

## DAFTAR PUSTAKA

- Cadenhead, Tyson. 2015. *Socket.IO Cookbook*. Packt Publishing. ISBN 978-1-7858-8443-6
- Fernández, Luis López dkk. 2013. *Kurento: a media server technology for convergent WWW/mobile real-time multimedia communications supporting WebRTC*. IEEE 14th International Symposium.
- Flanagan, David. 2011. *JavaScript: The Definitive Guide, Sixth Edition*. O'Reilly Media. ISBN: 978-0-596-80552-4.
- Hartshorne, Richard and Haya Ajjan. 2009. *Examining student decisions to adopt Web 2.0 technologies: theory and empirical tests*. J Comput High Educ.
- Koran, Jaya Kumar C. 2002. *Aplikasi E-Learning Dalam Pengajaran dan Pembelajaran di Sekolah-Sekolah Malaysia*. Malaysia: Pasukan Projek Rintis Sekolah Bestari.
- Ndia, John Gichuki dkk. 2019. *A Survey of Cascading Style Sheets Complexity Metrics*. SSRN Electronic Journal.
- Osborn, Jeremy dkk. 2011. *Web Design with HTML and CSS Digital Classroom*. Wiley Publishing. ISBN: 978-0-470-58360-9.
- Rohman, Muhamad. 2016. *Rancang Bangun Aplikasi Aktivitas Live Coding Pada Pemrograman Web Secara Peer-to-Peer Pada Peramban Berbasis WebRTC*. Surabaya: Institut Teknologi Sepuluh Nopember. 9.

Safrin, Hermawan. 2020. *Implementasi Teknologi Web Real-Time Communication (WEBRTC) dalam Perancangan Kelas Virtual sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento*. Makassar: Universitas Hasanuddin.

Sari, Ani Oktarini. 2019. *Web Programming*. Yogyakarta: Graha Ilmu. ISBN: 978-623-228-221-6.

Tsonev, Krasimir. 2015. *Node.js by Example*. Packt Publishing. ISBN 978-1-78439-571-1.

Wang, Vanessa dkk. 2013. *The Defenitive Guide to HTML 5 WebSocket*. Apress. ISBN: 978-1-4302-4741-8.

Zhang, Lijing dan Xiaoxiao Shen. 2013. *Research and Development of Real-timeMonitoring System Based on WebSocketTechnology*. International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC).

# LAMPIRAN

## Script.js

```
1  const socket = io.connect("/");
2  const videoGrid = document.getElementById("video-grid");
3  const myVideo = document.createElement("video");
4  const showChat = document.querySelector("#showChat");
5  const backBtn = document.querySelector(".header__back");
6  myVideo.muted = true;
7
8  backBtn.addEventListener("click", () => {
9    document.querySelector(".main__left").style.display = "flex";
10   document.querySelector(".main__left").style.flex = "1";
11   document.querySelector(".main__right").style.display = "none";
12   document.querySelector(".header__back").style.display = "none";
13 });
14
15 showChat.addEventListener("click", () => {
16   document.querySelector(".main__right").style.display = "flex";
17   document.querySelector(".main__right").style.flex = "1";
18   document.querySelector(".main__left").style.display = "none";
19   document.querySelector(".header__back").style.display = "block";
20 });
21
22 const user = prompt("Enter your name");
23
24 // inisiasi PeerJS
25 var peer = new Peer({
26
27   // request STUN Server
28   config: { 'iceServers': [
29     { url: 'stun:stun.1.google.com:19302' },
30   ]}
31
32 }, {
33   path: "/peerjs",
34   host: "/",
35   port: "443",
36 });
37
38 let myVideoStream;
39 // Access user media
40 navigator.mediaDevices
41   .getUserMedia({
42     audio: true,
43     video: true,
44   })
```

```

45 .then((stream) => {
46   myVideoStream = stream;
47   addVideoStream(myVideo, stream); // Tampilkan video user
48   peer.on("call", (call) => { // ketika kita masuk room orang lain kita akan menerima call dari mereka
49     call.answer(stream); // stream media kita
50     const video = document.createElement("video"); // membuat tag video untuk mereka
51     call.on("stream", (userVideoStream) => { // menerima media mereka
52       addVideoStream(video, userVideoStream); // menampilkan video mereka
53     });
54   });
55   socket.on("user-connected", (userId) => { // ketika ada user baru bergabung
56     connectToNewUser(userId, stream);
57   });
58 });
59
60 const connectToNewUser = (userId, stream) => { // ini dijalankan ketika orang lain memasuki room
61   const call = peer.call(userId, stream); // melakukan call kepada orang yang baru bergabung
62   // menambahkan video mereka
63   const video = document.createElement("video");
64   call.on("stream", (userVideoStream) => {
65     addVideoStream(video, userVideoStream, userId);
66   });
67 };
68
69 peer.on("open", (id) => { // ketika kita join room
70   socket.emit("join-room", ROOM_ID, id, user);
71   socket.emit("message", user+" telah bergabung ke dalam room");
72   text.value = ""; // ketika user join room broadcast ke chat kalau user bergabung
73 });
74
75 const addVideoStream = (video, stream, userId) => { // menamnahkan video
76   video.srcObject = stream;
77   video.addEventListener("loadedmetadata", () => { // play video
78     video.play();
79     videoGrid.append(video);
80     video.setAttribute("controls", true);
81     video.setAttribute("id", userId);
82   });
83 };
84
85 let text = document.querySelector("#chat_message");
86 let send = document.getElementById("send");
87 let messages = document.querySelector(".messages");
88
89 // mengirim pesan ketika tombol kirim di pencet
90 send.addEventListener("click", (e) => {
91   if (text.value.length !== 0) {
92     socket.emit("message", text.value);
93     text.value = "";
94   }
95 });
96
97 // mengirim pesan ketika enter dipencet
98 text.addEventListener("keydown", (e) => {
99   if (e.key === "Enter" && text.value.length !== 0) {
100     socket.emit("message", text.value);
101     text.value = "";
102   }
103 });
104

