

**SKRIPSI**

**IMPLEMENTASI *VIDEO CONFERENCE ONE-TO-MANY* BERBASIS  
TEKNOLOGI WEBRTC MENGGUNAKAN MEDIA SERVER UNTUK  
PEMBELAJARAN *ONLINE***

**Disusun dan diajukan oleh:**

**IRHAM SAHBANA  
D421 16 513**



**PROGRAM STUDI TEKNIK INFORMATIKA S1  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
GOWA  
2023**

## LEMBAR PENGESAHAN SKRIPSI

### LEMBAR PENGESAHAN SKRIPSI

#### IMPLEMENTASI VIDEO CONFERENCE ONE-TO-MANY BERBASIS TEKNOLOGI WEBRTC MENGGUNAKAN MEDIA SERVER UNTUK PEMBELAJARAN ONLINE

Disusun dan diajukan oleh

**IRHAM SAHBANA**


**D42116513**


Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 09 Agustus 2023 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,


Pembimbing Utama,

Pembimbing Pendamping,

  
Dr. Eng. Ir. Muhammad Niswar, S.T., M.InfoTech.  
Nip. 197309221999031001

  
Igra' Aswad, S.T., M.T  
Nip. 199011282019043001

Ketua Program Studi,

  
Prof. Dr. Ir. Indrabati, S.T., M.T., M.Bus.Sys., IPM., ASEAN. Eng.  
Nip. 19750716 200212 1 004

## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;  
Nama : Irham Sahbana  
NIM : D42116513  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

IMPLEMENTASI VIDEO CONFERENCE ONE-TO-MANY BERBASIS  
TEKNOLOGI WEBRTC MENGGUNAKAN MEDIA SERVER UNTUK  
PEMBELAJARAN ONLINE

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 30 Juli 2023

Yang Menyatakan

  
Irham Sahbana

## ABSTRAK

**IRHAM SAHBANA.** *Implementasi Video Conference One-To-Many Berbasis Teknologi WebRTC Menggunakan Media Server untuk Pembelajaran Online* (dibimbing oleh Dr. Eng. Muhammad Niswar, S.T., M.IT. dan Iqra Aswad, S.T., M.T.)

Beberapa *platform* pembelajaran *online* memiliki penerapan dan metode sendiri dalam menerapkan pelajaran online seperti metode *self-paced*. Permasalahan dari metode ini yakni kehilangan keuntungan real-time. Selain itu, banyak mengkonsumsi resource yang menguras kuota internet serta penggunaan CPU dan RAM yang besar. Penelitian ini bertujuan untuk membangun sistem pembelajaran *online* dengan mengimplementasikan *audio*, *video*, *screen recording* berbasis teknologi WebRTC menggunakan media server serta *collaborative whiteboard* dan mengetahui kinerja dari teknologi WebRTC menggunakan Kurento Media Server dalam pembelajaran *online*. Penelitian ini menerapkan teknologi WebRTC (*Web Real-Time Communications*) dengan Kurento Media Server untuk mengintegrasikan komunikasi *real-time* yang ramah *resource*. Penelitian ini membandingkan performa tiga aplikasi yakni Gmeet, Zoom, serta aplikasi yang dibuat (MeetUp) dalam hal penggunaan CPU, penggunaan RAM, dan throughput saat menggunakan media webcam dan *screen share*. Hasilnya menunjukkan bahwa aplikasi MeetUp memiliki penggunaan CPU dan RAM terendah di setiap media pada keseluruhan client yang diujikan, sedangkan pada perbandingan *throughput*, aplikasi Gmeet lebih hemat dalam penggunaan data saat mendownload dan aplikasi MeetUp lebih hemat dalam penggunaan data saat mengupload pada kedua jenis media. Dalam kesimpulannya, penelitian ini menunjukkan bahwa penggunaan teknologi WebRTC dengan Kurento Media Server dapat memberikan solusi yang ramah *resource* dalam pembelajaran online.

Kata kunci: WebRTC, Performansi, CPU, RAM, Throughput, Kurento Media Server

## ABSTRACT

**IRHAM SAHBANA.** *Implementation of One-To-Many Video Conference Based on WebRTC Technology Using Media Server for Online Learning* (supervised by Dr. Eng. Muhammad Niswar, S.T., M.IT. and Iqra Aswad, S.T., M.T.)

Several online learning platforms have their own implementation and method of delivering online lessons, such as the self-paced method. The problem with this method is the loss of real-time benefits. In addition, it consumes a lot of resources that drain internet quotas as well as large CPU and RAM usage. This study aims to develop an online learning system by implementing audio, video, and screen recording based on WebRTC technology using a media server and collaborative whiteboard, and to evaluate the performance of WebRTC technology using Kurento Media Server in online learning. This study combines WebRTC technology with Kurento Media Server to integrate resource-friendly real-time communication. The study compares the performance of three applications, namely Gmeet, Zoom, and the custom-made application (MeetUp), in terms of CPU usage, RAM usage, and throughput when using a webcam and screen sharing media. The results show that the MeetUp application has the lowest CPU and RAM usage in every media on all tested clients, while in terms of throughput comparison, the Gmeet application is more economical in data usage when downloading and the MeetUp application is more economical in data usage when uploading on both types of media. In conclusion, this study shows that the use of WebRTC technology with Kurento Media Server can provide resource-friendly solutions for online learning.

Keywords: WebRTC, Performance, CPU, RAM, Throughput, Kurento Media Server

## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	ii
PERNYATAAN KEASLIAN.....	iii
ABSTRAK.....	iv
ABSTRACT.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	x
KATA PENGANTAR.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 <i>Video Conference</i> .....	5
2.2 <i>Web Real-Time Communication (WebRTC)</i> .....	6
2.2 <i>Network Address Translation (NAT)</i> .....	6
2.3 <i>Network Address Translation Traversal (NAT Traversal)</i> .....	7
2.4 <i>Real-time Transport Protocol (RTP)</i> .....	7
2.5 <i>RTP Control Protocol (RTCP)</i> .....	7
2.6 <i>Session Description Protocol (SDP)</i> .....	7
2.7 <i>Kurento Media Server</i> .....	8
2.8 <i>Virtual Machine</i> .....	10
2.9 <i>Docker</i> .....	11

2.10 Container.....	11
2.11 Docker Compose.....	12
2.12 NetBalancer.....	13
<b>BAB III METODE PENELITIAN.....</b>	<b>14</b>
3.1 Waktu dan Lokasi Penelitian.....	14
3.2 Perancangan.....	14
3.3 Bahan Uji dan Alat.....	18
3.4 Teknik Pengumpulan Data.....	18
3.5 Teknik Analisis.....	20
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>21</b>
4.1 Analisis.....	21
4.1.1 Performansi di sisi <i>server</i> .....	21
4.1.1.1 Pengujian CPU.....	21
4.1.1.2 Pengujian RAM.....	23
4.1.1.3 Pengujian <i>Throughput</i> .....	24
4.1.2 Performansi di sisi <i>client</i> .....	26
4.1.2.1 Pengujian CPU.....	26
4.1.2.1.1 Presenter.....	27
4.1.2.1.2 Viewer 1.....	28
4.1.2.1.3 <i>Viewer 2</i> .....	29
4.1.2.1.4 <i>Viewer 3</i> .....	30
4.1.2.1.5 <i>Viewer 4</i> .....	31
4.1.2.1.6 <i>Viewer 5</i> .....	32
4.1.2.1.7 <i>Viewer 6</i> .....	33
4.1.2.1.8 <i>Viewer 7</i> .....	34
4.1.2.1.9 <i>Viewer 8</i> .....	35

