

```

        //gambar objek baru lewat fungsi draw() dengan posisi x dan y sama
        dengan posisi x dan y pointer
        //menambahkan faktor offset agar posisi drag objek tidak bergeser
        sprite.x=e.clientX-offsetX, sprite.y=e.clientY-offsetY;
    };
    function mouseUp(e)
    {
        //hapus event untuk mousemove dan mouseup agar kondisi kembali
        seperti semula
        document.body.removeEventListener('mousemove',mouseMove);
        document.body.removeEventListener('mouseup',mouseUp);
        calculateDistance();
    }
    //fungsi untuk mendeteksi posisi pointer diatas objek
    function mouseIsOver (mouseX,mouseY,obj)
    {
        if(mouseX>obj.x && mouseX < obj.x+obj.width && mouseY > obj.y &&
        mouseY<obj.y+obj.height)
            return true;
        else return false;
    }
    //menghitung jarak antara sprite dengan target
    function calculateDistance()
    {
        var dist=Math.sqrt(((sprite.x-target.x)*(sprite.x-
        target.x))+((sprite.y-target.y)*(sprite.y-target.y)));
        if(dist<30){
            sprite.x=target.x;
            sprite.y=target.y;
        }
    }
};
startGame();
</script>
</body>
</head>

```

Kode lengkapnya ada di bawah ini :

```

<!DOCTYPE html>
<html>
  <head>

```

```

    <title>Drag Drop OOP</title>
<head>
    <style type="text/css">
        body{
            margin:0;
        }
        canvas{
            position: absolute;
            background-color: #cccc;

        }
    </style>
<body>
<script>
var sprites=[], targets=[];
function startGame() {
    myApp.start();
    for(var i=1; i<4; i++)
    {
        sprites[i]=new sprite('assets/obj_'+i+'.png',100,i*100);
        targets[i]=new sprite('assets/target_'+i+'.png',500,i*100);
        draggable(sprites[i],targets[i]);
    }
}
var myApp = {
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 800;
        this.canvas.height = 450;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
        this.interval = setInterval(appLoop, 20);
    },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
    }
}
function sprite(url,x,y)
{
    this.x=x;
    this.y=y;

```

```

    this.width=100;
    this.height=100;
    var ctx=myApp.context;
    var imgObject= new Image();
    imgObject.src = url;
    ctx.drawImage(imgObject,this.x,this.y);
    this.update=function()
    {
        ctx.drawImage(imgObject,this.x,this.y);
    }
}

function appLoop()
{
    myApp.clear();
    for(var i=1; i<4;i++)
    {
        targets[i].update();
        //sprites[i].update();
    }
    for(var i=1; i<4;i++)
    {
        //targets[i].update();
        sprites[i].update();
    }
}

function draggable(sprite,target)
{
    var offsetX, offsetY;
    document.body.addEventListener('mousedown',function(e)
    {
        offsetX=e.clientX-sprite.x;
        offsetY=e.clientY-sprite.y;

        if(mouseIsOver(e.clientX,e.clientY,sprite))
        {
            document.body.addEventListener('mousemove',mouseMove);
            document.body.addEventListener('mouseup',mouseUp);
        }
    });
    function mouseMove(e)

```

```

{
    sprite.x=e.clientX-offsetX, sprite.y=e.clientY-offsetY;
};
function mouseUp(e)
{
    document.body.removeEventListener('mousemove',mouseMove);
    document.body.removeEventListener('mouseup',mouseUp);
    calculateDistance();
}
//fungsi untuk mendeteksi posisi pointer diatas objek
function mouseIsOver (mouseX,mouseY,obj)
{
    if(mouseX>obj.x && mouseX < obj.x+obj.width && mouseY > obj.y &&
mouseY<obj.y+obj.height)
        return true;
    else return false;
}
function calculateDistance()
{
    var dist=Math.sqrt(((sprite.x-target.x)*(sprite.x-
target.x))+((sprite.y-target.y)*(sprite.y-target.y)));
    if(dist<30){
        sprite.x=target.x;
        sprite.y=target.y;
    }
}
};
startGame();
</script>
</body>
</head>

```

4. Simpan file dengan praktikum \_4\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github.

## Praktikum 5

# Animasi - Motion Tween

---

### Ringkasan

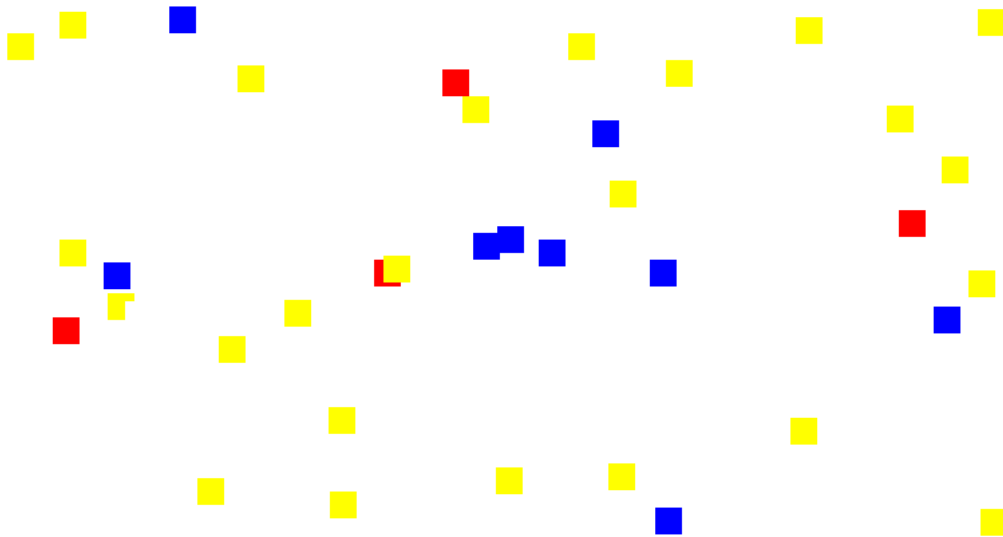
Animasi merupakan transformasi geometris objek secara terurut berdasarkan satuan waktu. Animasi dapat dibuat menggunakan Canvas HTML dengan melakukan perubahan atribut objek dalam interval waktu. Pada canvas HTML, interval waktu dapat dibuat menggunakan fungsi time interval atau animation frame. Animasi dalam aplikasi web dapat digunakan untuk membuat berbagai macam fitur untuk meningkatkan pengalaman pengguna. Bagian ini melatih mahasiswa untuk memanfaatkan fitur animasi pada canvas HTML untuk membuat multiple object motion tween.