```

```
106 const muteButton = document.querySelector("#muteButton");
107 const stopVideo = document.querySelector("#stopVideo");
108 // membuat event mute video
109 muteButton.addEventListener("click", () => {
110     const enabled = myVideoStream.getAudioTracks()[0].enabled;
111     if (enabled) {
112         myVideoStream.getAudioTracks()[0].enabled = false;
113         html = `fas fa-microphone-slash`></i>`;
114         muteButton.classList.toggle("background__red");
115         muteButton.innerHTML = html;
116     } else {
117         myVideoStream.getAudioTracks()[0].enabled = true;
118         html = `fas fa-microphone`></i>`;
119         muteButton.classList.toggle("background__red");
120         muteButton.innerHTML = html;
121     }
122 });
123
124 // membuat event hentikan video
125 stopVideo.addEventListener("click", () => {
126     const enabled = myVideoStream.getVideoTracks()[0].enabled;
127     if (enabled) {
128         myVideoStream.getVideoTracks()[0].enabled = false;
129         html = `fas fa-video-slash`></i>`;
130         stopVideo.classList.toggle("background__red");
131         stopVideo.innerHTML = html;
132     } else {
133         myVideoStream.getVideoTracks()[0].enabled = true;
134         html = `fas fa-video`></i>`;
135         stopVideo.classList.toggle("background__red");
136         stopVideo.innerHTML = html;
137     }
138 });
```

```

140 // membuat event ketika invite button di pencet
141 inviteButton.addEventListener("click", (e) => {
142     prompt(
143         "Copy this link and send it to people you want to meet with",
144         window.location.href
145     );
146 });
147
148 // ini dijalankan ketika ada pesan yang akan di broadcast
149 socket.on("createMessage", (message, userName) => {
150     messages.innerHTML =
151         messages.innerHTML +
152         `

## Server.js



```

1 // membuat express dan socket.io server
2 const express = require("express");
3 const app = express();
4
5 const https = require('https');
6 const http = require('http');
7 const fs = require('fs');
8
9 var options = {
10     key: fs.readFileSync('public/keys/server.key'),
11     cert: fs.readFileSync('public/keys/server.crt')
12 };
13
14
15 const server = https.createServer(options, app).listen(443, () => {
16 });
17
18 //app.set('port', process.env.PORT || process.env.OPENSIFT_NODEJS_PORT || 3030);
19
20 //const server = http.Server(app);
21
22 const { v4: uuidv4 } = require("uuid");
23
24 app.set("view engine", "ejs"); // memberitahukan express kalau kita menggunakan ejs
25

```


