

Tema Penelitian: Teknologi Informasi dan Komunikasi

**LAPORAN AKHIR
RISET UNGGULAN UNHAS (RUNAS)
“TERAPAN”**



JUDUL PENELITIAN

**APLIKASI KELAS VIRTUAL BERBASIS WEB-BASED REALTIME
COMMUNICATION (WEBRTC) SEBAGAI MEDIA PEMBELAJARAN JARAK JAUH
UNTUK PEMERATAAN PENDIDIKAN DI DAERAH**

TIM PENGUSUL:

Dr.Eng. Muhammad Niswar, ST, M.Info.Tech NIDN: 0022097301 Ketua

Amil Ahmad Ilham, ST., M.IT., Ph.D NIDN: 0010107302 Anggota-1

Dr.Eng. Wardi, ST, M.Eng NIDN: 0028087203 Anggota-2

Dr. Hendra, S.Si., M.Kom NIDN: 0002017604 Anggota-3

**FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
2018**

**HALAMAN PENGESAHAN
RISET UNGGULAN UNHAS (RUNAS)**

Judul Penelitian : Aplikasi Kelas Virtual berbasis Web-based Real-Time Communication (WebRTC) sebagai Media Pembelajaran Jarak Jauh untuk Pemenuhan Pendidikan di Daerah

Tema Penelitian : Teknologi Informasi dan Komunikasi

Jenis Penelitian : Terapan

Luaran Penelitian :

1. Publikasi Jurnal International ber-Scopus dan tercatat Scimago
2. Prototipe

Ketua Peneliti:

- a. Nama Lengkap : Dr-Eng. Muhammad Niswar, ST, M.Info.Tech
- b. NIDN : 0022097301
- c. Jabatan Fungsional : Lektor Kepala
- d. Fakultas/Jurusan : Teknik/Informatika
- e. Telpon/Faks/E-mail : 0852 5642 8572
- f. Alamat surel (e-mail) : niswar@unhas.ac.id

Anggota Peneliti 1:

- a. Nama Lengkap : Amil Ahmad Ilham
- b. NIDN : 0010107302
- c. Fakultas/Prodi : Teknik/Teknik Informatika

Anggota Peneliti 2:

- a. Nama Lengkap : Dr.Eng. Wardi, ST, M.Eng
- b. NIDN : 0028087203
- c. Fakultas/Prodi : Teknik/Teknik Elektro

Anggota Peneliti 3:

- a. Nama Lengkap : Dr. Hendra, S.Si., M.Kom
- b. NIDN : 0002017604
- c. Fakultas/Prodi : MIPA/Ilmu Komputer

Lama Penelitian : 8 Bulan

Biaya Penelitian


- a. Diusulkan ke Unhas : Rp. 115.000.000,-

Mengetahui
Dekan Fakultas Teknik.

Dr. Ing. Wahyu H. Piarah, MSME
NIP: 19600302 198609 1 001

Makassar, 23 Februari 2018

Ketua Peneliti,


Dr. Eng. M. Niswar, ST, M.Info.Tech
NIP: 19730922 199903 1 001

Menyetujui
Ketua Lembaga Penelitian dan Pengabdian Masyarakat,

Prof. Dr. Ir. Laode Asrul, MP
NIP: 19630307 198812 1 001

Daftar Isi

Lembar Pengesahan	2
Daftar Isi	3
Ringkasan	4
I. Pendahuluan	5
II. Tinjauan Pustaka	7
III. Metode	8
V. Hasil dan Pembahasan	13
VI. Kesimpulan	22
VII. Daftar Pustaka	23

RINGKASAN

Permasalahan penyelenggaraan pendidikan di daerah di Indonesia umumnya adalah karena kurangnya tenaga pendidik dan kualifikasi tenaga pendidik yang rendah. Selain itu, penerapan kurikulum di daerah yang belum sesuai standar menjadikan kesetaraan pendidikan di seluruh daerah di Indonesia belum terpenuhi. Pembelajaran jarak jauh melalui media internet merupakan salah satu solusi terhadap kesetaraan pendidikan di Indonesia. Pada kegiatan riset terapan ini, kami akan merancang dan menerapkan sebuah aplikasi kelas virtual dengan menggunakan teknologi Web based Real-Time Communication (WebRTC) sebagai media pembelajaran jarak jauh secara sinkron (interaktif). WebRTC merupakan API open source yang dibangun dari Javascript dan HTML5 yang dapat melakukan pertukaran data, video, audio antar browser tanpa plugin. WebRTC opensource API memudahkan dalam pengembangan aplikasi dan tidak diperlukannya plugin dan instalasi aplikasi *third party* yang dapat menambah beban kerja pada *end user*. Kami akan membangun sebuah aplikasi kelas virtual berbasis WebRTC yang memiliki fitur yang mendukung pembelajaran jarak jauh seperti video/audio conferencing, chat text, live editor/presentation, file dan desktop sharing dimana semua fitur tersebut terintegrasi dalam sebuah browser. Dengan aplikasi kelas virtual berbasis WebRTC ini, seluruh pelajar di daerah-daerah memiliki kesempatan untuk dapat mengikuti pembelajaran dimana dan kapan saja tanpa batasan lokasi melalui media internet sehingga kesetaraan pendidikan di Indonesia tercapai.

Keyword : *WebRTC, Kelas Virtual, Pembelajaran Jarak Jauh, Javascript, HTML5.*

BAB 1. PENDAHULUAN

Permasalahan penyelenggaraan pendidikan di daerah terpencil di Indonesia umumnya adalah karena kurangnya tenaga pendidik dan kualifikasi tenaga pendidik yang rendah. Selain itu, penerapan kurikulum di daerah yang belum sesuai standar menjadikan kesetaraan pendidikan di seluruh daerah di Indonesia belum terpenuhi. Perkembangan teknologi informasi dan komunikasi (TIK) menjadikan dunia terkoneksi sehingga masyarakat di seluruh dunia dapat saling berkomunikasi tanpa batasan lokasi. Saat ini, TIK telah digunakan sebagai alat pendukung dalam proses belajar mengajar khususnya pada pembelajaran jarak jauh. Dengan dikeluarkannya peraturan kementerian pendidikan dan kebudayaan (Permendikbud) tahun 2012 mengenai pembelajaran jarak jauh. Pada pasal 3 disebutkan bahwa pembelajaran jarak jauh memiliki karakteristik terbuka, belajar mandiri, belajar tuntas dan menggunakan TIK yang menunjukkan bahwa metode e-learning tidak menutup seseorang untuk menuntun ilmu pada sebuah wilayah yang terpencil, membuat pelajar menjadi lebih terbuka pada kesempatan untuk menambah ilmu pengetahuan dari mana dan kapan saja. Adapun pada pasal 4 mengenai penyelenggaraan PJJ. Pasal 4 menyebutkan bahwa PJJ dapat diselenggarakan pada lingkup program studi dan mata kuliah. Dengan tujuan pelajar tidak terpaku akan keterbatasan wilayah dan kondisi, selama instansi melakukan PJJ atas dasar pendidikan yang dibawah oleh sebuah Mata Kuliah atau sistem sebuah Program Studi. PJJ diperbolehkan untuk dilakukan.