### Topik

- Animasi menggunakan time interval
- Animasi menggunakan animation frame
- Animasi untuk multiple object

## Prosedur

1. Simak dan pahami tutorial yang ada pada modul berikut ini.
2. Membuat beberapa objek menggunakan perulangan (looping) pada javascript kemudian buatlah setiap objek bergerak secara berulang. Misalnya, setelah melewati tinggi maksimal canvas maka posisikan objek tersebut pada posisi nol ( $y=0$ ). Gambaran hasil akhir aplikasi adalah seperti pada gambar dibawah ini:



3. Langkah-langkah dalam membuatnya dapat mengikuti prosedur di bawah ini :
  - a. Langkah pertama adalah membuat program utama untuk menjalankan motion tween-nya dengan membuat fungsi startGame.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Animation - Tween Motion</title>
  </head>
  <body>
    <script type="text/javascript">
      // Animation code goes here
    </script>
  </body>
</html>
```

```

<script>
var red, blue, yellow;
var obj=[], objNum=50, color=['red','yellow','blue','white'];
function startGame() {
  for(var i=0; i<objNum; i++)
  {
    var pos=Math.floor(Math.random()*800);
    var speed=Math.floor(Math.random()*10)+1;
    var clr=Math.floor(Math.random()*10);
    obj[i]= new component(20, 20, color[clr], pos, 0,speed);
  }

  myGameArea.start();
}

```

var obj=[] berfungsi untuk menyimpan objek yang sudah di buat dalam bentuk array, lalu di masukkan ke looping atau perulangan di for ( var i = 0, i <objNum; i++) hal ini berfungsi supaya objek yang tampil pada canvas akan tampil secara acak.

- b. Langkah selanjutnya adalah membuat canvas atau lembar kerja untuk menampilkan objek yang di buat.

```

var myGameArea = {
  canvas : document.createElement("canvas"),
  start : function() {
    this.canvas.width = 800;
    this.canvas.height = 450;
    this.context = this.canvas.getContext("2d");
    document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
    this.interval = setInterval(updateGameArea, 20);
  },
  clear : function() {
    this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
  }
}

```

- c. Setelah itu membuat objeknya dengan fungsi di bawah ini :

```

function component(width, height, color, x, y,speed) {
  this.width = width;
  this.height = height;

```



```

    this.x = x;
    this.y = y;
    this.speed=speed;
    this.update = function(){
        ctx = myGameArea.context;
        ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
}

```

dalam fungsi di atas sengaja ukuran objek persegi panjangnya tidak diisi dengan angka supaya nanti ukuran persegi yang dibuat ukurannya akan acak dan tidak sama

- d. Langkah selanjutnya membuat fungsi untuk membuat objek bergerak (motion) secara acak

```

function updateGameArea() {
    myGameArea.clear();
    for(var i=0; i<objNum;i++)
    {
        obj[i].y+=obj[i].speed;
        if(obj[i].y>400){
            var pos=Math.floor(Math.random()*800);
            var speed=Math.floor(Math.random()*10)+1;
            obj[i].y=0;
            obj[i].x=pos;
            obj[i].speed=speed;
        }
        obj[i].update();
    }
}

```

- e. Langkah terakhir adalah memasukkan semua fungsi di atas ke program utamanya yaitu fungsi startGame

```
startGame()
```

```

</script>
</body>
</head>

```

Kode lengkapnya ada di bawah ini :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Animation - Tween Motion</title>
  </head>
  <style type="text/css">
    body{
      margin:0;
    }
    canvas{
      background-color: #cccc;
    }
  </style>
  <body>

  <script>
var red, blue, yellow;
var obj=[], objNum=50, color=['red','yellow','blue','white'];
function startGame() {
  for(var i=0; i<objNum; i++)
  {
    var pos=Math.floor(Math.random()*800);
    var speed=Math.floor(Math.random()*10)+1;
    var clr=Math.floor(Math.random()*10);
    obj[i]= new component(20, 20, color[clr], pos, 0,speed);
  }

  myGameArea.start();
}

var myGameArea = {
```

```

    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 800;
        this.canvas.height = 450;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
        this.interval = setInterval(updateGameArea, 20);
    },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
    }
}

function component(width, height, color, x, y,speed) {
    this.width = width;
    this.height = height;
    this.x = x;
    this.y = y;
    this.speed=speed;
    this.update = function(){
        ctx = myGameArea.context;
        ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
}

function updateGameArea() {
    myGameArea.clear();
    for(var i=0; i<objNum;i++)
    {
        obj[i].y+=obj[i].speed;
        if(obj[i].y>400){
            var pos=Math.floor(Math.random()*800);
            var speed=Math.floor(Math.random()*10)+1;
            obj[i].y=0;
            obj[i].x=pos;
            obj[i].speed=speed;
        }
        obj[i].update();
    }
}

```

```
startGame()
```

```
</script>
```

```
</body>
```

```
</head>
```

4. Simpan file dengan praktikum \_5\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github.

## Praktikum 6

# Animasi Gambar Terurut

---

### Ringkasan

Cara lain untuk membuat animasi pada aplikasi multimedia berbasis web adalah dengan menggunakan serangkaian gambar (raster image) yang ditampilkan secara terurut dalam satuan waktu sehingga menciptakan efek pergerakan. Teknik ini biasa digunakan untuk membuat animasi pada karakter permainan komputer misalnya karakter pemain dan NPC (Non-Player Character).

### Topik

- Menganimasikan gambar terurut
- Kontrol animasi menggunakan keyboard
- Membuat efek pergerakan karakter

### Prosedur

- 1.

# Praktikum 7

## Audio Visualizer

---

### Ringkasan

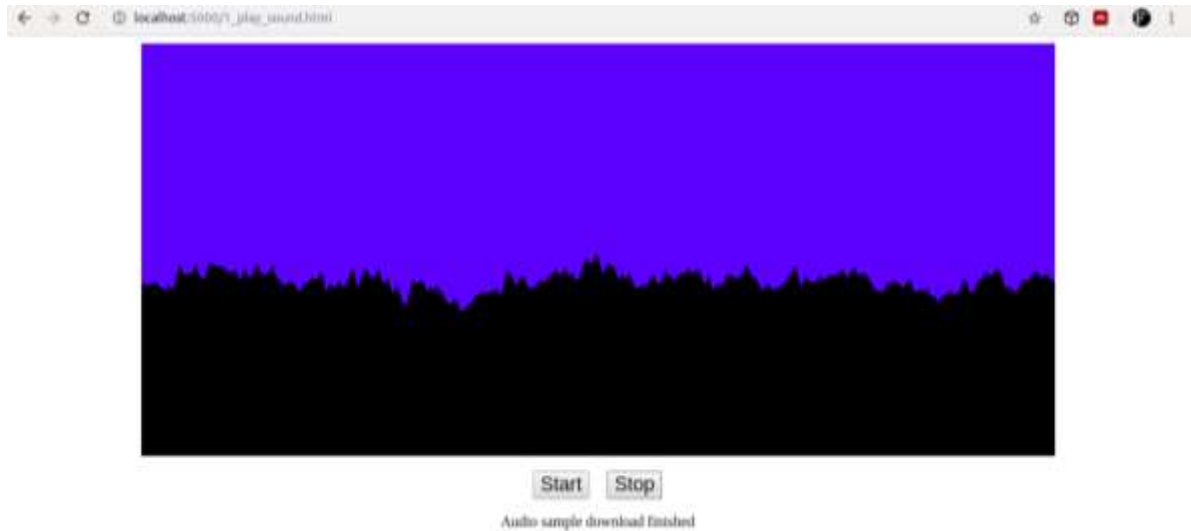
Salah satu elemen multimedia adalah audio. Penyajian dan pengambilan audio melalui aplikasi web dapat dikembangkan menggunakan canvas HTML. Melalui canvas HTML, data audio dapat diperoleh kemudian digunakan sebagai variabel input untuk memanipulasi elemen grafis yang dianimasikan (animated graphic element).