4.1.2.1.10 <i>Viewer 9</i> .....	36
4.1.2.1.11 <i>Viewer 10</i> .....	37
4.1.2.2 <i>Pengujian RAM</i> .....	38
4.1.2.2.1 <i>Presenter</i> .....	38
4.1.2.2.2 <i>Viewer 1</i> .....	39
4.1.2.2.3 <i>Viewer 2</i> .....	40
4.1.2.2.4 <i>Viewer 3</i> .....	41
4.1.2.2.5 <i>Viewer 4</i> .....	42
4.1.2.2.6 <i>Viewer 5</i> .....	43
4.1.2.2.7 <i>Viewer 6</i> .....	44
4.1.2.2.8 <i>Viewer 7</i> .....	45
4.1.2.2.9 <i>Viewer 8</i> .....	46
4.1.2.2.10 <i>Viewer 9</i> .....	47
4.1.2.2.11 <i>Viewer 10</i> .....	48
4.1.2.3 <i>Throughput</i> .....	49
4.1.2.3.1 <i>Presenter</i> .....	49
4.1.2.3.2 <i>Viewer 1</i> .....	51
4.1.2.3.3 <i>Viewer 2</i> .....	53
4.1.2.3.4 <i>Viewer 3</i> .....	55
4.1.2.3.5 <i>Viewer 4</i> .....	57
4.1.2.3.6 <i>Viewer 5</i> .....	59
4.1.2.3.7 <i>Viewer 6</i> .....	61
4.1.2.3.8 <i>Viewer 7</i> .....	63
4.1.2.3.9 <i>Viewer 8</i> .....	65
4.1.2.3.10 <i>Viewer 9</i> .....	67
4.1.2.3.11 <i>Viewer 10</i> .....	69



4.2 Pembahasan.....	71
4.2.1 Performansi di sisi <i>server</i> .....	71
4.2.1.1 Penggunaan CPU.....	71
4.2.1.2 Penggunaan RAM.....	71
4.2.1.1 <i>Throughput</i> .....	72
4.2.2 Performansi di sisi <i>client</i> .....	72
4.2.2.1 Penggunaan CPU.....	72
4.2.2.2 Penggunaan RAM.....	73
4.2.2.1 <i>Throughput</i> .....	74
BAB V KESIMPULAN DAN SARAN.....	75
5.1 Kesimpulan.....	75
5.2 Saran.....	76
DAFTAR PUSTAKA.....	77
LAMPIRAN.....	79

## DAFTAR GAMBAR

Gambar 1. Contoh SDP offer.....	8
Gambar 2. Contoh SDP answer.....	8
Gambar 3. Kelebihan kurento media server .....	10
Gambar 4. Perbedaan teknologi container dan virtual machine .....	12
Gambar 5. Desain Sistem WebRTC.....	15
Gambar 6. Arsitektur media pipeline yang dibuat.....	16
Gambar 7. Use case diagram system.....	17
Gambar 8. Monitoring penggunaan CPU menggunakan perintah docker stats.....	21
Gambar 9. Performansi penggunaan CPU saat menggunakan webcam test case 1 .....	22
Gambar 10. Performansi penggunaan CPU saat menggunakan webcam test case 2 .....	22
Gambar 11. Performansi pengguna CPU saat menggunakan screen sharing test case 1.....	22
Gambar 12. Performansi pengguna CPU saat menggunakan screen sharing test case 2.....	22
Gambar 13. Monitoring penggunaan RAM menggunakan perintah “docker stats” .....	23
Gambar 14. Performansi penggunaan RAM saat menggunakan webcam test case 1 .....	23
Gambar 15. Performansi penggunaan RAM saat menggunakan webcam test case 2 .....	23
Gambar 16. Performansi penggunaan RAM saat menggunakan screen sharing test case 1.....	23
Gambar 17. Performansi penggunaan RAM saat menggunakan screen sharing test case 2.....	24
Gambar 18. Monitoring throughput menggunakan perintah “vnstat -l”.....	24
Gambar 19. Performansi download saat menggunakan webcam test case 1.....	24
Gambar 20. Performansi download saat menggunakan webcam test case 2.....	24
Gambar 21. Performansi upload saat menggunakan webcam test case 1.....	24

Gambar 22. Performansi upload saat menggunakan webcam test case 2.....	25
Gambar 23. Performansi download saat menggunakan screen sharing test case 125	
Gambar 24. Performansi download saat menggunakan screen sharing test case 225	
Gambar 25. Performansi upload saat menggunakan screen sharing test case 1....	25
Gambar 26. Performansi upload saat menggunakan screen sharing test case 2....	26
Gambar 27. Performansi penggunaan CPU presenter saat menggunakan webcam test case 1.....	26
Gambar 29. Performansi penggunaan CPU presenter saat menggunakan screen sharing test case 1.....	27
Gambar 28. Performansi penggunaan CPU presenter saat menggunakan webcam test case 2.....	27
Gambar 30. Performansi penggunaan CPU presenter saat menggunakan screen sharing test case 2.....	27
Gambar 31. Performansi penggunaan CPU viewer 1 saat menggunakan webcam test case 1.....	27
Gambar 32. Performansi penggunaan CPU viewer 1 saat menggunakan screen sharing test case 1.....	28
Gambar 33. Performansi penggunaan CPU viewer 1 saat menggunakan webcam test case 2.....	28
Gambar 34. Performansi penggunaan CPU viewer 1 saat menggunakan screen sharing test case 2.....	28
Gambar 35. Performansi penggunaan CPU viewer 2 saat menggunakan webcam test case 1.....	28
Gambar 36. Performansi penggunaan CPU viewer 2 saat menggunakan screen sharing test case 1.....	29
Gambar 37. Performansi penggunaan CPU viewer 2 saat menggunakan webcam test case 2.....	29
Gambar 38. Performansi penggunaan CPU viewer 2 saat menggunakan screen sharing test case 2.....	29
Gambar 39. Performansi penggunaan CPU viewer 3 saat menggunakan webcam test case 1.....	29

Gambar 40. Performansi penggunaan CPU viewer 3 saat menggunakan screen sharing test case 1.....	30
Gambar 41. Performansi penggunaan CPU viewer 3 saat menggunakan webcam test case 2.....	30
Gambar 42. Performansi penggunaan CPU viewer 3 saat menggunakan screen sharing test case 2.....	30
Gambar 43. Performansi penggunaan CPU viewer 4 saat menggunakan webcam test case 1.....	30
Gambar 44. Performansi penggunaan CPU viewer 4 saat menggunakan screen sharing test case 1.....	31
Gambar 45. Performansi penggunaan CPU viewer 4 saat menggunakan webcam test case 2.....	31
Gambar 46. Performansi penggunaan CPU viewer 4 saat menggunakan screen sharing test case 2.....	31
Gambar 47. Performansi penggunaan CPU viewer 5 saat menggunakan webcam test case 1.....	31
Gambar 48. Performansi penggunaan CPU viewer 5 saat menggunakan screen sharing test case 1.....	32
Gambar 49. Performansi penggunaan CPU viewer 5 saat menggunakan webcam test case 2.....	32
Gambar 50. Performansi penggunaan CPU viewer 5 saat menggunakan screen sharing test case 2.....	32
Gambar 51. Performansi penggunaan CPU viewer 6 saat menggunakan webcam test case 1.....	32
Gambar 52. Performansi penggunaan CPU viewer 6 saat menggunakan screen sharing test case 1.....	33
Gambar 53. Performansi penggunaan CPU viewer 6 saat menggunakan webcam test case 2.....	33
Gambar 54. Performansi penggunaan CPU viewer 6 saat menggunakan screen sharing test case 2.....	33
Gambar 55. Performansi penggunaan CPU viewer 7 saat menggunakan webcam test case 1.....	33