Saat ini, pelajar dapat mengakses materi pembelajaran seperti materi-materi pembelajaran on-line dalam bentuk elektronik file (.PDF, PPT, Doc) dan video melalui internet sehingga kesetaraan pendidikan di masyarakat dunia khususnya Indonesia dapat tercapai. Namun, metode pembelajaran konvensional tatap muka dalam kelas memberikan pengalaman dan efek pembelajaran yang lebih besar dibandingkan belajar mandiri dengan mengakses materi-materi pembelajaran on-line. Dengan berinteraksi langsung dengan pengajar, pelajar memiliki kesempatan untuk bertanya langsung dan mendapatkan jawaban secara instan. Oleh karena itu, dibutuhkan suatu media pembelajaran jarak jauh sinkron yang dapat menggantikan metode konvensional tatap muka kelas agar pelajar dapat berinteraksi langsung dengan pengajar tanpa batasan geografis. School On Internet (SOI) merupakan salah satu projek pembelajaran jarak jauh yang sinkron (interaktif) skala internasional yang memiliki mitra perguruan tinggi di Asia

khususnya di Indonesia, mitra SOI adalah ITB, UNHAS, UNIBRAW, UNSYIAH, dan UNSRAT. Dalam pelaksanaan pembelajaran jarak jauh, umumnya mitra menggunakan perangkat high-end video conferencing seperti Polycom yang cukup mahal. Selain itu, untuk file sharing materi kuliah dan komunikasi text chat menggunakan sebuah aplikasi learning management system. Oleh karena itu, kita membutuhkan sebuah teknologi dan aplikasi yang mudah dan murah dimana semua fitur pembelajaran terintegrasi dalam sebuah aplikasi sehingga aplikasi ini dapat di implementasikan pada daerah-daerah di Indonesia.

Kegiatan riset ini bertujuan merancang dan menerapkan aplikasi kelas virtual berbasis Web-based Realtime Communication (WebRTC) untuk mendukung pembelajaran jarak jauh secara interaktif agar pengajar dan pelajar dapat berinteraksi langsung melalui media internet. WebRTC dibangun dari Javascript dan HTML5 yang dikembangkan pertama kali oleh Google pada tahun 2011. Aplikasi ini merupakan sebuah *API (Application Programming Interface)* yang dapat menjadi jembatan komunikasi antar *browser* tanpa memakai *plugin*, dan tidak perlu menginstall atau mengunduh aplikasi *third-party*. Dengan adanya kelebihan dari API WebRTC ini, kami berinisiatif untuk mengembangkan sebuah aplikasi yang menggunakan WebRTC untuk melakukan *video chat, voice chat, live editor, desktop* dan *file sharing* dengan hanya menggunakan *browser* agar pembelajaran jarak jauh secara sinkron (berinteraksi langsung) antara pengajar dan pelajar dapat tercapai secara efektif dan efisien. Dengan WebRTC, masing-masing pengajar dan pelajar hanya memerlukan sebuah PC/laptop yang memiliki built-in webcam/mic dan akses internet serta pengguna tidak diberi beban untuk menginstall aplikasi-aplikasi *third-party* dan mengkonfigurasinya untuk menjalankan pembelajaran jarak jauh, melainkan cukup dengan membuka browser yang *by default* sudah terinstal pada PC/laptop mereka. Dengan kelebihan yang ditawarkan oleh API WebRTC, implementasi pembelajaran jarak jauh menjadi lebih mudah dan murah sehingga mudah dalam implementasi pada daerah-daerah di Indonesia. Adapun Target-target penelitian ini di perlihatkan pada rencana target capaian tahunan pada Tabel I berikut:

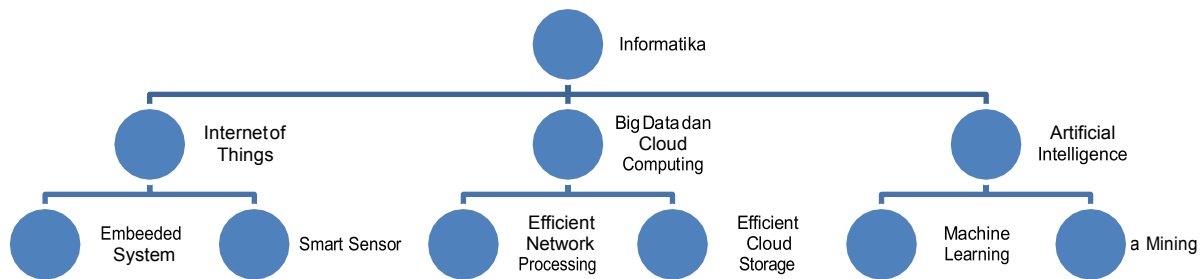
Tabel I. Rencana target capaian tahunan

No	Jenis Luaran				Indikator Capaian
	Kategori	Sub Kategori	Wajib	Tambahan	Tahun 2018 (TS)
1	Artikel ilmiah dimuat di jurnal	Internasional Bereputasi		√	Submitted

2	Model/Purwarupa/Desain/Sistem/Karya Seni/Rekayasa Sosial	√		Penerapan
3	Tingkat Kesiapan Teknologi (TKT)			Skala 4

BAB 2. TINJAUAN PUSTAKA

Salah satu bidang unggulan Universitas Hasanuddin yang ditetapkan adalah merupakan roadmap penelitian Program Studi yang berasal seluruh Fakultas di lingkungan Universitas Hasanuddin. Departemen Teknik Informatika Unhas sekarang ini memiliki tiga konsentrasi, yaitu *Big Data* dan *Cloud Computing*, *Internet of Things* (IoT), dan Kecerdasan buatan dan robotika. Jika ditinjau dari proses deduksi dan induksi, maka *road map* penelitian ilmu informatika khususnya di Universitas Hasanuddin dapat digambarkan sebagai berikut:



Gambar 1. Road map penelitian ilmu informatika

Penelitian sekarang ini adalah merancang dan menerapkan aplikasi kelas virtual berbasis WebRTC yang menggunakan media server yang akan di letakkan pada *cloud*. Media server ini yang akan mengatur video streaming perkuliahan agar penggunaan resource seperti bandwidth dan CPU processing server lebih efisien. Sehingga penelitian ini sangat terkait dengan bidang konsentrasi *Big Data* dan *Cloud Computing*.