### Topik

- Canvas audio player
- Audio Context
- Audio context analyzer
- Audio visualizing

### Prosedur

1. Simak dan pahami prosedur pembuatannya pada modul di bawah ini
2. Membuat aplikasi web untuk memvisualkan audio yang diambil dari sumber daya lokal. Lakukan pengembangan pada model visualisasi sehingga menghasilkan efek animasi yang berbeda. Contoh audio visualizer adalah seperti gambar berikut:



3. Langkah-langkah dalam membuatnya dapat diikuti pada prosedur di bawah ini :
  - a. Langkah pertama buat canvas untuk menampilkan audio visualizernya dan membuat dua button untuk start dan stop

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width">

    <title>example</title>

    <link rel="stylesheet" href="">

  </head>

  <body onload="startApp()">
    <canvas id="canvas"></canvas>

    <p id="controls">
      <input type="button" id="start_button" value="Start">
      &nbsp; &nbsp; &nbsp;
      <input type="button" id="stop_button" value="Stop">
      <p id="msg"></p>
    </p>

    <!-- ----- -->
```

```

<style>
  #canvas {
    margin-left: auto;
    margin-right: auto;
    display: block;
    background-color: black;
  }
  #controls {
    text-align: center;
  }
  #start_button, #stop_button {
    font-size: 16pt;
  }
  #msg {
    text-align: center;
  }
</style>

<!-- ----- -->

```

- b. Langkah selanjutnya adalah mendefinisikan variabel yang akan digunakan untuk membuat audio visualizernya

```

<script type="text/javascript">
  // Hacks to deal with different function names in different
  browsers

  window.requestAnimationFrame = (function(){
    return window.requestAnimationFrame ||
           window.webkitRequestAnimationFrame ||
           window.mozRequestAnimationFrame ||
           function(callback, element){
             window.setTimeout(callback, 1000 / 60);
           };
  })();

  window.AudioContext = (function(){
    return window.webkitAudioContext || window.AudioContext ||
           window.mozAudioContext;
  })();

  // Global Variables for Audio
  var audioContext;

```



```

var audioBuffer;
var sourceNode;
var analyserNode;
var javascriptNode;
var audioData = null;
var audioPlaying = false;
var sampleSize = 1024; //sampel angka yang di ambil sebelum
data di analisis
var amplitudeArray; // array untuk menyimpan data domainnya
var audioUrl = "Calvin_Harris_My_Way.mp3";//sampel audio yang
akan di ambil untuk di jadikan atau di convert menjadi grafik visual

var canvasWidth = 1000; //mendefinisikan lebar kanvas
var canvasHeight = 450; //mendefinisikan tinggi kanvas
var ctx;

```

- c. Membuat fungsi utama untuk menjalankan audionya, pada fungsi ini akan membuat kode program javascript untuk menjalankan tombol start dan stop lalu memanggil kanvas dan Fungsi ini juga berfungsi untuk memanggil fungsi-fungsi yang lainnya.

```

function startApp() {
    var c=document.getElementById('canvas');
    c.width=canvasWidth;
    c.height=canvasHeight;

    ctx = c.getContext("2d");

    try {
        audioContext = new AudioContext();
    } catch(e) {
        alert('Web Audio API is not supported in this
browser');
    }
    // Ketika tombol start di klik, selesai mengatur node
audio, dan memainkan audio
    // mengumpulkan sampel data untuk di analisa lalu
memperbarui kanvas
    var startBtn=document.getElementById('start_button');
    startBtn.onclick= function(e) {
        javascriptNode

```

```

        setupAudioNodes();

        javascriptNode.onaudioprocess = function () {
            // get the Time Domain data for this sample
            analyserNode.getByteTimeDomainData(amplitudeArray);
            // draw the display if the audio is playing
            if (audioPlaying == true) {
                requestAnimationFrame(drawTimeDomain);
            }
        }
        // Load the Audio the first time through, otherwise
        play it from the buffer
        if(audioData == null) {
            loadSound(audioUrl);
        } else {
            playSound(audioData);
        }
    };
    // Stop the audio playing
    var stopBtn=document.getElementById('stop_button');
    stopBtn.onclick=function(e) {
        e.preventDefault();
        sourceNode.stop(0);
        audioPlaying = false;
    };
};

```

- d. Selanjutnya adalah membuat fungsi `setupAudioNodes`, fungsi ini berguna untuk memanggil audio dari URL via ajax dan menyimpannya secara global pada variabel `audioData`

```

function setupAudioNodes() {
    sourceNode      = audioContext.createBufferSource();
    analyserNode    = audioContext.createAnalyser();
    javascriptNode =
audioContext.createScriptProcessor(sampleSize, 1, 1);
    // Create the array for the data values
    amplitudeArray = new
Uint8Array(analyserNode.frequencyBinCount);
    // Now connect the nodes together
    sourceNode.connect(audioContext.destination);
    sourceNode.connect(analyserNode);
    analyserNode.connect(javascriptNode);

```

```

        javascriptNode.connect(audioContext.destination);
    }

```

- e. Langkah selanjutnya adalah membuat fungsi loopSound & playSound, fungsi ini berguna untuk memainkan audio sampai tombol stop di klik.

```

function loadSound(url) {
    document.getElementById('msg').textContent = "Loading
audio...";
    var request = new XMLHttpRequest();
    request.open('GET', url, true);
    request.responseType = 'arraybuffer';
    // When loaded, decode the data and play the sound
    request.onload = function () {
        audioContext.decodeAudioData(request.response, function
(buffer) {
            document.getElementById('msg').textContent = "Audio
sample download finished";
            audioData = buffer;
            playSound(audioData);
        }, onError);
    }
    request.send();
}
function playSound(buffer) {
    sourceNode.buffer = buffer;
    sourceNode.start(0);    // Play the sound now
    sourceNode.loop = true;
    audioPlaying = true;
}

```

- f. Setelah itu membuat tampilan ketika terdapat error bisa menggunakan fungsi onError

```

function onError(e) {
    console.log(e);
}

```

- g. Langkah terakhir adalah membuat fungsi untuk membuat tampilan grafik dari audionya atau audio visualizernya, dimana grafik ini akan bergerak sesuai dengan nada yang ada pada audio yang di panggil.