Gambar 56. Performansi penggunaan CPU viewer 7 saat menggunakan screen sharing test case 1.....	34
Gambar 57. Performansi penggunaan CPU viewer 7 saat menggunakan webcam test case 2.....	34
Gambar 58. Performansi penggunaan CPU viewer 7 saat menggunakan screen sharing test case 2.....	34
Gambar 59. Performansi penggunaan CPU viewer 8 saat menggunakan webcam test case 1.....	34
Gambar 60. Performansi penggunaan CPU viewer 8 saat menggunakan screen sharing test case 1.....	35
Gambar 61. Performansi penggunaan CPU viewer 8 saat menggunakan webcam test case 2.....	35
Gambar 62. Performansi penggunaan CPU viewer 8 saat menggunakan screen sharing test case 2.....	35
Gambar 63. Performansi penggunaan CPU viewer 9 saat menggunakan webcam test case 1.....	35
Gambar 64. Performansi penggunaan CPU viewer 9 saat menggunakan screen sharing test case 1.....	36
Gambar 65. Performansi penggunaan CPU viewer 9 saat menggunakan webcam test case 2.....	36
Gambar 66. Performansi penggunaan CPU viewer 9 saat menggunakan screen sharing test case 2.....	36
Gambar 67. Performansi penggunaan CPU viewer 10 saat menggunakan webcam test case 1.....	36
Gambar 68. Performansi penggunaan CPU viewer 10 saat menggunakan screen sharing test case 1.....	37
Gambar 69. Performansi penggunaan CPU viewer 10 saat menggunakan webcam test case 2.....	37
Gambar 70. Performansi penggunaan CPU viewer 10 saat menggunakan screen sharing test case 2.....	37
Gambar 71. Performansi penggunaan RAM presenter saat menggunakan webcam test case 1.....	37

Gambar 72. Performansi penggunaan RAM presenter saat menggunakan screen sharing test case 1.....	38
Gambar 73. Performansi penggunaan RAM presenter saat menggunakan webcam test case 2.....	38
Gambar 74. Performansi penggunaan RAM presenter saat menggunakan screen sharing test case 2.....	38
Gambar 75. Performansi penggunaan RAM viewer 1 saat menggunakan webcam test case 1.....	38
Gambar 76. Performansi penggunaan RAM viewer 1 saat menggunakan screen sharing test case 1.....	39
Gambar 77. Performansi penggunaan RAM viewer 1 saat menggunakan webcam test case 2.....	39
Gambar 78. Performansi penggunaan RAM viewer 1 saat menggunakan screen sharing test case 2.....	39
Gambar 79. Performansi penggunaan RAM viewer 2 saat menggunakan webcam test case 1.....	39
Gambar 80. Performansi penggunaan RAM viewer 2 saat menggunakan screen sharing test case 1.....	40
Gambar 81. Performansi penggunaan RAM viewer 2 saat menggunakan webcam test case 2.....	40
Gambar 82. Performansi penggunaan RAM viewer 2 saat menggunakan screen sharing test case 2.....	40
Gambar 83. Performansi penggunaan RAM viewer 3 saat menggunakan webcam test case 1.....	40
Gambar 84. Performansi penggunaan RAM viewer 3 saat menggunakan screen sharing test case 1.....	41
Gambar 85. Performansi penggunaan RAM viewer 3 saat menggunakan webcam test case 2.....	41
Gambar 86. Performansi penggunaan RAM viewer 3 saat menggunakan screen sharing test case 2.....	41
Gambar 87. Performansi penggunaan RAM viewer 4 saat menggunakan webcam test case 1.....	41

Gambar 88. Performansi penggunaan RAM viewer 4 saat menggunakan screen sharing test case 1.....	42
Gambar 89. Performansi penggunaan RAM viewer 4 saat menggunakan webcam test case 2.....	42
Gambar 90. Performansi penggunaan RAM viewer 4 saat menggunakan screen sharing test case 2.....	42
Gambar 91. Performansi penggunaan RAM viewer 5 saat menggunakan webcam test case 1.....	42
Gambar 92. Performansi penggunaan RAM viewer 5 saat menggunakan screen sharing test case 1.....	43
Gambar 93. Performansi penggunaan RAM viewer 5 saat menggunakan webcam test case 2.....	43
Gambar 94. Performansi penggunaan RAM viewer 5 saat menggunakan screen sharing test case 2.....	43
Gambar 95. Performansi penggunaan RAM viewer 6 saat menggunakan webcam test case 1.....	43
Gambar 96. Performansi penggunaan RAM viewer 6 saat menggunakan screen sharing test case 1.....	44
Gambar 97. Performansi penggunaan RAM viewer 6 saat menggunakan webcam test case 2.....	44
Gambar 98. Performansi penggunaan RAM viewer 6 saat menggunakan screen sharing test case 2.....	44
Gambar 99. Performansi penggunaan RAM viewer 7 saat menggunakan webcam test case 1.....	44
Gambar 100. Performansi penggunaan RAM viewer 7 saat menggunakan screen sharing test case 1.....	45
Gambar 101. Performansi penggunaan RAM viewer 7 saat menggunakan webcam test case 2.....	45
Gambar 102. Performansi penggunaan RAM viewer 7 saat menggunakan screen sharing test case 2.....	45
Gambar 103. Performansi penggunaan RAM viewer 8 saat menggunakan webcam test case 1.....	45

Gambar 104. Performansi penggunaan RAM viewer 8 saat menggunakan screen sharing test case 1.....	46
Gambar 105. Performansi penggunaan RAM viewer 8 saat menggunakan webcam test case 2.....	46
Gambar 106. Performansi penggunaan RAM viewer 8 saat menggunakan screen sharing test case 2.....	46
Gambar 107. Performansi penggunaan RAM viewer 9 saat menggunakan webcam test case 1.....	46
Gambar 108. Performansi penggunaan RAM viewer 9 saat menggunakan screen sharing test case 1.....	47
Gambar 109. Performansi penggunaan RAM viewer 9 saat menggunakan webcam test case 2.....	47
Gambar 110. Performansi penggunaan RAM viewer 9 saat menggunakan screen sharing test case 2.....	47
Gambar 111. Performansi penggunaan RAM viewer 10 saat menggunakan webcam test case 1.....	47
Gambar 112. Performansi penggunaan RAM viewer 10 saat menggunakan screen sharing test case 1.....	48
Gambar 113. Performansi penggunaan RAM viewer 10 saat menggunakan webcam test case 2.....	48
Gambar 114. Performansi penggunaan RAM viewer 10 saat menggunakan screen sharing test case 2.....	48
Gambar 115. Performansi throughput (download) presenter saat menggunakan webcam test case 1.....	48
Gambar 116. Performansi throughput (download) presenter saat menggunakan screen sharing test case 1.....	49
Gambar 117. Performansi throughput (upload) presenter saat menggunakan webcam test case 1.....	49
Gambar 118. Performansi throughput (upload) presenter saat menggunakan screen sharing test case 1.....	49
Gambar 119. Performansi throughput (download) presenter saat menggunakan webcam test case 2.....	49



Gambar 120. Performansi throughput (download) presenter saat menggunakan screen sharing test case 2.....	50
Gambar 121. Performansi throughput (upload) presenter saat menggunakan webcam test case 2.....	50
Gambar 122. Performansi throughput (upload) presenter saat menggunakan screen sharing test case 2.....	50
Gambar 123. Performansi throughput (download) viewer 1 saat menggunakan webcam test case 1.....	50
Gambar 124. Performansi throughput (download) viewer 1 saat menggunakan screen sharing test case 1.....	51
Gambar 125. Performansi throughput (upload) viewer 1 saat menggunakan webcam test case 1.....	51
Gambar 126. Performansi throughput (upload) viewer 1 saat menggunakan screen sharing test case 1.....	51
Gambar 127. Performansi throughput (download) viewer 1 saat menggunakan webcam test case 2.....	51
Gambar 128. Performansi throughput (download) viewer 1 saat menggunakan screen sharing test case 2.....	52
Gambar 129. Performansi throughput (upload) viewer 1 saat menggunakan webcam test case 2.....	52
Gambar 130. Performansi throughput (upload) viewer 1 saat menggunakan screen sharing test case 2.....	52
Gambar 131. Performansi throughput (download) viewer 2 saat menggunakan webcam test case 1.....	52
Gambar 132. Performansi throughput (download) viewer 2 saat menggunakan screen sharing test case 1.....	53
Gambar 133. Performansi throughput (upload) viewer 2 saat menggunakan webcam test case 1.....	53
Gambar 134. Performansi throughput (upload) viewer 2 saat menggunakan screen sharing test case 1.....	53
Gambar 135. Performansi throughput (download) viewer 2 saat menggunakan webcam test case 2.....	53

