

# Praktikum WebRTC

## Percobaan 1 : GetUserMedia

Langkah percobaan

1. Buatlah folder js
2. Buatlah file main.js pada folder js tersebut

main.js

```
'use strict';

var errorElement = document.querySelector('#errorMsg');
var video = document.querySelector('video');

// Put variables in global scope to make them available to the browser console.
var constraints = window.constraints = {
  audio: false,
  video: true
};

function handleSuccess(stream) {
  var videoTracks = stream.getVideoTracks();
  console.log('Got stream with constraints:', constraints);
  console.log('Using video device: ' + videoTracks[0].label);
  stream.oninactive = function() {
    console.log('Stream inactive');
  };
  window.stream = stream; // make variable available to browser console
  video.srcObject = stream;
}

function handleError(error) {
  if (error.name === 'ConstraintNotSatisfiedError') {
    errorMsg('The resolution ' + constraints.video.width.exact + 'x' +
      constraints.video.height.exact + ' px is not supported by your device.');
```

3. Buatlah file dengan nama percobaan1.html

percobaan1.html

```
<html>
<head>
  <base target="_blank">
  <title>getUserMedia</title>
</head>
<body>
  <div id="container">
    <div class="highlight">
      <p>Get Media</p>
    </div>
    <h1>Percobaan 1 : Menampilkan Webcam Dengan HTML5</h1>
    <video id="gum-local" autoplay playsinline></video>
    <div id="errorMsg"></div>
  </div>
  <script src="js/main.js"></script>
</body>
</html>
```

4. Akses aplikasi yang telah anda kerjakan dengan menggunakan browser chrome (dengan web cam)
5. Amati hasil yang terjadi

## Percobaan 2 : Take Snapshot

Langkah percobaan

1. Buatlah file main2.js pada folder js yang telah dibuat pada percobaan 1
2. Buatlah file dengan nama percobaan2.html
3. Akses aplikasi yang telah anda kerjakan dengan menggunakan browser chrome (dengan web cam)
4. Amati hasil yang terjadi

## main2.js

```
'use strict';

// Put variables in global scope to make them available to the browser console.
var video = document.querySelector('video');
var canvas = window.canvas = document.querySelector('canvas');
canvas.width = 480;
canvas.height = 360;

var button = document.querySelector('button');
button.onclick = function() {
  canvas.width = video.videoWidth;
  canvas.height = video.videoHeight;
  canvas.getContext('2d').
    drawImage(video, 0, 0, canvas.width, canvas.height);
};

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

function handleSuccess(stream) {
  window.stream = stream; // make stream available to browser console
  video.srcObject = stream;
}

function handleError(error) {
  console.log('navigator.getUserMedia error: ', error);
}

navigator.mediaDevices.getUserMedia(constraints).
  then(handleSuccess).catch(handleError);
```

## Percobaan2.html

```
<html>
<head>
  <base target="_blank">
  <title>getUserMedia to canvas</title>
</head>
<body>
  <div id="container">
    <div class="highlight">
      <p>Get Media</p>
    </div>
    <h1>Percobaan 2 : Take Snapshot</h1>
    <video autoplay></video>
    <button>Take snapshot</button>
    <canvas></canvas>
  </div>
  <script src="js/main2.js"></script>
</body>
</html>
```

## Percobaan 3 : Add Filter

Langkah percobaan

1. Buatlah file main3.js pada folder js yang telah dibuat pada percobaan 1
2. Buatlah file dengan nama percobaan3.html
3. Akses aplikasi yang telah anda kerjakan dengan menggunakan browser chrome (dengan web cam)
4. Amati hasil yang terjadi

Main3.js

```
'use strict';

var snapshotButton = document.querySelector('button#snapshot');
var filterSelect = document.querySelector('select#filter');

// Put variables in global scope to make them available to the browser console.
var video = window.video = document.querySelector('video');
var canvas = window.canvas = document.querySelector('canvas');
canvas.width = 480;
canvas.height = 360;

snapshotButton.onclick = function() {
  canvas.className = filterSelect.value;
  canvas.getContext('2d').drawImage(video, 0, 0, canvas.width, canvas.height);
};
filterSelect.onchange = function() {
  video.className = filterSelect.value;
};
var constraints = {
  audio: false,
  video: true
};
function handleSuccess(stream) {
  window.stream = stream; // make stream available to browser console
  video.srcObject = stream;
}
function handleError(error) {
  console.log('navigator.getUserMedia error: ', error);
}

navigator.mediaDevices.getUserMedia(constraints)
  .then(handleSuccess).catch(handleError);
```

## Percobaan3.html

```
<html>
<head>
  <base target="_blank">
  <title>getUserMedia + CSS filters</title>
  <style>
    .none {
      -webkit-filter: none;
      filter: none;
    }
    .blur {
      -webkit-filter: blur(3px);
      filter: blur(3px);
    }
    .grayscale {
      -webkit-filter: grayscale(1);
      filter: grayscale(1);
    }
    .invert {
      -webkit-filter: invert(1);
      filter: invert(1);
    }
    .sepia {
      -webkit-filter: sepia(1);
      filter: sepia(1);
    }
    button#snapshot {
      margin: 0 10px 25px 0;
      width: 110px;
    }
    video {
      object-fit: cover;
    }
  </style>
</head>
<body>
  <div id="container">
    <h1>Percobaan 3 : Filter</h1>
    <video autoplay></video>
    <label for="select">Filter: </label>
    <select id="filter">
      <option value="none">None</option>
      <option value="blur">Blur</option>
      <option value="grayscale">Grayscale</option>
      <option value="invert">Invert</option>
      <option value="sepia">Sepia</option>
    </select>
    <button id="snapshot">Take snapshot</button>
    <canvas></canvas>
  </div>
  <script src="js/main3.js"></script>
</body>
</html>
```

## Percobaan 3 : Add Filter

Langkah percobaan

1. Buatlah file main4.js pada folder js yang telah dibuat pada percobaan 1
2. Buatlah file dengan nama percobaan4.html

3. Akses aplikasi yang telah anda kerjakan dengan menggunakan browser chrome (dengan web cam)
4. Amati hasil yang terjadi

main4.js

```
'use strict';

// Put variables in global scope to make them available to the browser console.
var audio = document.querySelector('audio');

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

function handleSuccess(stream) {
  var audioTracks = stream.getAudioTracks();
  console.log('Got stream with constraints:', constraints);
  console.log('Using audio device: ' + audioTracks[0].label);
  stream.oninactive = function() {
    console.log('Stream ended');
  };
  window.stream = stream; // make variable available to browser console
  audio.srcObject = stream;
}

function handleError(error) {
  console.log('navigator.getUserMedia error: ', error);
}

navigator.mediaDevices.getUserMedia(constraints)
  .then(handleSuccess).catch(handleError);
```

percobaan4.html

```
<html>
<head>

  <meta charset="utf-8">
  <meta name="description" content="WebRTC code samples">
  <meta name="mobile-web-app-capable" content="yes">
  <meta id="theme-color" name="theme-color"
content="#ffffff">

  <base target="_blank">

  <title> audio</title>

  <link rel="stylesheet" href="css/main.css">

</head>

<body>

  <div id="container">

    <h1>Percobaan 4: Get Audio</h1>

    <audio id="gum-local" controls autoplay></audio>

    <p class="warning">Warning: if you're not using
headphones, pressing play will cause feedback.</p>

  </div>

  <script src="js/main4.js"></script>

</body>
</html>
```