```

function drawTimeDomain() {

```

```

        clearCanvas();
        for (var i = 0; i < amplitudeArray.length; i++) {
            var value = amplitudeArray[i] / 300;
            var y = canvasHeight - (canvasHeight * value) - 1;
            var x = i*canvasWidth/amplitudeArray.length;
            var clr=['red','blue','yellow'];
            var color=Math.floor(Math.random()*10);
            ctx.fillStyle = 'blue';
            ctx.fillRect(x, 0, 2, y);
        }
    }
    function clearCanvas() {
        ctx.clearRect(0, 0, canvasWidth, canvasHeight);
    }
</script>
</body>
</html>

```

Kode lengkapnya ada di bawah ini :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width">

    <title>example</title>

    <link rel="stylesheet" href="">

  </head>

  <body onload="startApp()">
    <canvas id="canvas"></canvas>

    <p id="controls">
      <input type="button" id="start_button" value="Start">
      &nbsp; &nbsp; &nbsp;
    </p>
  </body>
</html>

```

```

    <input type="button" id="stop_button" value="Stop">
    <p id="msg"></p>
</p>

<!-- ----- -->

<style>
    #canvas {
        margin-left: auto;
        margin-right: auto;
        display: block;
        background-color: black;
    }
    #controls {
        text-align: center;
    }
    #start_button, #stop_button {
        font-size: 16pt;
    }
    #msg {
        text-align: center;
    }
</style>

<!-- ----- -->

<script type="text/javascript">
    // Hacks to deal with different function names in different
browsers
    window.requestAnimationFrame = (function(){
        return window.requestAnimationFrame ||
            window.webkitRequestAnimationFrame ||
            window.mozRequestAnimationFrame ||
            function(callback, element){
                window.setTimeout(callback, 1000 / 60);
            };
    })();
    window.AudioContext = (function(){
        return window.webkitAudioContext || window.AudioContext ||
window.mozAudioContext;
    })();
    // Global Variables for Audio

```

```

var audioContext;
var audioBuffer;
var sourceNode;
var analyserNode;
var javascriptNode;
var audioData = null;
var audioPlaying = false;
var sampleSize = 1024;
var amplitudeArray;
var audioUrl = "Calvin_Harris_My_Way.mp3";
// Global Variables for the Graphics
var canvasWidth = 1000;
var canvasHeight = 450;
var ctx;
function startApp() {
    var c=document.getElementById('canvas');
    c.width=canvasWidth;
    c.height=canvasHeight;

    ctx = c.getContext("2d");
    try {
        audioContext = new AudioContext();
    } catch(e) {
        alert('Web Audio API is not supported in this
browser');
    }

    // gather samples for the analysis, update the canvas
    var startBtn=document.getElementById('start_button');
    startBtn.onclick= function(e) {
        e.preventDefault();
        setupAudioNodes();

javascriptNode.onaudioprocess = function () {

analyserNode.getByteTimeDomainData(amplitudeArray);
        // draw the display if the audio is playing
        if (audioPlaying == true) {
            requestAnimFrame(drawTimeDomain);
        }
    }

        if(audioData == null) {
            loadSound(audioUrl);

```

```

        } else {
            playSound(audioData);
        }
    };
    // Stop the audio playing
    var stopBtn=document.getElementById('stop_button');
    stopBtn.onclick=function(e) {
        e.preventDefault();
        sourceNode.stop(0);
        audioPlaying = false;
    };
};
function setupAudioNodes() {
    sourceNode      = audioContext.createBufferSource();
    analyserNode    = audioContext.createAnalyser();
    javascriptNode =
audioContext.createScriptProcessor(sampleSize, 1, 1);
        amplitudeArray = new
Uint8Array(analyserNode.frequencyBinCount);
    // Now connect the nodes together
    sourceNode.connect(audioContext.destination);
    sourceNode.connect(analyserNode);
    analyserNode.connect(javascriptNode);
    javascriptNode.connect(audioContext.destination);
}

        function loadSound(url) {
            document.getElementById('msg').textContent = "Loading
audio...";
            var request = new XMLHttpRequest();
            request.open('GET', url, true);
            request.responseType = 'arraybuffer';

            request.onload = function () {
                audioContext.decodeAudioData(request.response, function
(buffer) {
                    document.getElementById('msg').textContent = "Audio
sample download finished";
                    audioData = buffer;
                    playSound(audioData);
                }, onError);
            }
            request.send();
        }
}

```

```

        function playSound(buffer) {
            sourceNode.buffer = buffer;
            sourceNode.start(0);
            sourceNode.loop = true;
            audioPlaying = true;
        }
        function onError(e) {
            console.log(e);
        }
        function drawTimeDomain() {
            clearCanvas();
            for (var i = 0; i < amplitudeArray.length; i++) {
                var value = amplitudeArray[i] / 300;
                var y = canvasHeight - (canvasHeight * value) - 1;
                var x = i*canvasWidth/amplitudeArray.length;
                var clr=['red','blue','yellow'];
                var color=Math.floor(Math.random()*10);
                ctx.fillStyle = 'blue';
                ctx.fillRect(x, 0, 2, y);
            }
        }
        function clearCanvas() {
            ctx.clearRect(0, 0, canvasWidth, canvasHeight);
        }
    </script>
</body>
</html>

```

4. Simpan file dengan praktikum \_7\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github.



# Praktikum 8

## Online Audio Recording

---

### Ringkasan

Perekaman audio pada aplikasi web dapat dilakukan menggunakan canvas HTML. Perekaman audio adalah mengambil data audio melalui perangkat microphone. Data audio yang diperoleh selanjutnya dapat digunakan lebih lanjut misalnya: disimpan pada perangkat atau diunggah ke server. Menggunakan kemampuan perekaman pada canvas HTML salah satunya dapat digunakan untuk membuat percakapan audio secara online.

### Topik

- Device media permission
- Canvas media recorder
- Audio data conversion

### Prosedur

1. Simak dan pahami tutorial yang ada pada modul ini
2. Membuat aplikasi web yang dapat merekam dan memutar kembali rekaman tersebut. Ilustrasi aplikasi adalah sebagai berikut:



3. Langkah-langkah untuk membuatnya dapat diikuti pada prosedur di bawah ini :
  - a. Langkah pertama adalah membuat tombol REC dan STOP untuk kode programnya dapat dilihat di bawah ini :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width">

    <title>example</title>

    <link rel="stylesheet" href="">

  </head>

  <body>

    <button id="rec">REC</button>
    <button id="stop">STOP</button>
```

- b. Langkah selanjutnya adalah mendefinisikan variabel untuk merekam suaranya

```
<script>
  if (navigator.mediaDevices) {
    console.log('getUserMedia supported.');
```

```
    var constraints = { audio: true, video: false };
    var chunks = [];
```