Gambar 136. Performansi throughput (download) viewer 2 saat menggunakan screen sharing test case 2.....	54
Gambar 137. Performansi throughput (upload) viewer 2 saat menggunakan webcam test case 2.....	54
Gambar 138. Performansi throughput (upload) viewer 2 saat menggunakan screen sharing test case 2.....	54
Gambar 139. Performansi throughput (download) viewer 3 saat menggunakan webcam test case 1.....	54
Gambar 140. Performansi throughput (download) viewer 3 saat menggunakan screen sharing test case 1.....	55
Gambar 141. Performansi throughput (upload) viewer 3 saat menggunakan webcam test case 1.....	55
Gambar 142. Performansi throughput (upload) viewer 3 saat menggunakan screen sharing test case 1.....	55
Gambar 143. Performansi throughput (download) viewer 3 saat menggunakan webcam test case 2.....	55
Gambar 144. Performansi throughput (download) viewer 3 saat menggunakan screen sharing test case 2.....	56
Gambar 145. Performansi throughput (upload) viewer 3 saat menggunakan webcam test case 2.....	56
Gambar 146. Performansi throughput (upload) viewer 3 saat menggunakan screen sharing test case 2.....	56
Gambar 147. Performansi throughput (download) viewer 4 saat menggunakan webcam test case 1.....	56
Gambar 148. Performansi throughput (download) viewer 4 saat menggunakan screen sharing test case 1.....	57
Gambar 149. Performansi throughput (upload) viewer 4 saat menggunakan webcam test case 1.....	57
Gambar 150. Performansi throughput (upload) viewer 4 saat menggunakan screen sharing test case 1.....	57
Gambar 151. Performansi throughput (download) viewer 4 saat menggunakan webcam test case 2.....	57

Gambar 152. Performansi throughput (download) viewer 4 saat menggunakan screen sharing test case 2.....	58
Gambar 153. Performansi throughput (upload) viewer 4 saat menggunakan webcam test case 2.....	58
Gambar 154. Performansi throughput (upload) viewer 4 saat menggunakan screen sharing test case 2.....	58
Gambar 155. Performansi throughput (download) viewer 5 saat menggunakan webcam test case 1.....	58
Gambar 156. Performansi throughput (download) viewer 5 saat menggunakan screen sharing test case 1.....	59
Gambar 157. Performansi throughput (upload) viewer 5 saat menggunakan webcam test case 1.....	59
Gambar 158. Performansi throughput (upload) viewer 5 saat menggunakan screen sharing test case 1.....	59
Gambar 159. Performansi throughput (download) viewer 5 saat menggunakan webcam test case 2.....	59
Gambar 160. Performansi throughput (download) viewer 5 saat menggunakan screen sharing test case 2.....	60
Gambar 161. Performansi throughput (upload) viewer 5 saat menggunakan webcam test case 2.....	60
Gambar 162. Performansi throughput (upload) viewer 5 saat menggunakan screen sharing test case 2.....	60
Gambar 163. Performansi throughput (download) viewer 6 saat menggunakan webcam test case 1.....	60
Gambar 164. Performansi throughput (download) viewer 6 saat menggunakan screen sharing test case 1.....	61
Gambar 165. Performansi throughput (upload) viewer 6 saat menggunakan webcam test case 1.....	61
Gambar 166. Performansi throughput (upload) viewer 6 saat menggunakan screen sharing test case 1.....	61
Gambar 167. Performansi throughput (download) viewer 6 saat menggunakan webcam test case 2.....	61

Gambar 168. Performansi throughput (download) viewer 6 saat menggunakan screen sharing test case 2.....	62
Gambar 169. Performansi throughput (upload) viewer 6 saat menggunakan webcam test case 2.....	62
Gambar 170. Performansi throughput (upload) viewer 6 saat menggunakan screen sharing test case 2.....	62
Gambar 171. Performansi throughput (download) viewer 7 saat menggunakan webcam test case 1.....	62
Gambar 172. Performansi throughput (download) viewer 7 saat menggunakan screen sharing test case 1.....	63
Gambar 173. Performansi throughput (upload) viewer 7 saat menggunakan webcam test case 1.....	63
Gambar 174. Performansi throughput (upload) viewer 7 saat menggunakan screen sharing test case 1.....	63
Gambar 175. Performansi throughput (download) viewer 7 saat menggunakan webcam test case 2.....	63
Gambar 176. Performansi throughput (download) viewer 7 saat menggunakan screen sharing test case 2.....	64
Gambar 177. Performansi throughput (upload) viewer 7 saat menggunakan webcam test case 2.....	64
Gambar 178. Performansi throughput (upload) viewer 7 saat menggunakan screen sharing test case 2.....	64
Gambar 179. Performansi throughput (download) viewer 8 saat menggunakan webcam test case 1.....	64
Gambar 180. Performansi throughput (download) viewer 8 saat menggunakan screen sharing test case 1.....	65
Gambar 181. Performansi throughput (upload) viewer 8 saat menggunakan webcam test case 1.....	65
Gambar 182. Performansi throughput (upload) viewer 8 saat menggunakan screen sharing test case 1.....	65
Gambar 183. Performansi throughput (download) viewer 8 saat menggunakan webcam test case 2.....	65

Gambar 184. Performansi throughput (download) viewer 8 saat menggunakan screen sharing test case 2.....	66
Gambar 185. Performansi throughput (upload) viewer 8 saat menggunakan webcam test case 2.....	66
Gambar 186. Performansi throughput (upload) viewer 8 saat menggunakan screen sharing test case 2.....	66
Gambar 187. Performansi throughput (download) viewer 9 saat menggunakan webcam test case 1.....	66
Gambar 188. Performansi throughput (download) viewer 9 saat menggunakan screen sharing test case 1.....	67
Gambar 189. Performansi throughput (upload) viewer 9 saat menggunakan webcam test case 1.....	67
Gambar 190. Performansi throughput (upload) viewer 9 saat menggunakan screen sharing test case 1.....	67
Gambar 191. Performansi throughput (download) viewer 9 saat menggunakan webcam test case 2.....	67
Gambar 192. Performansi throughput (download) viewer 9 saat menggunakan screen sharing test case 2.....	68
Gambar 193. Performansi throughput (upload) viewer 9 saat menggunakan webcam test case 2.....	68
Gambar 194. Performansi throughput (upload) viewer 9 saat menggunakan screen sharing test case 2.....	68
Gambar 195. Performansi throughput (download) viewer 10 saat menggunakan webcam test case 1.....	68
Gambar 196. Performansi throughput (download) viewer 10 saat menggunakan screen sharing test case 1.....	69
Gambar 197. Performansi throughput (upload) viewer 10 saat menggunakan webcam test case 1.....	69
Gambar 198. Performansi throughput (upload) viewer 10 saat menggunakan screen sharing test case 1.....	69
Gambar 199. Performansi throughput (download) viewer 10 saat menggunakan webcam test case 2.....	69