```

```

26 const io = require("socket.io")(server, {
27   cors: {
28     origin: '*'
29   }
30 });
31
32 const { ExpressPeerServer } = require("peer");
33
34 const peerServer = ExpressPeerServer(server, {
35   debug: true,
36 });
37
38 app.use("/peerjs", peerServer);
39 app.use(express.static("public"));
40
41 app.get("/", (req, res) => { // ketika url "/" diakses bukan halaman index
42   res.render("index");
43 });
44
45 app.get("/quick", (req, res) => { // ketika url "/quick" di akses generate random UUID dan
46   res.redirect(`/${uuidv4()}`);
47 });
48
49 // ketika user join spesifik room, render room tersebut
50 app.get("/:room", (req, res) => {
51   res.render("room", { roomId: req.params.room });
52 });
53
54 // ketika seseorang connect ke server
55 io.on("connection", (socket) => {
56   // ketika seseorang ingin bergabung ke room
57   socket.on("join-room", (roomId, userId, userName) => {
58
59     // join room
60     socket.join(roomId);
61     socket.to(roomId).broadcast.emit("user-connected", userId);
62     socket.on("message", (message) => {
63       io.to(roomId).emit("createMessage", message, userName);
64     });
65     socket.on("disconnect", ()=>{
66       io.to(roomId).emit("user-disconnected", userId);
67     });
68   });
69 });
70 //server.listen(process.env.PORT || 3030);

```



## Index.ejs

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
6     <meta name="description" content="" />
7     <meta name="author" content="" />
8     <title>WebRTC API</title>
9     <link rel="icon" type="image/x-icon" href="/assets/favicon.ico" />
10    <!-- Font Awesome icons (free version)-->
11    <script src="https://use.fontawesome.com/releases/v5.15.3/js/all.js" crossorigin="anonymous"></script>
12    <!-- Google fonts-->
13    <link rel="preconnect" href="https://fonts.gstatic.com" />
14    <link href="https://fonts.googleapis.com/css2?family=Tinos:ital,wght@0,400;0,700;1,400;1,700&display=swap" rel="stylesheet" />
15    <link href="https://fonts.googleapis.com/css2?family=DM+Sans:ital,wght@0,400;0,500;0,700;1,400;1,500;1,700&display=swap" rel="stylesheet" />
16    <!-- Core theme CSS (includes Bootstrap)-->
17    <link href="/css/styles.css" rel="stylesheet" />
18  </head>
19  <body>
20    <!-- Background Video-->
21    </img>
22    <!-- Masthead-->
23    <div class="masthead">
24      <div class="masthead-content text-white">
25        <div class="container-fluid px-4 px-lg-0">
26          <h1 class="fst-lh-1 mb-4">WebRTC API</h1>
27          <p class="mb-5">A teleconferencing platform that can support all your communication needs, including meetings, chats, webinars and online events.</p>
28          <!-- Room name input-->
29          <div class="row input-group-newsletter">
30            <div class="col"><input class="form-control" id="roomname" type="text" onkeypress="CheckSpaces(event)" placeholder="Enter a room name" />
31            <div class="col-auto">
32              <a class="btn btn-primary" id="createRoom">
33                Create
34              </a>
35            </div>
36          </div>
37          <div class="row input-group-newsletter">
38            <p class="mb-txt">Or</p>
39            <a href="/quick"><button class="btn btn-secondary" >Quick Join</button></a>
40          </div>
41        </div>
42      </div>
43    </div>
44    <script type="text/javascript">
45      function CheckSpaces(event) {
46        if(event.which == 32) {
47          event.preventDefault();
48          return false;
49        }
50      };
51      var link= document.getElementById('createRoom');
52      var input= document.getElementById('roomname');
53      input.onchange=input.onkeyup= function() {
54        link.href = '/' +input.value;
55      };
56    </script>
57    <!-- Bootstrap core JS-->
58    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
59    <!-- Core theme JS-->
60    <script src="/js/scripts.js"></script>
61    <script src="https://cdn.startbootstrap.com/sb-forms-latest.js"></script>
62  </body>
63 </html>
```

## Room.ejs

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Web RTC RPi</title>
7     <link rel="stylesheet" href="style.css" />
8     <script src="/socket.io/socket.io.js"></script>
9     <script src="https://kit.fontawesome.com/c939d0e917.js"></script>
10    <script src="https://unpkg.com/peerjs@1.3.1/dist/peerjs.min.js"></script>
11    <script>
12      const ROOM_ID = "<%= roomId %>";
13    </script>
14  </head>
15  <body>
16    <div class="header">
17      <div class="logo">
18        <div class="header__back">
19          <i class="fas fa-angle-left"></i>
20        </div>
21        <h3>Web RTC RPi</h3>
22      </div>
23    </div>
24    <div class="main">
25      <div class="main__left">
26        <div class="videos__group">
27          <div id="video-grid">
28
29          </div>
30        </div>
31        <div class="options">
32          <div class="options__left">
33
34          <div id="stopVideo" class="options__button">
35            <i class="fa fa-video-camera"></i>
36          </div>
37          <div id="muteButton" class="options__button">
38            <i class="fa fa-microphone"></i>
39          </div>
40          <!--
41          <div id="screenButton" class="options__button-choose">
42            <i class="fa fa-desktop"></i>
43          </div>
44          -->
45          <div id="showChat" class="options__button">
46            <i class="fa fa-comment"></i>
47          </div>
48        </div>
49      </div>
50    </div>
51  </body>
52 </html>
```

```

48     <div class="options__right">
49         <div id="inviteButton" class="options__button">
50             <i class="fas fa-user-plus"></i>
51         </div>
52     </div>
53 </div>
54 </div>
55 <div class="main__right">
56     <div class="main__chat_window">
57         <div class="messages">
58
59         </div>
60     </div>
61     <div class="main__message_container">
62         <input id="chat_message" type="text" autocomplete="off" placeholder="Type message here...">
63         <div id="send" class="options__button">
64             <i class="fa fa-plus" aria-hidden="true"></i>
65         </div>
66     </div>
67 </div>
68 </div>
69 </body>
70 <script src="script.js"></script>
71 </html>