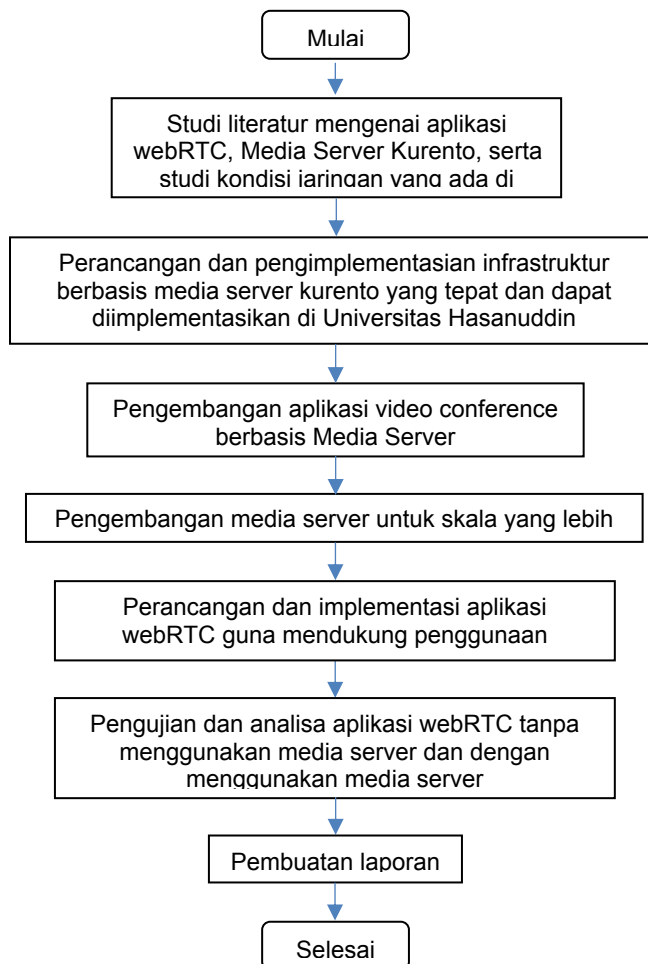
Teknologi WebRTC telah banyak digunakan dalam aplikasi seperti audio/video conferencing system (S.Yoon, et al 2016) dan file sharing (R.Dukiya, et al 2017). Namun belum banyak peneliti yang melakukan implementasi WebRTC ini untuk pembelajaran pada lingkungan dengan keterbatasan infrastruktur (low-resources setting) seperti kecepatan internet yang lambat

(narrow bandwidth) seperti pada di daerah di Indonesia. Untuk itu, penelitian ini akan menguji kinerja teknologi WebRTC di daerah low-resource settings dan melakukan kustomisasi dan adaptasi agar teknologi WebRTC ini dapat diterapkan secara efisien untuk pembelajaran jarak jauh di daerah.

BAB 3. METODE PENELITIAN

3.1. Tahapan Penelitian

Gambar 2 menggambarkan tahap penelitian dimana dimulai dengan melakukan studi literatur yaitu melakukan pengkajian yang dibutuhkan untuk penelitian ini, baik berupa jurnal ilmiah, buku referensi, bahan kuliah, maupun sumber lainnya. Selain itu untuk menunjang terlaksananya penelitian ini, maka dilakukan juga studi tentang kondisi jaringan maupun server yang ada di Universitas Hasanuddin. Selanjutnya dilakukan perancangan model infrastruktur media server

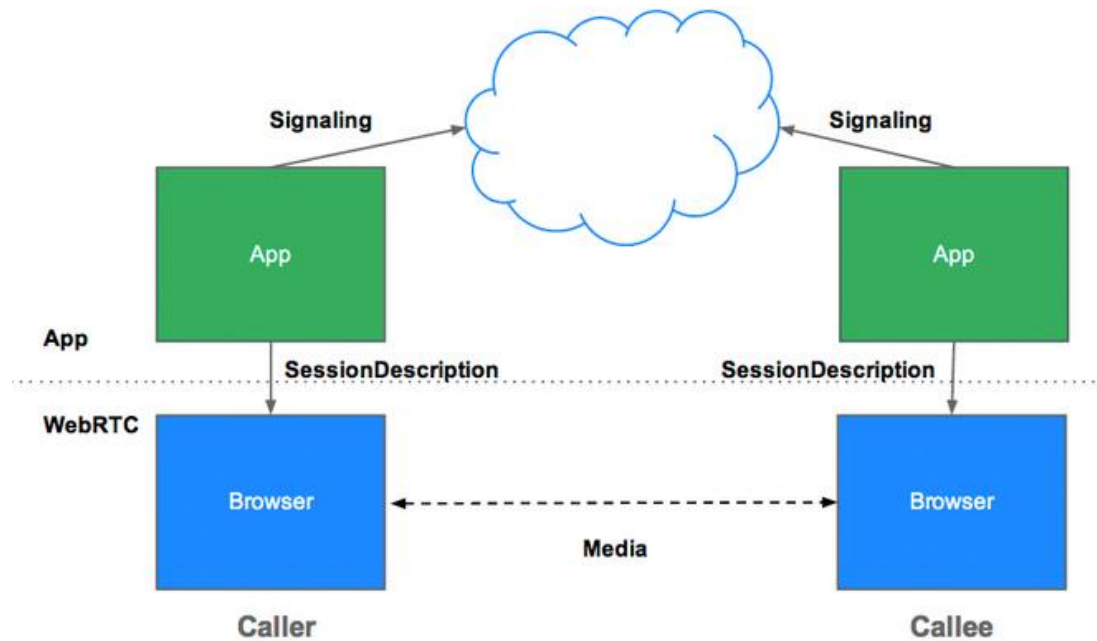


Gambar 2. Tahapan Penelitian

berbasis Kurento yang dapat diimplementasikan pada kondisi server maupun jaringan saat ini. Untuk menguji stabilitas media server, dikembangkan sebuah aplikasi webRTC berupa *video conference* berbasis media server Kurento. Aplikasi *video conference* dipilih dalam pengujian media server, karena *video conference* sendiri membutuhkan penggunaan resource yang cukup besar, baik itu berupa penggunaan bandwidth maupun *processor*. Tahap selanjutnya adalah dengan mengembangkan infrastruktur yang ada dengan melakukan penambahan beberapa media server serta penambahan beberapa modul sehingga diharapkan media server Kurento dapat digunakan untuk skala yang lebih besar. Pengembangan selanjutnya yang dilakukan adalah melakukan penambahan beberapa algoritma pada aplikasi *video conference*. Hal ini dilakukan agar *video conference* yang telah dibuat dapat memanfaatkan semua media server yang ada sehingga beban *processor*, bandwidth maupun memori tidak terpusat kepada satu server saja. Pengujian dan analisa dilakukan terhadap aplikasi *video conference* yang telah dibuat, baik itu ketika menggunakan Kurento media server maupun sebelum menggunakan media server. Tahapan terakhir yang dilakukan adalah melakukan penulisan laporan