Gambar 200. Performansi throughput (download) viewer 10 saat menggunakan screen sharing test case 2.....	70
Gambar 201. Performansi throughput (upload) viewer 10 saat menggunakan webcam test case 2.....	70
Gambar 202. Performansi throughput (upload) viewer 10 saat menggunakan screen sharing test case 2.....	70

## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT atas berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul “IMPLEMENTASI VIDEO CONFERENCE ONE-TO-MANY BERBASIS TEKNOLOGI WEBRTC MENGGUNAKAN MEDIA SERVER UNTUK PEMBELAJARAN ONLINE” sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin. Sholawat serta salam kepada nabi Muhammad SAW yang telah menunjukkan dan mengajarkan akhlak mulia sehingga didapatkan kenyamanan dan keramahan dalam berhubungan dengan orang disekitar. Dengan segala kerendahan hati, penulis menyadari bahwa dalam penyusunan dan penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir ini. Sehingga, pada kesempatan ini penulis mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Kedua Orang tua penulis, Bapak Sahbana, S.H. dan (Alm.) Ibu Rosmiati, S.Pd., M.Pd. yang selalu memberikan doa, serta memberikan kesempatan kepada penulis untuk menjadi manusia yang dapat merasakan pendidikan formal serta mendukung setiap aktivitas di luar hal tersebut yang membuka wawasan, sudut pandang yang beragam serta belajar berempati pada orang lain dalam kehidupan yang penulis jalani.
2. Untuk kedua adik penulis Hanna Nabila Salsabila dan Hanni Nabila Syahrani yang selalu memberikan bantuan, dukungan, semangat dan mendengarkan keluh kesah sehingga penulis dapat menyelesaikan skripsi.
3. Bapak Dr. Eng. Muhammad Niswar, S.T., M.IT. selaku pembimbing I, yang senantiasa memberikan saran-saran serta bantuan selama proses pengambilan data hingga selesainya sistem ini dibuat, dan Bapak Iqra Aswad, S.T., M.T. selaku pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran, semangat dan perhatian yang luar biasa dalam membimbing penulis menyusun tugas akhir ini.

4. Bapak Prof. Dr. Ir. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membimbing penulis selama masa perkuliahan.
5. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu dan memberikan banyak ilmu serta dukungan selama masa perkuliahan.
6. Kanda-kanda senior dan junior terkhusus kanda Hermawan Safrin, S.T., dan Reza Arisandy Safruddin telah memberikan *insight* dan bantuannya dalam perancangan tugas akhir ini.
7. Rekan-rekan penulis Muh. Raedi Radifan, S.T., Tuti Amalia, S.T., Andi Amelia Ramadanti, Muh. Fathin Abdillah Halik, S.T., Ayu Lestari Ramadani, S.T., Cici Purnamasari, S.T., Asri Oktianawati, S.T., yang telah membantu, menemani, dan memberikan semangat untuk menyelesaikan penyusunan tugas akhir ini.
8. Rekan-rekan Teknik Informatika angkatan 2016 atas dukungan dan semangat yang diberikan selama.
9. Orang-orang berpengaruh lainnya yang tanpa sadar telah memotivasi dan membantu.

Akhir kata dengan segala kerendahan hati, penulis menyadari tugas akhir ini masih banyak kekurangan, oleh karena itu kritik dan saran sangat diharapkan demi kesempurnaan tugas akhir ini. Penyusun berharap semoga tugas akhir ini dapat bermanfaat bagi semua pihak yang membacanya.



## **BAB I PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi dan komunikasi yang pesat telah memacu munculnya berbagai aplikasi baru yang dapat digunakan untuk mendorong penemuan dan inovasi dalam berbagai aspek kehidupan, salah satunya di dunia pendidikan. Penggunaan teknologi informasi dan komunikasi dalam pendidikan menghasilkan konsep *E-learning (Electronic Learning)* yang sering disebut dengan metode pembelajaran *online*.

Pembelajaran *online* merupakan cara baru dalam proses belajar mengajar yang menggunakan media elektronik khususnya internet sebagai sistem pembelajarannya. Pembelajaran *online* merupakan dasar dan konsekuensi logis dari perkembangan teknologi informasi dan komunikasi. Pembelajaran *online* juga dikatakan suatu jenis belajar mengajar yang memungkinkan tersampainya bahan ajar ke pelajar dengan menggunakan media internet, intranet atau media jaringan komputer lain (Sugiyani dan Risdiyani, 2017).

Pembelajaran *online* dianggap sebagai sarana yang signifikan untuk membangun pendidikan yang berkelanjutan dan berkualitas, yang mulai bersaing dengan pembelajaran tradisional dan memungkinkan apa yang disebut "sekolah virtual" untuk menawarkan berbagai macam spesialisasi. Ada tiga cara utama untuk melakukan pembelajaran jarak jauh, termasuk CATV, *video conference* dan *IP network*. Namun semua teknologi yang disebutkan di atas bergantung pada aplikasi klien (Zhang et al., 2018).

Sementara itu, beberapa *platform* pembelajaran *online* yang memberikan program sertifikasi, menawarkan perkuliahan "*as-a-service*" untuk tujuan pelatihan (misalnya Cisco, Microsoft, VMware), tetapi kebanyakan dilakukan dengan metode *self-paced* (pembelajaran mandiri), sehingga mereka kehilangan keuntungan *real-time*. Sebagian besar penyedia layanan pembelajaran memiliki penerapan dan metode sendiri untuk menerapkan pelajaran laboratorium yang dipersonalisasi sehingga sulit diintegrasikan dengan konten kursus dari penyedia layanan pembelajaran lainnya (Zhang et al., 2018).

Di sisi lain, dalam pelaksanaan pembelajaran online yang menggunakan media aplikasi video conference banyak mengkonsumsi *resources* sehingga menguras kuota internet yang digunakan oleh pelajar maupun pengajar serta penggunaan CPU dan RAM yang besar membuat aplikasi lambat sehingga mempengaruhi *user experience* .

Untuk mengatasi permasalahan ini, penulis mengusulkan solusi dengan menerapkan teknologi WebRTC (*Web Real-Time Communications*) menggunakan media server untuk mengintegrasikan komunikasi *real-time* yang ramah *resource*. Platform pembelajaran *online* menjadi mudah diakses dari *browser* tanpa perlu *plug-in* tambahan atau aplikasi klien khusus apapun, semua komunikasi *back-end* berjalan di *cloud* (Zhang et al., 2018).

WebRTC dapat dengan mudah mengintegrasikan komunikasi *real-time* dengan sumber daya yang sangat besar pada tingkat web browser. Teknologi webRTC menyediakan beberapa fungsi untuk berkomunikasi karena banyak fitur yang dapat dimanfaatkan, seperti pertukaran informasi melalui suara, video dan teks. Penggunaan webRTC sangat berbeda dengan web biasanya, karena mampu berkomunikasi tanpa harus melakukan instalasi apapun untuk dapat digunakan. WebRTC memiliki keamanan yang baik karena sudah dilengkapi dengan proses enkripsi pada media audio dan video serta *signalling* yang digunakan untuk mengkoordinasikan komunikasi dan mengirim pesan (Muchma dkk, 2016). Selain itu, media server sebagai *third party modules* yang berfungsi untuk melakukan pengolahan data audio dan video pada aplikasi yang menerapkan teknologi webRTC digunakan untuk membantu komputer klien yang memiliki keterbatasan *resource* saat melakukan proses tukar menukar data antara klien hingga proses *decoding* dan *encoding* video / audio (Aswad dkk, 2017).