```

## Style.css

```

1  @import url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap");
2
3  :root {
4      --main-darklg: #1d2635;
5      --main-dark: #161d29;
6      --primary-color: #2f80ec;
7      --main-light: #eeeeee;
8      --green: rgb(1, 179, 16);
9      font-family: "Poppins", sans-serif;
10 }
11
12 * {
13     margin: 0;
14     padding: 0;
15 }
16
17 .header {
18     display: flex;
19     justify-content: center;
20     align-items: center;
21     height: 8vh;
22     position: relative;
23     width: 100%;
24     background-color: var(--main-darklg);
25 }
26
27 .logo > h3 {
28     color: var(--main-light);
29 }

```

```
31 .main {
32     overflow: hidden;
33     height: 92vh;
34     display: flex;
35 }
36
37 .main__left {
38     flex: 0.7;
39     display: flex;
40     flex-direction: column;
41 }
42
43 .videos__group {
44     flex-grow: 1;
45     display: flex;
46     justify-content: center;
47     align-items: center;
48     padding: 1rem;
49     background-color: var(--main-dark);
50     overflow: auto;
51 }
52
53 video {
54     width: 260px;
55     height: 180px;
56     border-radius: 0.5rem;
57     margin: 0.5rem;
58     object-fit: cover;
59     transform: rotateY(180deg);
60     -webkit-transform: rotateY(180deg);
61     -moz-transform: rotateY(180deg);
62 }
63
64 .options {
65     padding: 1rem;
66     display: flex;
67     background-color: var(--main-darklg);
68 }
69
70 .options__left {
71     display: flex;
72 }
73
74 .options__right {
75     margin-left: auto;
76 }
77
```

```
78 .options__button {
79     display: flex;
80     justify-content: center;
81     align-items: center;
82     background-color: var(--primary-color);
83     height: 50px;
84     border-radius: 5px;
85     color: var(--main-light);
86     font-size: 1.2rem;
87     width: 50px;
88     margin: 0 0.5rem;
89 }
90
91 .options__button-choose {
92     display: flex;
93     justify-content: center;
94     align-items: center;
95     background-color: var(--green);
96     height: 50px;
97     border-radius: 5px;
98     color: var(--main-light);
99     font-size: 1.2rem;
100    width: 50px;
101    margin: 0 0.5rem;
102 }
103
104 .background__red {
105     background-color: #f6484a;
106 }
107
108 .main__right {
109     display: flex;
110     flex-direction: column;
111     flex: 0.3;
112     background-color: #242f41;
113 }
114
115 .main__chat_window {
116     flex-grow: 1;
117     overflow-y: scroll;
118 }
119
120 .main__chat_window::-webkit-scrollbar {
121     display: none;
122 }
123
```

```
124 .main_message_container {
125     padding: 1rem;
126     display: flex;
127     align-items: center;
128     justify-content: center;
129 }
130
131 .main_message_container > input {
132     height: 50px;
133     flex: 1;
134     font-size: 1rem;
135     border-radius: 5px;
136     padding-left: 20px;
137     border: none;
138 }
139