3.2. Perancangan Sistem

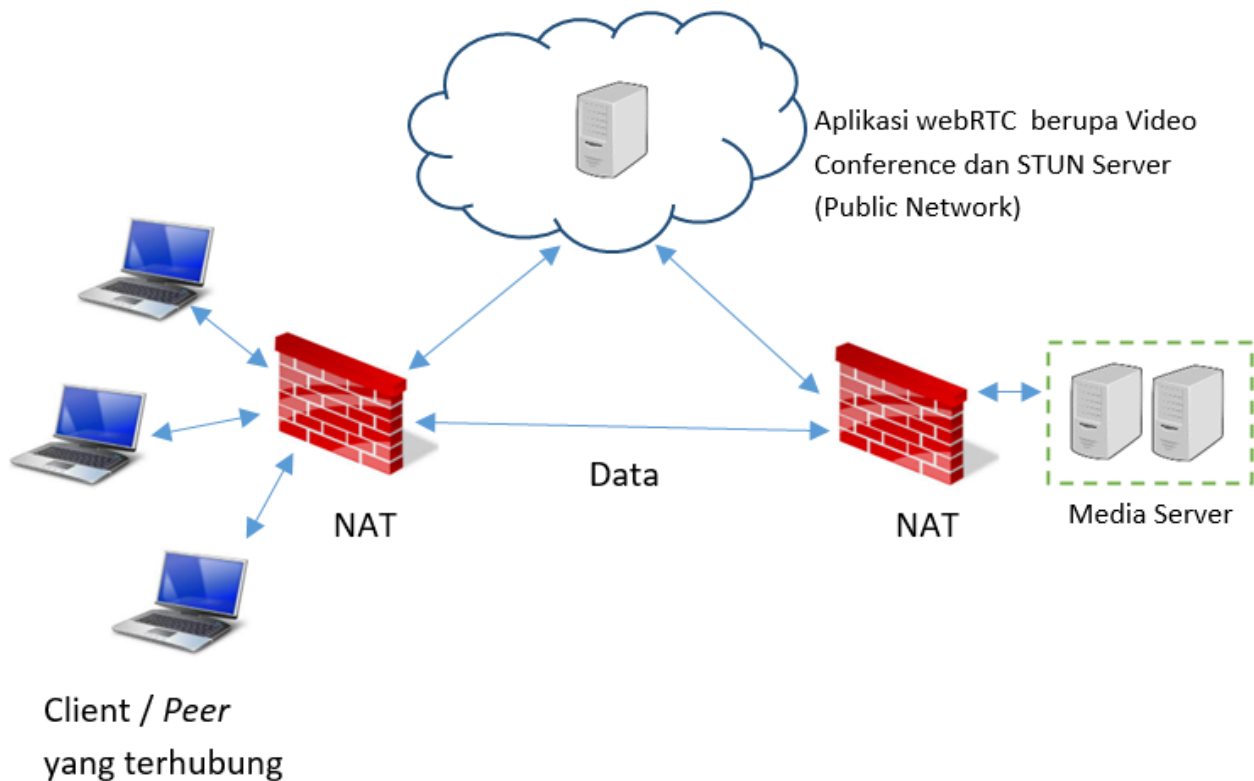
Konsep komunikasi peer to peer WebRTC dapat dilihat pada gambar 3. Arsitektur yang akan digunakan untuk menciptakan komunikasi antar browser (peers) ialah *Star*: medium N-way call. Mekanisme komunikasi streaming data arsitektur ini menunjuk satu peer yang akan menjadi initiator, dimana initiator akan mendapatkan tiap data yang dikirim oleh peer kemudian meneruskannya ke peer yang lain. Dengan arsitektur ini maka komunikasi data akan menghemat bandwidth dan penggunaan memorinya. Untuk menghubungkan antar peer terlebih dahulu peer harus bertukar informasi *session control message*, *network configuration*, dan *media capabilitiesnya*. Proses pertukaran informasi tersebut merupakan proses **signaling**, maka dari itu dibutuhkan server node.js dan modul socket.io sebagai media perantara untuk melakukan proses signaling. Peer yang terlebih dahulu melakukan koneksi ke server akan di anggap sebaga *initiator/Caller*. Jika ada peer yang baru masuk, maka server akan mengabari ke Initator, dimana initiator akan melakukan pemanggilan ke peer tersebut (Callee), kemudian Calle yang akan menjawab pemanggilan tersebut.



Gambar 3. Model Komunikasi WebRTC

Gambar 3 menunjukkan arsitektur yang akan digunakan dalam penelitian ini. Perancangan sistem yang dibuat berdasarkan kondisi jaringan di Universitas Hasanuddin saat ini dimana setiap laboratorium komputer maupun fakultas yang terhubung ke jaringan internal berada di belakang NAT. Dalam penelitian ini digunakan beberapa server yang bertugas sebagai media server dan sebuah server yang bertugas sebagai STUN sekaligus tempat berjalannya aplikasi webRTC. STUN dan webRTC berada pada server yang sama karena keduanya menggunakan resource yang cukup kecil. Hal ini dikarenakan setelah terjadi komunikasi di antara *client* dan media server, pengiriman media *streaming* tidak akan terjadi antara *client* dan webRTC tetapi antara *client* dan media server. STUN dan aplikasi webRTC membutuhkan sebuah alamat publik sehingga aplikasi webRTC dipastikan dapat diakses oleh *user* yang mungkin berada pada *private network* yang berbeda – beda. Media server dan *client* dapat berada pada *network* yang berbeda karena keduanya akan menggunakan STUN yang telah disediakan sebelumnya. STUN akan me-*reply* alamat ip maupun *port* UDP yang terbaca dari luar NAT sehingga nantinya dapat digunakan untuk saling berkomunikasi. Pada media server akan dipasang *web service* yang nantinya akan digunakan oleh