Untuk meningkatkan kualitas pembelajaran, beberapa penelitian telah dilakukan untuk melakukan implementasi *platform* pembelajaran *online* yang mengintegrasikan teknologi WebRTC untuk mencapai pembelajaran yang interaktif dan fleksibel. Pengembangan aplikasi *whiteboard online* berbasis web dapat membantu meningkatkan layanan pembelajaran *online* yang interaktif. Penerapan *whiteboard online* memiliki beberapa dampak pada pembelajaran, antara lain meningkatkan keterlibatan pelajar di dalam kelas, memotivasi pelajar,

dan meningkatkan semangat belajar. Seperti halnya papan tulis di dalam kelas, aplikasi *whiteboard online* memungkinkan pendidik untuk menulis dalam teks, menggambar diagram, sketsa dan model secara interaktif. Pendekatan ini mampu menutupi beberapa kekurangan dari fasilitas *chat* dan pemberian *E-Book* ajar yang ada, terutama yang berkaitan dengan penjelasan spontan melalui tulisan tangan (Prasetya & Ashar, 2017). Selain, itu video pembelajaran yang direkam dapat membantu para pelajar yang kesulitan dalam memahami materi pembelajaran di kelas mendapatkan waktu tambahan untuk memproses materi yang telah diajarkan. rekaman video pengajaran menjadi salah satu media pembelajaran paling kuat yang menangkap dan mendistribusikan informasi sekaligus memberikan fleksibilitas pada lingkungan belajar yang mendukung di mana pelajar dapat lebih memahami dan menyimpan informasi, juga memungkinkan mereka untuk menonton kembali konten, memberikan penjelasan tambahan dan memungkinkan penyelesaian pekerjaan rumah yang lebih aktif (Sablić et al., 2020). Disinilah fitur *audio*, *video*, dan *screen recording* dengan menggunakan teknologi webRTC diimplementasikan untuk meningkatkan kualitas pembelajaran *online*.

Oleh karena itu, penulis mengangkat penelitian sesuai dengan judul **“Implementasi *Video Conference One-To-Many* Berbasis Teknologi WebRTC Menggunakan Media Server untuk Pembelajaran *Online*”**.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana membuat sistem pembelajaran *online* dengan mengimplementasikan *video conference one-to-many* berbasis teknologi WebRTC menggunakan media server?
2. Bagaimana kinerja teknologi webRTC menggunakan Kurento Media Server dalam pembelajaran *online*?

### 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah pada sub-bab sebelumnya maka dapat dibuat tujuan sebagai berikut:

1. Membangun sistem pembelajaran *online* dengan mengimplementasikan *video conference one-to-many* berbasis teknologi WebRTC menggunakan media server.
2. Untuk mengetahui kinerja dari teknologi webRTC menggunakan Kurento Media Server dalam pembelajaran *online*.

### 1.4 Manfaat Penelitian

Dengan dilakukannya penelitian ini, manfaat yang diharapkan dari penelitian ini adalah:

1. Memberikan media pembelajaran yang fleksibel dan interaktif pada pembelajaran *online*.
2. Sistem ini dapat diterapkan pada instansi-instansi pendidikan.

### 1.5 Batasan Masalah

Yang menjadi batasan masalah dari penelitian ini adalah sebagai berikut:

1. Kelas virtual yang dibangun menggunakan Bootstrap dan NodeJS.
2. Website yang dibangun berupa *Single Page Application*.
3. Fitur yang dimiliki pada kelas virtual ini adalah *chat*, *video conference*, dan *screen sharing*.
4. Bentuk dari kelas virtual yang dimaksud adalah *one-to-many*.
5. Kinerja yang akan diukur adalah penggunaan CPU, penggunaan RAM, dan *throughput* terhadap jumlah *client* yang terkoneksi dalam sebuah kelas virtual.
6. Perangkat yang digunakan dalam pengujian aplikasi sejumlah 15 unit komputer dan sebuah laptop.

Tempat dilaksanakan tugas akhir ini adalah Laboratorium *Ubiquitous Computing and Networking* Lab Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

## **BAB II TINJAUAN PUSTAKA**

### ***2.1 Video Conference***

Menurut Fitriansyah (2020), Video conferencing adalah teknologi yang memungkinkan pengguna yang berada pada lokasi yang berbeda untuk mengadakan pertemuan tatap muka tanpa harus pindah ke satu lokasi bersama. Teknologi ini sangat nyaman serta praktis bagi para penggunanya yang bergerak di bidang bisnis, perusahaan maupun personal dari berbagai kota dan negara yang berbeda.

Menurut Gough (2006), video conferencing data dibagi menjadi 3 jenis, yaitu:

1. Personal video conferencing, melibatkan dua orang yang berinteraksi satu sama lain. Dalam personal video conferencing, terdapat komunikasi yang berupa video dan audio antara dua orang yang berinteraksi. Dapat ditambahkan fitur tambahan berupa pengiriman teks, seperti yang terdapat pada kebanyakan perangkat lunak instant messaging (IM), seperti Windows Live Messenger dan Yahoo! Messenger.
2. Business video conferencing, memiliki fitur yang sama dengan personal videoconferencing ditambah dengan beberapa fitur seperti:
  - Kemampuan untuk berkomunikasi tidak hanya antara dua orang, namun bisa lebih
  - Fitur untuk berbagi file (file sharing)
  - Kemampuan untuk melakukan presentasi
  - Fasilitas whiteboard dan fitur-fitur lainnya

Business video conferencing ini membutuhkan biaya yang lebih besar dibandingkan dengan personal video conferencing, sebagai akibat dari fasilitas-fasilitas yang disediakan.

3. Web videoconferencing, yaitu video call yang terdapat pada sebuah halaman web. Biasanya web video conferencing ini digunakan pada seminar yang menggunakan web, dimana pemirsa dapat melihat video yang dikirimkan oleh pembicara seminar. Oleh karena itu, web video conferencing merupakan komunikasi satu arah, karena pemirsa tidak dapat mengirimkan videonya kepada si pengirim.

## **2.2 Web Real-Time Communication (WebRTC)**

*Web Real-Time Communication* (WebRTC) adalah teknologi yang memungkinkan aplikasi Web dan situs untuk menangkap dan memungkinkan *stream* media audio dan/atau video, serta melakukan pertukaran data antar *browser* tanpa melalui perantara. Standar yang dimiliki WebRTC memungkinkan pertukaran data dan melakukan *teleconference* secara *peer-to-peer* tanpa pengguna harus memasang *plug-ins* atau *third-party software* lainnya. Menurut Purba, dkk (2020), WebRTC adalah kemampuan browser, mobile platform, atau perangkat IoT untuk melakukan komunikasi data video dan audio secara real time

WebRTC terdiri dari beberapa *Application Programming Interface* (API) dan protokol yang saling terkait yang bekerja sama untuk memungkinkan teknologi ini (MDN Web Docs, 2021).

## **2.2 Network Address Translation (NAT)**

*Network Address Translation* (NAT) merupakan mekanisme yang tersembunyi dari akses publik terhadap *private IP address* komputer di dalam jaringan. Mekanisme NAT biasanya ditemukan di semua jenis perangkat jaringan, mulai dari *router* rumahan hingga *firewall* perusahaan yang telah mapan. dari semua kasus efeknya sama: mesin yang berada di dalam NAT tidak bisa diakses secara bebas dari luar. NAT memberikan banyak permasalahan terhadap komunikasi WebRTC: mesin di dalam jaringan dapat mengirimkan data ke luar, tetapi tidak dapat menerima data dari peserta jarak jauh yang berada di luar jaringan mesin yang mengirimkan data. Untuk memungkinkan hal ini, perangkat NAT biasanya mengizinkan untuk mengkonfigurasi *bindings* agar memungkinkan data masuk dari luar jaringan; pembuatan NAT *bindings* ini disebut sebagai NAT Traversal, juga biasa disebut “*opening ports*” (Kurento, 2021).