```
navigator.mediaDevices.getUserMedia(constraints)
.then(function(stream) {

    var mediaRecorder = new MediaRecorder(stream);

    var record=document.getElementById('rec');
    var stop=document.getElementById('stop');
```

var chunks = [] berfungsi untuk menyimpan audio atau suara hasil dari rekaman dalam bentuk array.

var constraints = berfungsi untuk mendefinisikan bahwa yang dapat direkam hanya yang berformat audio bukan video

c. Membuat fungsi ketika tombol record di klik

```
record.onclick = function() {
    mediaRecorder.start();
    console.log(mediaRecorder.state);
    console.log("recorder started");
    record.style.background = "red";
    record.style.color = "black";
}
```

d. Membuat fungsi ketika tombol stop di klik

```
stop.onclick = function() {
    mediaRecorder.stop();
    console.log(mediaRecorder.state);
    console.log("recorder stopped");
    record.style.background = "";
    record.style.color = "";
}
```

e. Langkah terakhir adalah membuat fungsi ketika tombol stop di klik makan audio yang sudah terekam akan di olah pada fungsi mediaRecorder.onStop dan akan dapat di putar kembali hasil rekaman suara dari fungsi record.

```
mediaRecorder.onstop = function(e) {
    console.log("data available after MediaRecorder.stop() called.");
```

```

var clipName = prompt('Enter a name for your sound clip');

var clipContainer = document.createElement('article');
var clipLabel = document.createElement('p');
var audio = document.createElement('audio');
var deleteButton = document.createElement('button');

clipContainer.classList.add('clip');
audio.setAttribute('controls', '');
deleteButton.innerHTML = "Delete";
clipLabel.innerHTML = clipName;

clipContainer.appendChild(audio);
clipContainer.appendChild(clipLabel);
clipContainer.appendChild(deleteButton);
document.body.appendChild(clipContainer);

audio.controls = true;
var blob = new Blob(chunks, { 'type' : 'audio/ogg; codecs=opus' });
chunks = [];
var audioURL = URL.createObjectURL(blob);
audio.src = audioURL;
console.log("recorder stopped");

deleteButton.onclick = function(e) {
    evtTgt = e.target;
    evtTgt.parentNode.parentNode.removeChild(evtTgt.parentNode);
}

mediaRecorder.ondataavailable = function(e) {
    chunks.push(e.data);
}
})
.catch(function(err) {
    console.log('The following error occurred: ' + err);
})
}

</script>

</body>
</html>

```

Kode lengkapnya ada di bawah ini :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width">

    <title>example</title>

    <link rel="stylesheet" href="">

  </head>

  <body>

<button id="rec">REC</button>
<button id="stop">STOP</button>

<script>
  if (navigator.mediaDevices) {
    console.log('getUserMedia supported.');


var constraints = { audio: true,video:false };



var chunks = [];



navigator.mediaDevices.getUserMedia(constraints)
  .then(function(stream) {



var mediaRecorder = new MediaRecorder(stream);



var record=document.getElementById('rec');



var stop=document.getElementById('stop');



record.onclick = function() {



mediaRecorder.start();



console.log(mediaRecorder.state);



console.log("recorder started");


```

```

    record.style.background = "red";
    record.style.color = "black";
}

stop.onclick = function() {
    mediaRecorder.stop();
    console.log(mediaRecorder.state);
    console.log("recorder stopped");
    record.style.background = "";
    record.style.color = "";
}

mediaRecorder.onstop = function(e) {
    console.log("data available after MediaRecorder.stop() called.");

    var clipName = prompt('Enter a name for your sound clip');

    var clipContainer = document.createElement('article');
    var clipLabel = document.createElement('p');
    var audio = document.createElement('audio');
    var deleteButton = document.createElement('button');

    clipContainer.classList.add('clip');
    audio.setAttribute('controls', '');
    deleteButton.innerHTML = "Delete";
    clipLabel.innerHTML = clipName;

    clipContainer.appendChild(audio);
    clipContainer.appendChild(clipLabel);
    clipContainer.appendChild(deleteButton);
    document.body.appendChild(clipContainer);

    audio.controls = true;
    var blob = new Blob(chunks, { 'type' : 'audio/ogg; codecs=opus' });
    chunks = [];
    var audioURL = URL.createObjectURL(blob);
    audio.src = audioURL;
    console.log("recorder stopped");

    deleteButton.onclick = function(e) {
        evtTgt = e.target;
        evtTgt.parentNode.removeChild(evtTgt.parentNode);
    }
}

```

```
    }

    mediaRecorder.ondataavailable = function(e) {
        chunks.push(e.data);
    }
})
.catch(function(err) {
    console.log('The following error occurred: ' + err);
})
}

</script>

</body>
</html>
```

4. Simpan file dengan praktikum\_8\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github

# Praktikum 9

## Image Manipulating

---

### Ringkasan

Menggunakan fungsi `drawImage()` pada canvas HTML, objek gambar dapat dimanipulasi ukuran maupun tampilannya. Fungsi tersebut dapat digunakan untuk membuat alat editing gambar, khususnya untuk keperluan pemotongan (cropping) gambar. Praktikum ini mengasah kemampuan mahasiswa untuk menggunakan fungsi tersebut dikombinasikan dengan materi sebelumnya (interaktivitas) untuk membuat alat pemotong gambar sederhana berbasis web.

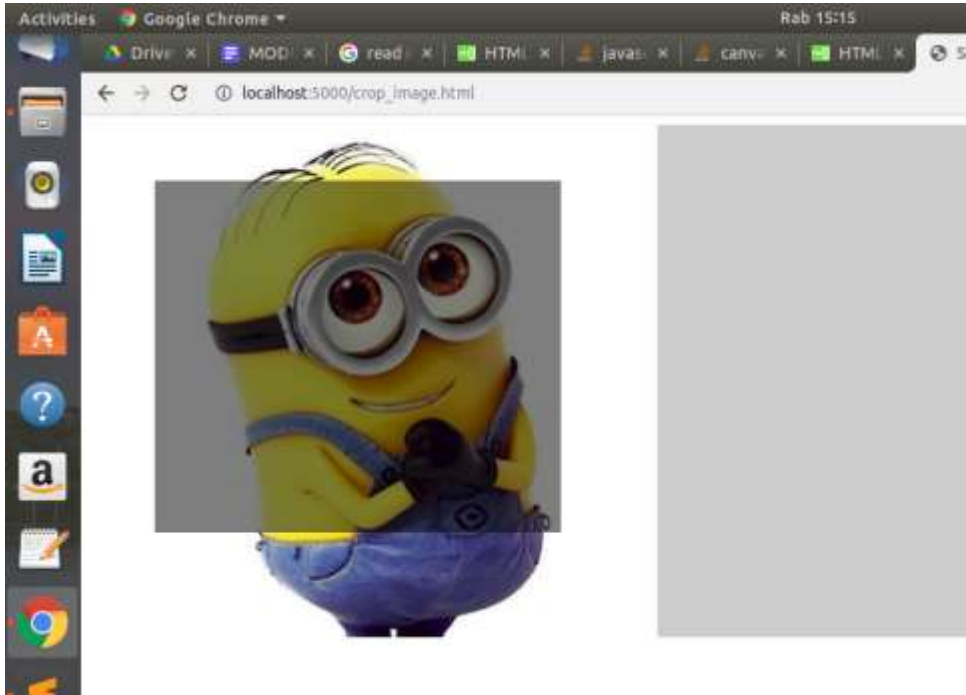
### Topik

- `drawImage()`
- Canvas Image Class
- Interactivity (lanjut)



## Prosedur

1. Simak dan pahami tutorial pada modul ini
2. Membuat aplikasi image editor sederhana berbasis web dengan ilustrasi aplikasi adalah seperti gambar berikut:



Ketika kursor ditekan maka objek segi empat dengan warna transparansi hitam akan dibuat. Setelah mouse dilepas maka image di potong sesuai dengan area persegi yang dibuat.