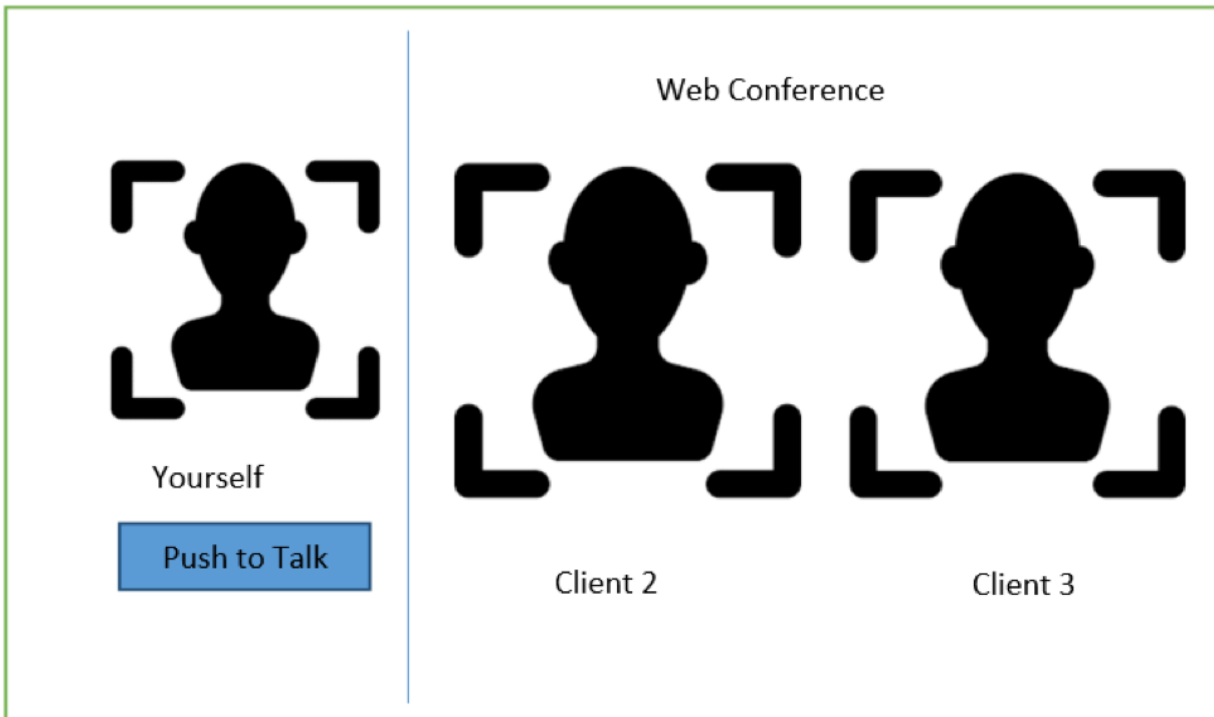
aplikasi webRTC untuk memonitor media server sehingga aplikasi webRTC tersebut dapat menentukan kapan akan menggunakan media server yang tersedia.



Gambar 4. Topologi Network

Web service dan aplikasi webRTC yang dijalankan menggunakan bahasa pemrograman javascript, html dan dijalankan dengan menggunakan node.js. Selain perancangan infrastruktur, Penelitian ini juga difokuskan pada pembuatan aplikasi berbasis webRTC berupa *video conference* guna menunjang media server yang digunakan. *Video conference* yang dibuat tidak menggunakan Kurento *group-call* API yang disediakan oleh pengembang Kurento. Hal ini dikarenakan API tersebut hanya didesain untuk menggunakan satu media server Kurento saja. Karena keterbatasan tersebut maka pada penelitian ini dikembangkan pula aplikasi *video conference* yang menerapkan konsep aplikasi *broadcasting* milik Kurento (*one-2-many*). Pada aplikasi *one-2-many* yang

disajikan oleh pengembang Kurento, terdapat beberapa *peer* / komputer dimana salah satu komputer bertindak sebagai presenter dan komputer lainnya bertindak sebagai *viewer*. Komputer yang bertindak sebagai presenter hanya akan membroadcast media yang dimilikinya. Dan komputer yang bertindak sebagai *viewer* hanya akan melihat media yang dibroadcast oleh presenter. Setiap presenter dan *viewer* akan berada pada *instance* yang sama dan berada pada *pipeline* yang sama di media server Kurento. *Video conference* yang dibangun menggabungkan fitur broadcast yang dimiliki oleh aplikasi tersebut. Dimana setiap *user* yang terhubung pada aplikasi *video conference* akan bertindak sebagai presenter yang akan membroadcast media miliknya (video dan audio) sehingga dapat ditangkap oleh client lain yang bertindak sebagai *viewer*. Selain sebagai presenter, *user* tersebut juga bertindak sebagai *viewer* yang akan melihat media yang dikirim oleh *user* lain yang juga bertindak sebagai presenter. Konsep pembuatan *instance* per presenter oleh aplikasi broadcasting milik Kurento dianggap sesuai dengan penelitian ini karena setiap presenter akan memiliki saluran komunikasi (*pipeline*) yang berbeda-beda. Hal ini cukup berbeda dengan konsep yang digunakan oleh Kurento *group-call* API dimana pembuatan *pipeline* dilakukan per kelas (*room*) sehingga pembagian pembuatan *pipeline* pada media server yang berbeda – beda menjadi lebih sulit. Selain itu diterapkan pula algoritma yang akan menentukan mekanisme pemilihan media server sebelum sebuah *pipeline* dibuat. Hal ini ditujukan agar beban yang mungkin terjadi pada media server dapat terbagi secara merata pada media – media server yang tersedia.



Gambar 5. Rancangan User Interface

Gambar 4 menunjukkan rancangan sederhana antarmuka (user interface) pada web browser. Pada user interface ini akan menampilkan semua peer yang bergabung dalam sebuah conference room.