### **2.3 Network Address Translation Traversal (NAT Traversal)**

NAT Traversal adalah istilah umum untuk teknik membangun dan memelihara koneksi protokol internet yang melintasi NAT *gateways*, yang menjembatani konektivitas *end-to-end*. Menahan dan memodifikasi *traffic* hanya dapat dilakukan secara transparan tanpa adanya enkripsi dan otentikasi yang aman (Kurento, 2021).

### **2.4 Real-time Transport Protocol (RTP)**

*Real-time Transport Protocol* (RTP) adalah format paket standar yang dirancang untuk mentransmisikan aliran audio dan video pada jaringan *internet protocol*. Digunakan bersama dengan *RTP Control Protocol*. Transmisi menggunakan profil audio/video RTP (RTP/AVP) biasanya menggunakan SDP untuk deskripsikan parameter teknis aliran media (Kurento, 2021).

### **2.5 RTP Control Protocol (RTCP)**

*RTP Control Protocol* (RTCP) adalah protokol saudara dari RTP, yang menyediakan statistik *out-of-band* dan informasi kontrol untuk aliran RTP (Kurento, 2021).

### **2.6 Session Description Protocol (SDP)**

*Session Description Protocol* (SDP) adalah dokumen teks yang berisi parameter pada saat melakukan sesi media *streaming*. Biasanya digunakan untuk menggambarkan karakteristik aliran RTP (dan protokol terkait seperti RTSP).

SDP *Offer/Answer* adalah model negosiasi antara dua *peer* pada aliran *unicast*, di mana pengirim dan penerima berbagi kumpulan aliran media dan *codec* yang ingin mereka gunakan, bersama dengan alamat IP dan *port* yang ingin mereka gunakan untuk menerima media.

Ini adalah contoh negosiasi SDP *Offer/Answer*. Pertama, harus ada *peer* yang ingin memulai negosiasi; kita menyebutnya sebagai *offerer*. Ini menyusun Penawaran SDP berikut dan mengirimkannya ke *peer* lain yang kita sebut *answerer*.

```
v=0
o=- 0 0 IN IP4 127.0.0.1
s=Example sender
c=IN IP4 127.0.0.1
t=0 0
m=audio 5006 RTP/AVP 96
a=rtpmap:96 opus/48000/2
a=sendonly
m=video 5004 RTP/AVP 103
a=rtpmap:103 H264/90000
a=sendonly
```

Gambar 1. Contoh SDP offer

Setelah menerima Penawaran itu, *answerer* mempelajari parameter yang diminta oleh *offerer*, memutuskan apakah parameter tersebut dapat dipenuhi, dan menyusun jawaban SDP yang sesuai yang akan dikirim kembali ke *offerer*:

```
v=0
o=- 3696336115 3696336115 IN IP4 192.168.56.1
s=Example receiver
c=IN IP4 192.168.56.1
t=0 0
m=audio 0 RTP/AVP 96
a=rtpmap:96 opus/48000/2
a=recvonly
m=video 31278 RTP/AVP 103
a=rtpmap:103 H264/90000
a=recvonly
```

Gambar 2. Contoh SDP answer

Jawaban SDP adalah langkah terakhir dari model SDP *Offer/Answer*. Dengan itu, *answerer* menyetujui beberapa parameter yang diminta oleh *offerer*, tetapi tidak semua.

Pada contoh diatas, `audio 0` berarti *answerer* menolak aliran audio yang ingin dikirim oleh *offerer*; juga, ia menerima aliran video, dan `a=recvonly` memberitahukan bahwa *answerer* akan secara eksklusif bertindak sebagai penerima, dan tidak akan mengirim aliran apa pun kembali ke *peer* lainnya (Kurento, 2021).

## 2.7 Kurento Media Server

Kurento Media Server (KMS) adalah paket server multimedia yang dapat digunakan untuk mengembangkan aplikasi video canggih untuk platform WebRTC. Ini adalah proyek Sumber Terbuka, dengan kode sumber dirilis di bawah persyaratan Lisensi Apache Versi 2.0 dan tersedia di GitHub.



Karakteristik yang paling menonjol dari Kurento adalah sebagai berikut:

- *Modular Pipelines*

Kurento menyediakan blok bangunan seperti pengirim & penerima WebRTC dan RTP, mixer audio/video, perekaman media, dan banyak lagi. Elemen Media ini adalah objek mandiri yang memiliki kemampuan media tertentu; sangat mudah dibuat dengan memasukkan, mengaktifkan, atau menonaktifkannya kapan saja, bahkan saat media sudah melakukan *streaming*.

- *Built-in Modules*

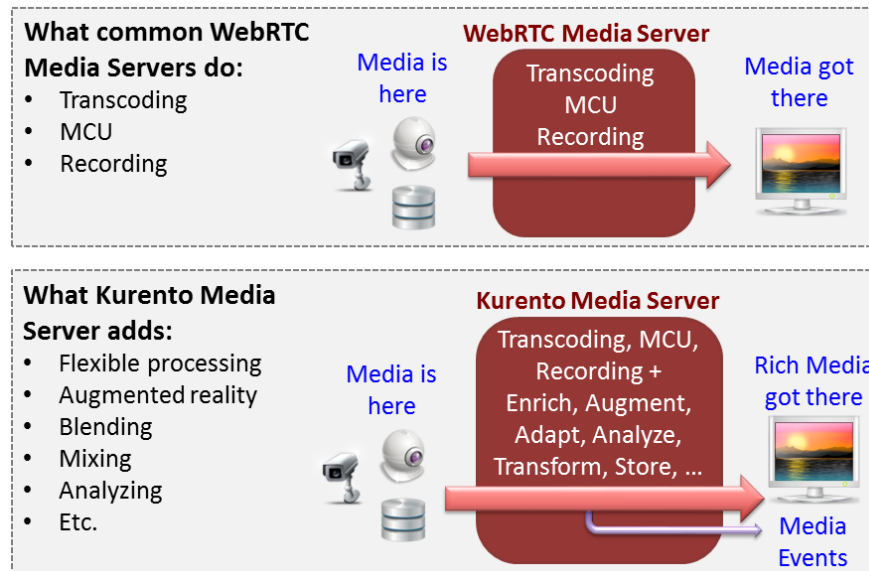
Kurento memberikan berbagai macam media elemen sebagai bagian dari API-nya, sebagai contoh:

- *WebRtcEndpoint* dapat mengirim dan menerima aliran media WebRTC.
- *PlayerEndpoint* dapat digunakan untuk menggunakan media dari RTSP, HTTP, atau sumber lokal.
- *RecorderEndpoint* dapat menyimpan aliran media ke sistem file lokal atau jarak jauh.
- *FaceOverlayFilter* adalah contoh *Computer Vision* sederhana yang mendeteksi wajah orang di streaming video, untuk menambahkan gambar overlay di atasnya.

Kurento Media Server (KMS) dapat digunakan dalam model WebRTC Media Server, untuk memungkinkan transmisi, pemrosesan, perekaman, dan pemutaran media. KMS dibangun di atas perpustakaan multimedia GStreamer, dan menyediakan fitur-fitur berikut:

- Protokol streaming jaringan, termasuk HTTP, RTP, dan WebRTC.
- Komunikasi grup (baik fungsi MCU dan SFU) mendukung pencampuran media dan perutean/pengiriman media.
- Dukungan umum untuk filter yang menerapkan algoritma *Computer Vision* dan *Augmented Reality*.
- Penyimpanan media yang mendukung operasi penulisan untuk WebM dan MP4 dan memutar dalam semua format yang didukung oleh GStreamer.

- Transcoding media otomatis antara salah satu codec yang didukung oleh GStreamer, termasuk VP8, H.264, H.263, AMR, OPUS, Speex, G.711, dan lainnya.