3. Langkah-langkah untuk membuatnya bisa dengan mengikuti prosedur di bawah ini :
  - a. Langkah pertama adalah mendefinisikan variabel-variabel yang akan digunakan untuk membuat manipulasi image

```
<!doctype html>
<html>
<head>
  <title>Show Image</title>
  <meta charset='utf-8'>
  </script>
</head>
<style type="text/css">
  canvas{
```

```

        background-color: #ccc;
    }
</style>
<body>

<script>
var img;
var rect;
var croppedImage;
var btn;
var avatar;
var cropping=false;

```

- b. Selanjutnya membuat program utama untuk melakukan crop image kode program lengkapnya ada di bawah ini :

```

function startApp()
{

    btn=document.getElementById('btn');
    avatar=new Image();
    avatar.src='minion.jpg';
    myApp.start();
    img=new image(0,0);
    rect=new rectComponent(0,0,1,1);

    drawRect();
}

```

variabel avatar.src berfungsi untuk mengambil sampel gambar yang akan dimasukkan ke dalam kanvas

- c. Setelah itu membuat variabel untuk membuat kanvas atau lembar kerja untuk memanggil objek

```

var myApp = {
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 800;
        this.canvas.height = 450;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
        this.interval = setInterval(apploop, 20);
    }
}

```

```

        //window.requestAnimationFrame(gameLoop);
        //document.body.addEventListener('mousedown',addImage);
    },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
    }
}

```

- d. Langkah selanjutnya adalah membuat fungsi untuk memanggil gambar ke kanvas untuk kode programnya ada di bawah ini

```

function image(x,y)
{
    this.x=x;
    this.y=y;
    this.width=400;
    this.height=400;
    var ctx=myApp.context;
    this.startCropX=x;
    this.startCropY=y;
    this.cropWidth=200;
    this.cropHeight=200;
    this.cropping=false;
    this.update=function()
    {
        if(!this.cropping)
            ctx.drawImage(avatar,this.x,this.y);
        else

        //ctx.drawImage(avatar,this.startCropX,this.startCropY,this.croptWidth,
this.cropHeight,this.x,this.y,this.width,this.height);

        ctx.drawImage(avatar,this.startCropX,this.startCropY,this.cropWidth,t
his.cropHeight,this.x,this.y,this.width,this.height);
    }
}
function apploop()
{
    myApp.clear();
    img.update();
}

```

```
rect.update();
}
```

- e. Setelah itu membuat kotak yang nantinya digunakan untuk menyimpan gambar yang sudah di crop atau di potong.

```
function rectComponent(x,y,w,h)
{
    this.x=x;
    this.y=y;
    this.width=w;
    this.height=h;
    var contex=myApp.context;
    this.update=function()
    {
        contex.beginPath();
        contex.fillStyle = "rgba(0, 0, 0, 0.5)";
        contex.fillRect(this.x, this.y, this.width, this.height);
    }
}

function editingMode(mode)
{
    if(img.cropping) img.cropping=false; else img.cropping=true;
}
```

- f. Langkah selanjutnya adalah membuat fungsi crop gambar kode lengkapnya ada di bawah ini :

```
function cropImage()
{
    /*
    var ctx=myApp.context;
    var cropStartX=rect.x,
    cropStartY=rect.y,
    cropWidth=100,
    cropHeight=100,
    imageX=500,
    imageY=500,
    imageWidth=cropWidth,
    imageHeight=cropHeight;
    var imageSource=new Image();
    imageSource.src='minion.jpg';
```

```

        ctx.drawImage(imageSource, cropStartX, cropStartY, cropWidth, cropHeight,
imageX, imageY, imageWidth, imageHeight);
    */
    img.x=rect.x;
    img.y=rect.y;
    img.width=rect.width;
    img.height=rect.height;
    img.startCropX=rect.x;
    img.startCropY=rect.y;
    img.cropWidth=rect.width;
    img.cropHeight=rect.height;
    console.log(img);
}

function drawRect()
{
    var x,y,rw,rh;
    document.body.addEventListener('mousedown',function(e)
    {
        x=e.offsetX;
        y=e.offsetY;
        rect=new rectComponent(x,y,1,1);
        document.body.addEventListener('mousemove',mouseMove);
        document.body.addEventListener('mouseup',mouseUp);
    });
    function mouseMove(e)
    {
        rw=e.offsetX-rect.x;
        rh=e.offsetY-rect.y;
        rect.width=rw;
        rect.height=rh;
    }
    function mouseUp(e)
    {
        document.body.removeEventListener('mousemove',mouseMove);
        document.body.removeEventListener('mouseup',mouseUp);
        img.cropping=true;
        cropImage();
    }
}

```

- g. Langkah terakhir adalah menyimpan hasil dari fungsi crop image ke fungsi program utama.

```
function addImage()  
{  
    img.update();  
}  
  
startApp();  
  
</script>  
</body>  
  
</html>
```

Kode program lengkapnya ada di bawah ini :

```
<!doctype html>  
<html>  
<head>  
    <title>Show Image</title>  
    <meta charset='utf-8'>  
    </script>  
</head>  
<style type="text/css">  
    canvas{  
        background-color: #ccc;  
    }  
</style>  
<body>  
  
<script>  
var img;  
var rect;  
var croppedImage;  
var btn;  
var avatar;  
var cropping=false;  
function startApp()  
{
```

```

    btn=document.getElementById('btn');
    avatar=new Image();
    avatar.src='minion.jpg';
    myApp.start();
    img=new image(0,0);
    rect=new rectComponent(0,0,1,1);

    drawRect();
}
var myApp = {
  canvas : document.createElement("canvas"),
  start : function() {
    this.canvas.width = 800;
    this.canvas.height = 450;
    this.context = this.canvas.getContext("2d");
    document.body.insertBefore(this.canvas,
document.body.childNodes[0]);
    this.interval = setInterval(apploop, 20);
    //window.requestAnimationFrame(gameLoop);
    //document.body.addEventListener('mousedown',addImage);
  },
  clear : function() {
    this.context.clearRect(0, 0, this.canvas.width,
this.canvas.height);
  }
}

function image(x,y)
{
  this.x=x;
  this.y=y;
  this.width=400;
  this.height=400;
  var ctx=myApp.context;
  this.startCropX=x;
  this.startCropY=y;
  this.cropWidth=200;
  this.cropHeight=200;
  this.cropping=false;
  this.update=function()
  {
    if(!this.cropping)