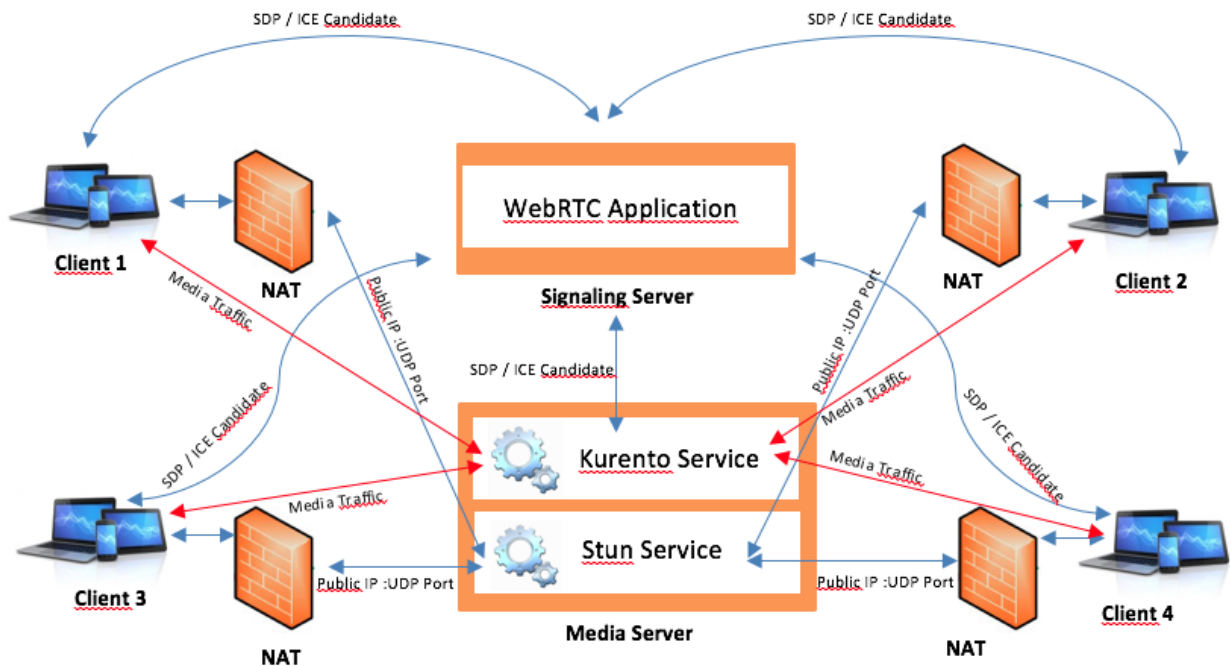
BAB 4 . HASIL DAN PEMBAHASAN

Kami telah melakukan instalasi dan konfigurasi sistem dan jaringan di lingkungan kampus UNHAS dan membuat user interface sederhana dengan menggunakan bahasa skrip pemrograman HTML5, CSS, dan Javascript.

4.1. Instalasi dan konfigurasi sistem

Kami telah melakukan instalasi aplikasi web server berbasis javascript (node.js) dan media server kurento pada virtual server pada salah satu server yang ada pada datacenter Unhas. Gambar 5 memperlihatkan sistem dan alur komunikasi video dimana setiap Client yang memanggil aplikasi WebRTC akan dihubungkan dengan Stun Server yang berada pada Media Server. Stun Server

akan memberikan respon kepada client berupa alamat IP dan UDP port yang terbaca dari belakang NAT. Alamat IP beserta port yang telah diterima oleh Client akan disimpan kedalam ICE Candidate yang nantinya akan dikirim ke Signalling Server untuk dipertukarkan dengan IP dari Media Server. Hal ini dimaksudkan agar Client dan Media Server dapat saling berkomunikasi dengan melakukan pertukaran IP terlebih dahulu. Sesi komunikasi diatur sepenuhnya oleh Aplikasi WebRTC. Semua media stream akan melalui Media Server dan dikirim ke Client lain sesuai dengan arahan Aplikasi WebRTC yang berada pada Signalling Server.

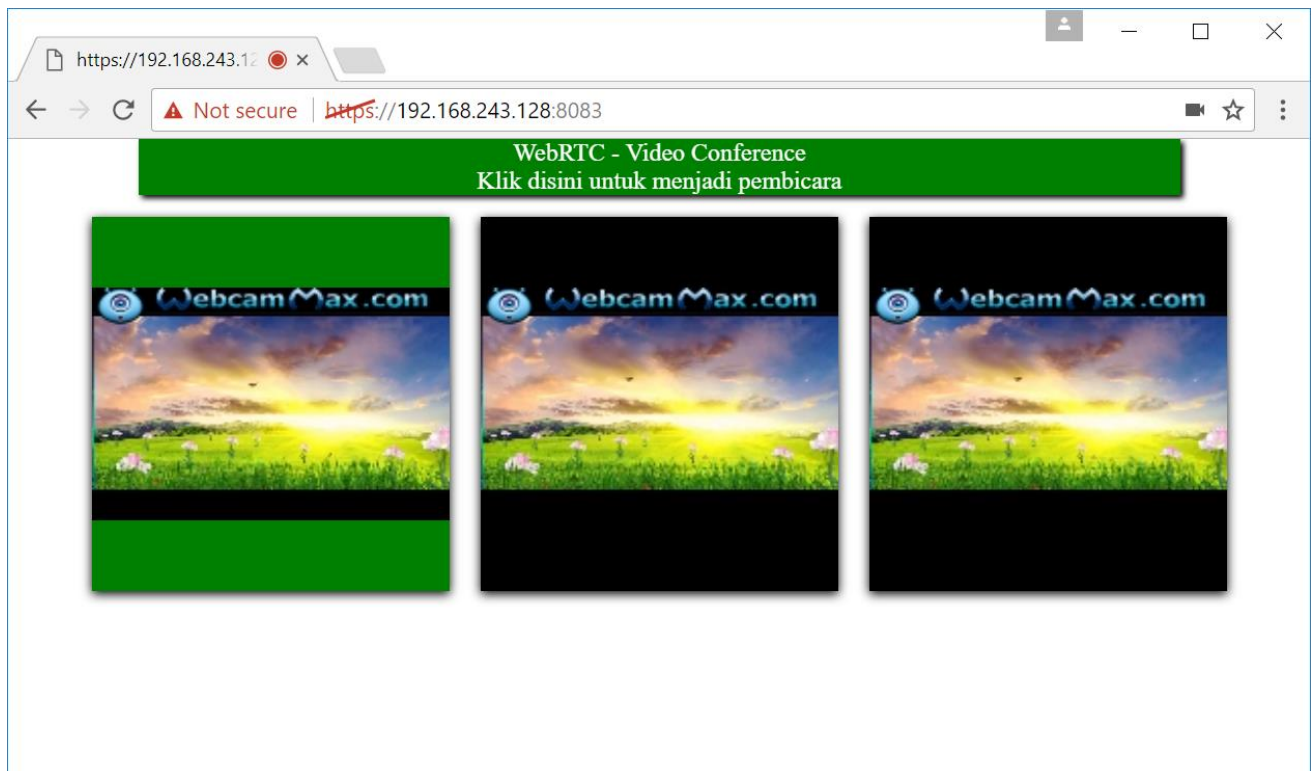


Gambar 5. Sistem dan Alur Komunikasi

4.2. Perancangan Web User Interface

Kami telah merancang web user interface dengan menggunakan dengan menggunakan bahasa skrip pemrograman HTML5, CSS, dan Javascript pada browser chrome. Gambar 6

memperlihatkan screenshot tampilan user interface menggunakan chrome. Pada user interface ini terdapat 1 sesi presenter dan 2 sesi client.