Gambar 3. Kelebihan kurento media server  
(Sumber: <https://doc-kurento.readthedocs.io>)

Untuk menerapkan komunikasi *one-to-many* yang dilengkapi dengan fitur *recording* dapat menggunakan *Media Pipeline* yang disusun oleh 1+N *WebRtcEndpoints* dan juga *RecorderEndpoint*. Komputer presenter dapat mengirimkan *stream* kepada pengguna yang berperan sebagai penerima. Penerima dikonfigurasi dalam mode *receive-only*. *Pipeline* media yang diimplementasikan diilustrasikan pada gambar berikut:

## 2.8 Virtual Machine

*Virtual machine* (VM) adalah program perangkat lunak atau sistem operasi virtual yang bisa digunakan pada sebuah perangkat keras bersamaan dengan OS asli perangkat tersebut. Penyedia layanan VM biasa disebut dengan *Hypervisor*. *Hypervisor* menangani manajemen VM pada sebuah host. Kelebihan saat kita menggunakan VM antara lain:

- Fleksibel dan efisien, pengguna dapat menggunakan berbagai macam sistem operasi untuk kebutuhan lain, di dalam satu *device*.

- Lebih aman, Meskipun *virtual machine* berbagi sumber daya dengan komputer fisik, namun *virtual machine* tetap terisolasi. Pemisahan tersebut menciptakan lingkungan yang lebih aman untuk menjalankan aplikasi.
- Enkapsulasi, VM bertindak sebagai perangkat lunak tunggal, sehingga mudah dikelola. VM dapat memindahkan dan menyalin *virtual machine* dari satu lokasi ke lokasi lain, menyimpan di media penyimpanan apapun seperti file perangkat lunak lainnya.
- Mudah di *maintenance*, Berbeda dengan komputer fisik, VM lebih mudah dikelola. Biaya perawatan VM pun jauh lebih murah.

## 2.9 Docker

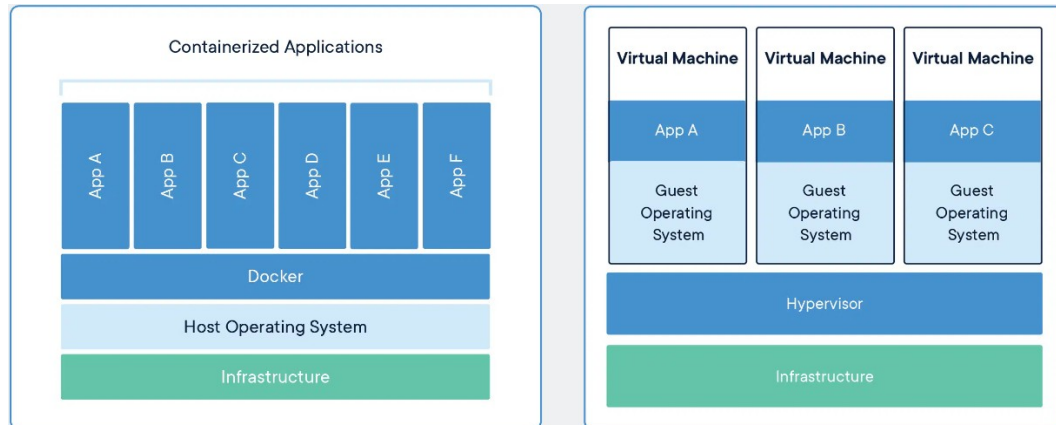
Docker adalah platform perangkat lunak yang memungkinkan Anda membuat, menguji, dan menerapkan aplikasi dengan cepat. Docker mengemas perangkat lunak ke dalam unit standar yang disebut container yang memiliki semua yang diperlukan perangkat lunak agar dapat berfungsi termasuk pustaka, alat sistem, kode, dan waktu proses. Dengan menggunakan Docker, kita dapat dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode kita akan berjalan (Docker.com).

## 2.10 Container

*Container* adalah unit perangkat lunak standar yang mengemas kode dan semua ketergantungannya sehingga aplikasi berjalan dengan cepat dan andal dari satu lingkungan komputasi ke lingkungan lainnya. *Docker Container image* adalah paket perangkat lunak yang ringan, berdiri sendiri, dan dapat dieksekusi yang mencakup semua yang diperlukan untuk menjalankan aplikasi: kode, *runtime*, alat sistem, pustaka sistem, dan pengaturan (Docker.com).

Container image menjadi container saat runtime, pada contoh kasus saat menggunakan Docker Container, *image* menjadi *container* ketika *container* dijalankan di atas *Docker Engine*. Tersedia pada sistem operasi linux dan windows, membuat kontainerisasi perangkat lunak akan selalu berjalan dengan hasil yang sama, apapun infrastrukturnya. Container mengisolasi perangkat lunak dari *environment* komputer host dan memastikan akan bekerja seragam meskipun ada perbedaan antara *environment development* dan *staging*.

Container dan VM memiliki manfaat isolasi dan alokasi sumber daya yang serupa, tetapi fungsinya berbeda karena *container* memvirtualisasikan sistem operasi, bukan perangkat keras. *Container* lebih portabel dan efisien.



Gambar 4. Perbedaan teknologi container dan virtual machine  
(Sumber: [www.docker.com](http://www.docker.com))

*Container* merupakan abstraksi pada lapisan aplikasi yang mengemas kode dan dependensi secara bersamaan. Beberapa container dapat berjalan di mesin yang sama dan berbagi kernel OS dengan container lain, masing-masing berjalan sebagai proses yang terisolasi di ruang pengguna. *Container* mengambil lebih sedikit ruang penyimpanan daripada VM (*container image* biasanya berukuran puluhan MB), dapat menangani lebih banyak aplikasi dan membutuhkan lebih sedikit daripada *virtual machine* dan sistem operasi.

*Virtual machine* (VM) adalah abstraksi dari perangkat keras fisik yang mengubah satu server menjadi banyak server. *Hypervisor* memungkinkan beberapa VM berjalan di satu mesin. Setiap VM memiliki salinan lengkap dari sistem operasi, aplikasi, binari dan pustaka yang diperlukan – menghabiskan puluhan GB. VM juga bisa lambat saat *boot*.

Container dan VM yang digunakan bersamaan memberikan banyak fleksibilitas dalam penyediaan layanan dan manajemen aplikasi.

## 2.11 Docker Compose

Compose adalah alat untuk mendefinisikan dan menjalankan aplikasi Docker multi-container. Dengan Compose, kita menggunakan file YAML untuk mengkonfigurasi layanan aplikasi (Docker.com). Kemudian, dengan satu perintah

dapat membuat dan memulai semua layanan dari konfigurasi yang telah didefinisikan.

Compose berfungsi di semua lingkungan: *production*, *staging*, *development*, *testing*, serta alur kerja *Continuous Integration*. Docker compose juga memiliki perintah untuk mengelola seluruh *lifecycle* aplikasi:

- Memulai, menghentikan, dan membangun kembali *service*
- Melihat status dan log dari *service* yang sedang berjalan
- Mempersingkat perintah panjang dan banyak pada sebuah *service*

Fitur utama Docker Compose yang membuatnya menjadi lebih efektif ketika kita menggunakannya adalah:

- Dapat membuat banyak lingkungan yang saling terisolasi dalam sebuah komputer
- Dapat menyimpan data saat container dibuat
- Mendukung *environment variable* dan komposisinya pada *environment* komputer yang berbeda-beda.

## 2.12 NetBalancer

Menurut Kurniawan, dkk (2017), NetBalancer adalah sebuah aplikasi yang digunakan untuk memantau trafik jaringan dalam bentuk chart. Aplikasi ini juga bisa digunakan untuk manajemen *bandwidth* jaringan yaitu mengatur bagian *bandwidth* yang akan diberikan kepada tiap-tiap *ip address*.