```

```

        ctx.drawImage(avatar, this.x, this.y);
    else

        //ctx.drawImage(avatar, this.startCropX, this.startCropY, this.croptWidth, this.cropHeight, this.x, this.y, this.width, this.height);

        ctx.drawImage(avatar, this.startCropX, this.startCropY, this.cropWidth, this.cropHeight, this.x, this.y, this.width, this.height);
    }
}

function apploop()
{
    myApp.clear();
    img.update();
    rect.update();
}

function rectComponent(x,y,w,h)
{
    this.x=x;
    this.y=y;
    this.width=w;
    this.height=h;
    var contex=myApp.context;
    this.update=function()
    {
        contex.beginPath();
        contex.fillStyle = "rgba(0, 0, 0, 0.5)";
        contex.fillRect(this.x, this.y, this.width, this.height);
    }
}

function editingMode(mode)
{
    if(img.cropping) img.cropping=false; else img.cropping=true;
}

function cropImage()
{
    /*
    var ctx=myApp.context;
    var cropStartX=rect.x,

```



```

        cropStartY=rect.y,
        cropWidth=100,
        cropHeight=100,
        imageX=500,
        imageY=500,
        imageWidth=cropWidth,
        imageHeight=cropHeight;
        var imageSource=new Image();
        imageSource.src='minion.jpg';
        ctx.drawImage(imageSource,cropStartX,cropStartY,cropWidth,cropHeight,
imageX,imageY,imageWidth,imageHeight);
    */
    img.x=rect.x;
    img.y=rect.y;
    img.width=rect.width;
    img.height=rect.height;
    img.startCropX=rect.x;
    img.startCropY=rect.y;
    img.cropWidth=rect.width;
    img.cropHeight=rect.height;
    console.log(img);
}

function drawRect()
{
    var x,y,rw,rh;
    document.body.addEventListener('mousedown',function(e)
    {
        x=e.offsetX;
        y=e.offsetY;
        rect=new rectComponent(x,y,1,1);
        document.body.addEventListener('mousemove',mouseMove);
        document.body.addEventListener('mouseup',mouseUp);
    });
    function mouseMove(e)
    {
        rw=e.offsetX-rect.x;
        rh=e.offsetY-rect.y;
        rect.width=rw;
        rect.height=rh;
    }
    function mouseUp(e)

```

```
{
    document.body.removeEventListener('mousemove',mouseMove);
    document.body.removeEventListener('mouseup',mouseUp);
    img.cropping=true;
    cropImage();
}
}
function addImage()
{
    img.update();
}

startApp();

</script>
</body>

</html>
```

4. Simpan file dengan praktikum\_9\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github

# Praktikum 10

## Online Photo Capturing

---

### Ringkasan

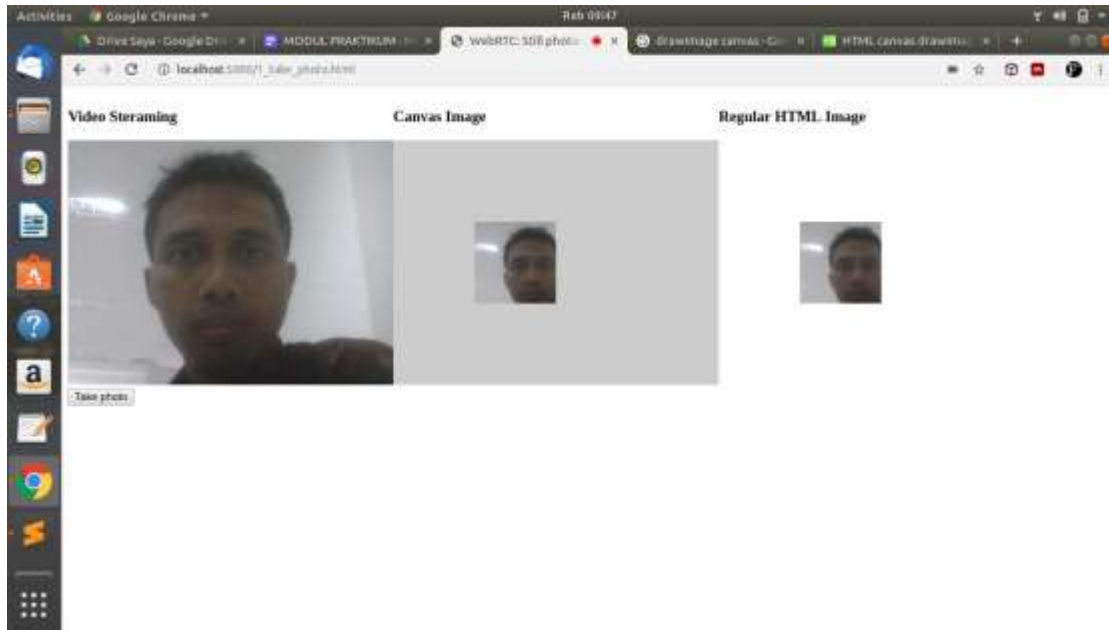
Perangkat yang dapat diakses melalui aplikasi web salah satunya adalah kamera. Kamera tersebut dapat dimanfaatkan untuk mengambil salah satu frame sebagai foto. Data foto adalah dalam bentuk DATA\_URL dengan format base64. Data tersebut selanjutnya dapat dimanfaatkan untuk dikirim ke server atau diunduh dalam format PNG.

### Topik

- Event listener untuk video capturing
- Konversi canvas ke file image
- Fungsi draw image pada canvas HTML

### Prosedur

1. Simak dan pahami tutorial pada modul ini
2. Membuat aplikasi untuk mengambil foto melalui halaman web memanfaatkan web camera. Ilustrasi aplikasi adalah sebagai berikut:



3. Langkah-langkah dalam membuatnya dapat diikuti pada prosedur di bawah ini :
  - a. Langkah pertama adalah membuat kanvas, untuk mengambil foto (gambar) dan tombol untuk take picture

```
<!doctype html>
<html>
<head>
  <title>WebRTC: Still photo capture demo</title>
  <meta charset='utf-8'>
  <link rel="stylesheet" href="main.css" type="text/css" media="all">
  <style type="text/css">
    #canvas{
      background-color: #ccc;
    }
    .camera, .canvas-container, .image-container{
      float: left;
    }
  </style>
</head>
<body onload="startup()">

  <div class="camera">
    <h3>Video Streaming</h3>
    <video id="video">Video stream not available.</video>
    <div>
      <button id="startbutton">Take photo</button>
    </div>
  </div>
```

```

</div>
<div class="canvas-container">
  <h3>Canvas Image</h3>
  <canvas id="canvas"></canvas>
</div>
<div class="image-container">
  <h3>Regular HTML Image</h3>
  <div class="output">
    <img id="photo" alt="The screen capture will appear in this box.">
  </div>
</div>!doctype html>
<html>
<head>
  <title>WebRTC: Still photo capture demo</title>
  <meta charset='utf-8'>
  <link rel="stylesheet" href="main.css" type="text/css" media="all">
<style type="text/css">
  #canvas{
    background-color: #ccc;
  }
  .camera, .canvas-container, .image-container{
    float: left;
  }
</style>
</head>
<body onload="startup()">

  <div class="camera">
    <h3>Video Steraming</h3>
    <video id="video">Video stream not available.</video>
    <div>
      <button id="startbutton">Take photo</button>
    </div>
  </div>
  <div class="canvas-container">
    <h3>Canvas Image</h3>
    <canvas id="canvas"></canvas>
  </div>
  <div class="image-container">
    <h3>Regular HTML Image</h3>
    <div class="output">
      <img id="photo" alt="The screen capture will appear in this box.">
    </div>
  </div>

```

```
</div>
```

- b. Langkah selanjutnya adalah mendefinisikan variabel-variabel yang akan digunakan untuk mengambil gambar, kode programnya ada di bawah ini :

```
<script type="text/javascript">
    var width = 400;    //kita mengatur skala fotonya dengan variabel ini
    var height = 0;

    var streaming = false; //Variabel yang menunjukkan apakah kita sedang
    streaming atau tidak

    // variabel-variabel ini akan di set pada startup() function.

    var video = null;
    var canvas = null;
    var photo = null;
    var startbutton = null;
```

- c. Selanjutnya adalah membuat fungsi startup yang digunakan untuk merekam semua yang tertangkap oleh kamera.

```
function startup() {
    video = document.getElementById('video');
    canvas = document.getElementById('canvas');
    photo = document.getElementById('photo');
    startbutton = document.getElementById('startbutton');

    navigator.mediaDevices.getUserMedia({video: true, audio: false})
    .then(function(stream) {
        video.srcObject = stream;
        video.play();
    })
    .catch(function(err) {
        console.log("An error occurred: " + err);
    });

    video.addEventListener('canplay', function(ev){
        if (!streaming) {
            height = video.videoHeight / (video.videoWidth/width);
```

```

    if (isNaN(height)) {
        height = width / (4/3);
    }

    video.setAttribute('width', width);
    video.setAttribute('height', height);
    canvas.setAttribute('width', width);
    canvas.setAttribute('height', height);
    streaming = true;
}
}, false);
startbutton.addEventListener('click', function(ev){
    takepicture();
    ev.preventDefault();
}, false);
clearphoto();
}
function clearphoto() {
    var context = canvas.getContext('2d');
    context.fillStyle = "#AAA";
    context.fillRect(0, 0, canvas.width, canvas.height);
    var data = canvas.toDataURL('image/png');
    photo.setAttribute('src', data);
}

```

d. Langkah terakhir adalah membuat fungsi mengambil gambar atau take picture

```

function takepicture() {
    var context = canvas.getContext('2d');
    if (width && height) {
        canvas.width = width;
        canvas.height = height;
        var cropStartX=0,
        cropStartY=0,
        cropWidth=400,
        cropHeight=400,
        imageX=100,
        imageY=100,
        imageWidth=100,

```

```

        imageHeight=100;
        context.drawImage(video, cropStartX,
cropStartY,cropWidth,cropHeight,imageX,imageY, imageWidth, imageHeight);
        var data = canvas.toDataURL('image/png');
        photo.setAttribute('src', data);
    } else {
        clearphoto();
    }
}
</script>
</body>
</html>

```

Kode lengkapnya ada di bawah ini :

```

<!doctype html>
<html>
<head>
    <title>WebRTC: Still photo capture demo</title>
    <meta charset='utf-8'>
    <link rel="stylesheet" href="main.css" type="text/css" media="all">
    <style type="text/css">
        #canvas{
            background-color: #ccc;
        }
        .camera, .canvas-container, .image-container{
            float: left;
        }
    </style>
</head>
<body onload="startup()">

    <div class="camera">
        <h3>Video Streaming</h3>
        <video id="video">Video stream not available.</video>
        <div>
            <button id="startbutton">Take photo</button>
        </div>
    </div>
    <div class="canvas-container">

```



```

    <h3>Canvas Image</h3>
    <canvas id="canvas"></canvas>
</div>
<div class="image-container">
    <h3>Regular HTML Image</h3>
    <div class="output">
        <img id="photo" alt="The screen capture will appear in this box.">
    </div>
</div>
<script type="text/javascript">
    var width = 400;
    var height = 0;

    var streaming = false;

    var video = null;
    var canvas = null;
    var photo = null;
    var startbutton = null;

    function startup() {
        video = document.getElementById('video');
        canvas = document.getElementById('canvas');
        photo = document.getElementById('photo');
        startbutton = document.getElementById('startbutton');

        navigator.mediaDevices.getUserMedia({video: true, audio: false})
        .then(function(stream) {
            video.srcObject = stream;
            video.play();
        })
        .catch(function(err) {
            console.log("An error occurred: " + err);
        });

        video.addEventListener('canplay', function(ev){
            if (!streaming) {
                height = video.videoHeight / (video.videoWidth/width);

```

```

    if (isNaN(height)) {
        height = width / (4/3);
    }

    video.setAttribute('width', width);
    video.setAttribute('height', height);
    canvas.setAttribute('width', width);
    canvas.setAttribute('height', height);
    streaming = true;
}
}, false);
startbutton.addEventListener('click', function(ev){
    takepicture();
    ev.preventDefault();
}, false);
clearphoto();
}

function clearphoto() {
    var context = canvas.getContext('2d');
    context.fillStyle = "#AAA";
    context.fillRect(0, 0, canvas.width, canvas.height);
    var data = canvas.toDataURL('image/png');
    photo.setAttribute('src', data);
}

function takepicture() {
    var context = canvas.getContext('2d');
    if (width && height) {
        canvas.width = width;
        canvas.height = height;
        var cropStartX=0,
            cropStartY=0,
            cropWidth=400,
            cropHeight=400,
            imageX=100,
            imageY=100,
            imageWidth=100,
            imageHeight=100;
        context.drawImage(video, cropStartX,
cropStartY,cropWidth,cropHeight,imageX,imageY, imageWidth, imageHeight);
        var data = canvas.toDataURL('image/png');

```

```
        photo.setAttribute('src', data);  
    } else {  
        clearphoto();  
    }  
}  
</script>  
</body>  
</html>
```

4. Simpan file dengan praktikum\_10\_[NIM].html
5. Unggah hasil pada repository github masing-masing dan tautkan url pada halaman utama github