Gambar 6. Web User Interface

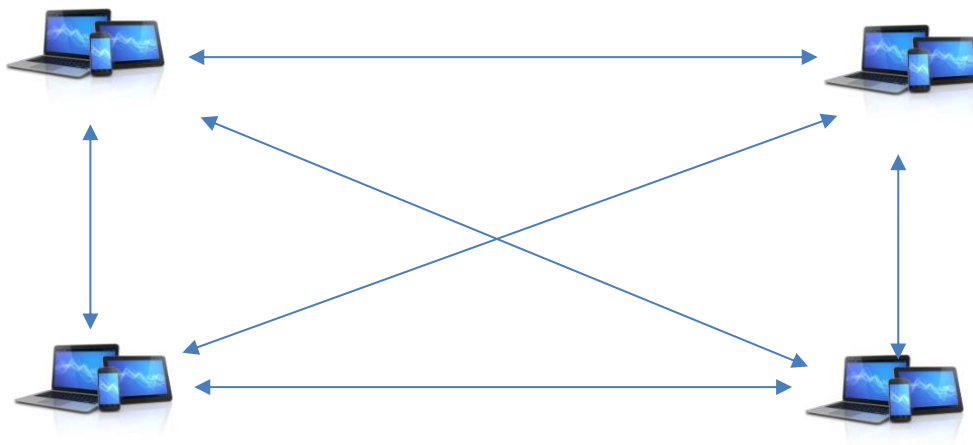
4.3. Uji Kinerja

Untuk melakukan pengujian digunakan aplikasi webRTC dengan membandingkan kinerja yang diperoleh saat menggunakan media server dan saat tidak menggunakan media server. Pengujian dilakukan dengan melakukan log terhadap peningkatan penggunaan bandwidth, processor, maupun memory dari client yang digunakan saat melakukan penambahan jumlah user / *participant*. Berikut ini hasil dari pengujian yang telah dilakukan :

a. Analisis *video conference* berbasis webRTC tanpa menggunakan media server

Pengujian pertama dilakukan dengan menggunakan sebuah server yang bertindak sebagai *signaling* sekaligus STUN. Adapun perlakuan yang diberlakukan terhadap aplikasi tersebut adalah dengan memberikan batasan – batasan media yang akan dikirim maupun diterima seperti *framerate* yang digunakan oleh webRTC sebesar 15 fps, serta lebar maksimal media sebesar 340px.

Pengujian dilakukan dengan menambahkan *participant / user* satu demi satu ke aplikasi *video conference* yang disediakan. Model komunikasi yang dilakukan untuk pengujian pertama menggunakan topologi *mesh* seperti yang ditunjukkan pada gambar dibawah ini.



Gambar 7. Model topologi mesh pada aplikasi *video conference*

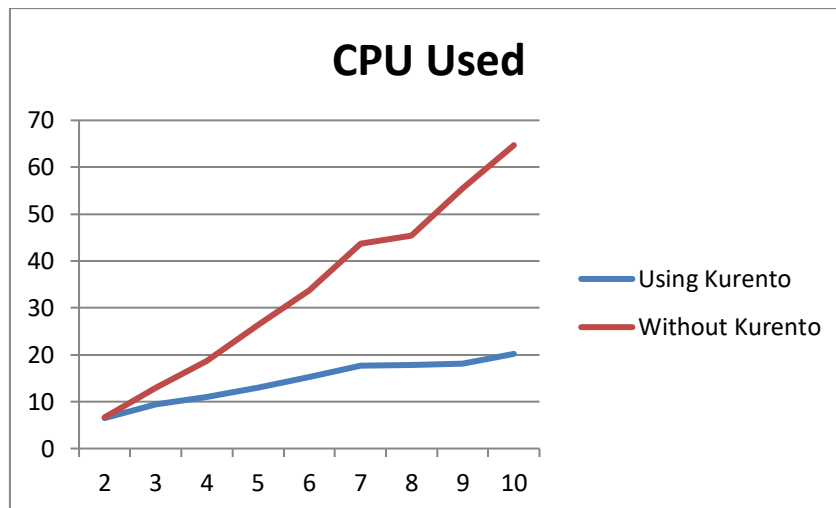
Berdasarkan penelitian yang dilakukan diperoleh hasil pengukuran sebagai berikut:

Tabel 1. Tabel peningkatan beban kerja CPU (%)

No	Jumlah Participant	CPU Used (x)	$X_n - X_{n-1}$
1	2	6.65%	-
2	3	12.94%	6.29%
3	4	18.67%	5.73%
4	5	26.29%	7.62%
5	6	33.82%	7.53%
6	7	43.69%	9.87%
7	8	47.26%	3.57%
8	9	55.55%	8.29%
9	10	64.71%	9.16%

Berdasarkan tabel diatas, maka diperoleh rata – rata peningkatan penggunaan CPU untuk setiap penambahan *1participant / user* sebesar :

$$x = \frac{6.29\%+5.73\%+7.62\%+7.53\%+9.87\%+3.57\%+8.29\%+9.16\%}{8} = \pm 7.2\%$$



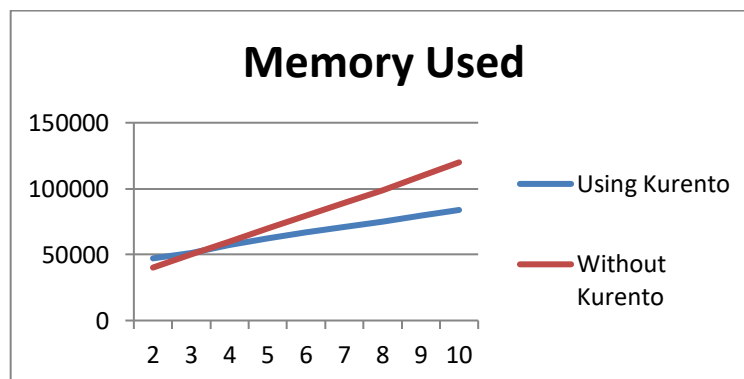
Adapun besar peningkatan penggunaan memori oleh aplikasi *video conference* untuk jumlah *user* yang berbeda dapat dilihat pada tabel berikut :