140 .messages {
141     display: flex;
142     flex-direction: column;
143     margin: 1.5rem;
144 }
145
146 .message {
147     display: flex;
148     flex-direction: column;
149 }
150
151 .message > b {
152     color: #eeeeee;
153     display: flex;
154     align-items: center;
155
156     text-transform: capitalize;
157 }
158
159 .message > b > i {
160     margin-right: 0.7rem;
161     font-size: 1.5rem;
162 }
163
164 .message > span {
165     background-color: #eeeeee;
166     margin: 1rem 0;
167     padding: 1rem;
168     border-radius: 5px;
169 }
```

```
170 #video-grid {
171     display: flex;
172     justify-content: center;
173     flex-wrap: wrap;
174     margin-block: auto;
175 }
176
177 #showChat {
178     display: none;
179 }
180
181 .header__back {
182     display: none;
183     position: absolute;
184     font-size: 1.3rem;
185     top: 17px;
186     left: 28px;
187     color: #fff;
188 }
189
190 @media (max-width: 700px) {
191     .main__right {
192         display: none;
193     }
194     .main__left {
195         width: 100%;
196         flex: 1;
197     }
198
199     video {
200         height: auto;
201         width: 100%;
202     }
203
204     #showChat {
205         display: flex;
206     }
207 }
```



UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

BERITA ACARA UJIAN SKRIPSI

Pada hari ini Kamis tanggal 19 Agustus 2021 Pukul 13.00 WITA - Selesai bertempat di Lab. UBICON Departemen Teknik Informatika Gowa, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Muh. Nur Alamsyah Ibrahim  
No. Stambuk : D421 14 513  
Fakultas/Departemen : Teknik / Teknik Informatika  
Judul Skripsi : "Implementasi Web dan Signaling Server untuk Web Real-Time Communication (WebRTC) pada Raspberry Pi"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	Nama	Jabatan	Tanda tangan
1.	Dr. Eng. Muhammad Niswar, ST., M.I.T	Pemb I/Ketua	1.
2.	Iqra Aswad, ST., M.T	Pemb II/Sekretaris	2.
3.	Dr. Ir. Zahir Zainuddin, M.Sc	Anggota	3.
4.	Dr. Eng. Zulkifli Tahir, ST., M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : Lulus / Tidak lulus dengan nilai angka ..... ..... dan huruf .... .....

Gowa, 19 Agustus 2021

Ketua/Sekretaris Panitia Ujian,

Dr. Eng. Muhammad Niswar, ST., M.I.T



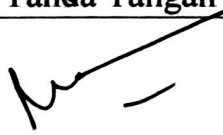



## LEMBAR PERBAIKAN SKRIPSI

**“IMPLEMENTASI WEB DAN SIGNALING SERVER UNTUK WEB REAL-TIME COMMUNICATION (WEBRTC) PADA RASPBERRY PI”**  
**OLEH:**

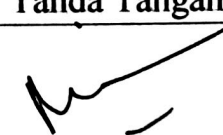
**MUH NUR ALAMSYAH IBRAHIM**  
**D42114513**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 20 Agustus 2021. Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Eng. Muhammad Niswar, ST., M.IT.	
Sekretaris	Iqra Aswad, ST., MT.	
Anggota	Dr. Ir. Zahir Zainuddin, M.SC.	
	Dr. Eng. Zulkifli Tahir, S.T., M.SC.	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Eng. Muhammad Niswar, ST., M.IT.	
II	Iqra Aswad, ST., MT.	