Tabel 2. Tabel peningkatan jumlah kebutuhan *memory* (bytes)

No	Jumlah <i>Participant</i>	<i>MemoryUsed</i> (m)	$m_n - m_{n-1}$
1	2	40143	-
2	3	50028	9885
3	4	59667	9639
4	5	69811	10144
5	6	79280	9469
6	7	88911	9631
7	8	98668	9757
8	9	109360	10692
9	10	119951	10591

Berdasarkan tabel diatas, dapat disimpulkan rata – rata peningkatan memori yang digunakan untuk setiap penambahan *user* sebesar ± 9976 KB.

$$m = \frac{9885+9639+10144+9469+9631+9757+10692+10591}{8} = \pm 9976 \text{ KB}$$



Peningkatan kebutuhan bandwidth untuk setiap penambahan *user* diperlihatkan pada tabel dibawah :

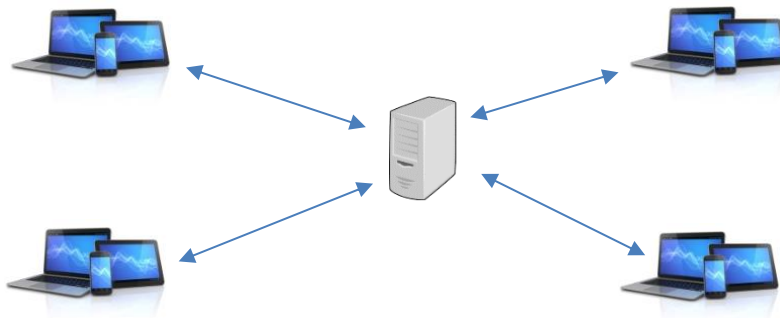
Tabel 3. Tabel peningkatan kebutuhan *bandwidth*(bps)

No	Jumlah Participant	bitsSentPerSecond (bps)	bitsReceivedPerSecond (bps)
1	2	572540	535636
2	3	1009455	1002808
3	4	1425179	1601384
4	5	2152680	2190642
5	6	2757710	2623765
6	7	3038434	2983911
7	8	3537813	3452398
8	9	4114360	4275704
9	10	4585664	4837820

b. Analisis *video conference* berbasis webRTC menggunakan media server.

Pengujian selanjutnya dilakukan dengan menggunakan infrastruktur media server yang telah dibuat sebelumnya. Perlakuan yang dilakukan sama dengan pengujian sebelumnya, dimana dilakukan pembatasan berupa lebar maksimal media sebesar 340px, *framerate* sebesar 15fps. Pengaturan ini dilakukan dengan memodifikasi file *kurento-utils.js*.

Model komunikasi pada pengujian kedua dapat dilihat pada gambar dibawah ini :



Gambar 8. Topologi aplikasi *video conference* berbasis *client-server*

Dengan menggunakan batasan yang telah ditentukan sebelumnya, diperoleh data yang menunjukkan peningkatan beban CPU seperti pada table dibawah :

Tabel 4. Tabel peningkatan penggunaan CPU.

No	Jumlah <i>Participant</i>	CPU Used (x)	$X_n - X_{n-1}$
1	2	6.5%	-
2	3	9.4%	2.9%
3	4	11%	1.6%
4	5	13.02%	2.02%
5	6	15.25%	2.23%
6	7	17.71%	2.46%
7	8	17.81%	0.1%
8	9	18.06%	0.25%
9	10	20.18%	2.12%

Berdasarkan tabel diatas, maka diperoleh rata – rata peningkatan penggunaan CPU untuk setiap penambahan 1 *participant / user* sebesar :

$$x = \frac{2.9\%+1.6\%+2.02\%+2.23\%+2.46\%+0.1\%+0.25\%+2.12\%}{8} = \pm 1.71\%$$

Adapun besar peningkatan penggunaan memori oleh aplikasi *video conference* untuk jumlah *user* yang berbeda dapat dilihat pada tabel berikut :

Tabel 5. Tabel peningkatan jumlah kebutuhan *memory* (bytes)

No	Jumlah <i>Participant</i>	<i>Memory Used</i> (m)	$m_n - m_{n-1}$
1	2	47145	-
2	3	50982	3837
3	4	57132	6150
4	5	62174	5042
5	6	66980	4806
6	7	70830	3850
7	8	74883	4053
8	9	79548	4665
9	10	83828	4280

Berdasarkan tabel diatas, dapat disimpulkan rata – rata peningkatan memori yang digunakan untuk setiap penambahan *user* sebesar ± 4585 KB.

$$m = \frac{3837+6150+5042+4806+3850+4053+4665+4280}{8} = \pm 4585 \text{ KB}$$

Peningkatan kebutuhan *bandwidth* untuk setiap penambahan *user* diperlihatkan pada tabel dibawah :

Tabel 6. Tabel peningkatan kebutuhan *bandwidth*(bps)

No	Jumlah Participant	bitsSentPerSecond (bps)	bitsReceivedPerSecond (bps)
1	2	517551	452328
2	3	452420	1142850
3	4	463827	1608481
4	5	585455	2137822
5	6	582077	2612936
6	7	496145	3301510
7	8	568138	3514259
8	9	499425	4008544
9	10	438657	4687501

BAB 5. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan infrastruktur dan aplikasi komunikasi audio/video secara realtime berbasis web. Kami menggunakan teknologi WebRTC dan Media server (kurento) untuk mengakomodasi komunikasi One-to-Many antar browser. Kami telah membuat algoritma load balancing yang mampu melakukan handover dari sebuah media server yang sibuk ke media server yang idle agar high availability tercapai.

DAFTAR PUSTAKA

Peraturan Menteri Pendidikan dan Kebudayaan (Permendikbud) Nop.24 Tahun 2012 tentang Penyelenggaraan Pembelajaran Jark Jauh.

School On Internet <http://www.soi.asia/>

S. Yoon, T. Na and H. Y. Ryu, "An implementation of web-RTC based audio/video conferencing system on virtualized cloud," *2016 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2016, pp. 133-134.

R. Dukiya, S. Sharma, S. Koppu and M. Rao, "Reliable file sharing in distributed operating system using web RTC," *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, Vellore, 2017, pp. 1-4.

G. Carlucci, L. De Cicco, S. Holmer and S. Mascolo, "Congestion Control for Web Real-Time Communication," in *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2629-2642, Oct. 2017.