SKRIPSI

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

Disusun dan diajukan oleh :

MUH NUR ALAMSYAH IBRAHIM

D42114513

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

LEMBAR PENGESAHAN SKRIPSI

IMPLEMENTASI WEB DAN SIGNALING SERVER UNTUK WEB REAL-TIME COMMUNICATION (WEBRTC) PADA RASPBERRY PI

Disusun dan diajukan oleh

MUH NUR ALAMSYAH IBRAHIM
D421 14 513

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin

Pada tanggal 20 Agustus 2021

dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,	Pembimbing Pendamping,
Dr. Eng. Muhammad Niswar, S.T., M.IT.	Iqra Aswad, S.T., M.T.
NIP. 19730922 199903 1 001	NIP. 19901128 201904 3 001

Ketua Program Studi

Dr. Amil Ahmad Ilham, S.T., M.IT
NIP. 19731010 199802 1 001

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama	: MUH NUR ALAMSYAH IBRAHIM
NIM	: D421 14 513
Program Studi	: TEKNIK INFORMATIKA
Jenjang	: S1/S2/S3

Menyatakan dengan ini bahwa karya tulisan saya berjudul

IMPLEMENTASI WEB DAN SIGNALING SERVER UNTUK WEB REAL-TIME COMMUNICATION (WEBRTC) PADA RASPBERRY PI

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

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

Gowa, 12 Oktober 2021

Yang Menyatakan

Muh Nur Alamsyah Ibrahim

ABSTRAK

Berbagai teknologi di bidang pembelajaran terus di kembangkan, seperti misalnya pembelajaran berbasis elektronik atau biasa disebut *Electronic Learning* (disingkat *E-Learning*). Salah satu teknologi yang dapat membantu model pembelajaran *e-learning* ini adalah *Web Realtime Communication* (disingkat WebRTC). Dalam pengimplementasiannya, aplikasi yang dibangun dengan menggunakan WebRTC sering mengalami kendala seperti adanya keterbatasan jumlah *client* atau partisipan yang dikarenakan oleh kemampuan dari komputer *client* yang terbatas. Pada penelitian, peneliti membuat sistem berbasis *learning management system* (*website*) pada perangkat Raspberry Pi yang diharapkan mampu memaksimalkan kegiatan *e-learning*. Dengan memanfaatkan teknologi WebRTC dan pengimplementasian arsitektur *peer to peer*, Web Server tidak akan banyak memakan *resource* karena beban kinerja WebRTC dibebankan pada *client-side* yaitu setiap *user* yang terhubung. Hasil dari penelitian ini adalah terciptanya sebuah sistem yang menerapkan teknologi WebRTC berupa *video conference* dan *chatting*. Setelah sistem telah dibangun, akan dilakukan pengukuran terkait beban kinerja WebRTC terhadap Raspberry Pi. Skenario uji yang digunakan adalah melakukan penambahan *user* satu demi satu pada aplikasi WebRTC dengan jumlah tujuh *user*. Beban kinerja yang akan diukur adalah terkait RAM dan CPU. Hasil dari pengukuran beban kinerja WebRTC pada Raspberry Pi saat melakukan *video conference* menunjukkan konsumsi CPU sebesar 0,7% - 5,3% dan konsumsi RAM 43.612,052 *kilobytes* – 53.819,128 *kilobytes*.

Kata kunci : WebRTC, Socket.io, NodeJS, PeerJS, *Peer to Peer*.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Segala puji dan syukur Penulis panjatkan ke hadirat Allah S.W.T, Tuhan Yang Maha Esa yang dengan limpahan rahmat dan hidayah-Nya sehingga tugas akhir dengan judul "*Implementasi Web dan Signaling Server Untuk Web Real-Time Communication (Webrtc) Pada Raspberry Pi"* ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin. Selanjutnya *shalawat* serta salam Penulis panjatkan pula kepada sang revolusioner sejati, Nabi Muhammad S.A.W sebagai sosok tauladan Penulis yang membawa zaman dari zaman jahiliyah ke zaman yang *In Shaa Allah* terang benderang ini, *aamiin*.

Dalam penyusunan penelitian ini disajikan hasil penelitian terkait judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, prosiding pada seminar-seminar nasional ataupun internasional, buku, dan dari berbagai situs-situs terpercaya yang ada di internet.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, mulai dari masa perkuliahan sampai dengan penyusunan tugas akhir, sangatlah sulit untuk menyelesaikan tugas akhir ini. Oleh karena itu, pada kesempatan ini Penulis menyampaikan ucapan terima kasih sedalam-dalamnya kepada :

1) Allah S.W.T karena atas semua berkat, karunia serta pertolongan-Nya yang tiada batas, yang telah diberikan kepada Penulis disetiap langkah dalam pembuatan tugas akhir ini hingga penulisan laporan skripsi ini;

2) Kedua orang tua Penulis, Bapak Ibrahim Mada (Alm) dan Ibu Hadidjah Bando yang selalu mendidik Penulis dan menjadi tempat untuk berkeluh kesah serta selalu memberikan dukungan, doa dan semangat kepada Penulis sejak kecil. Terima kasih untuk selalu ada untuk Penulis;
3) Bapak Dr.Eng. Muhammad Niswar, ST., M.IT., selaku pembimbing I dan Bapak Iqra Aswad, ST., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan Penulis dalam penyusunan tugas akhir ini;
4) Bapak Robert, Bapak Zainuddin dan Ibu Yuanita serta segenap staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu kelancaran penyelesaian tugas akhir Penulis;
5) Saudara seperjuangan Penulis Teknik 2014 yang telah menemani perjalanan penulis dalam dunia pengaderan sebagai anak teknik;
6) Keluarga besar Rectifier'14 yang juga menjadi rekan sekaligus tempat berbagi keluh dan kesah selama mengarungi dunia kampus sebagai anak teknik;
7) Teman-teman Adhyaksa *Boys* khususnya Hermawan Safrin, ST., Yakip, ST., Syarif Hidayatullah, ST., Rahmat Firman, Muh. Ardiansyah, ST, Ahmad Setiadi, ST., Arya Jaka Putra, Muh. Yusuf Ramadhan, Siswono Nasir, ST., Ulfah Rojiyyah, Al Riefqi Dasmito, ST., Abdillah Satari, ST., yang selalu menemani Penulis dalam mengarungi segala rintangan dalam dunia perkuliahan sekaligus rumah untuk kembali pulang;

8) Teman-teman pengurus Himpunan Mahasiswa Informatika Fakultas Teknik Universitas Hasanuddin Periode 2017/2018 dan Ketua-ketua lembaga tinggi se-OKFT-UH Periode 2017/2018 yang telah menemani saya dalam perjalanan organisasi;
9) Muh. Ikbal, dan Khairul Hidayat *partner* seperjuangan yang sampai saat ini masih memberikan dukungan dan semangat kepada Penulis;
10) Seluruh pihak yang tidak sempat disebutkan satu persatu yang telah banyak meluangkan tenaga, waktu dan pikiran selama penyusunan laporan tugas akhir ini.

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

Wassalam

Makassar, Juli 2021

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Nomor **Halaman**

DAFTAR TABEL

Nomor **Halaman**

BAB I

PENDAHULUAN

1.1. Latar Belakang

Internet telah menjadi salah satu cara penting untuk menyediakan sumber daya untuk penelitian dan pembelajaran bagi guru dan siswa untuk berbagi dan memperoleh informasi (Richard dan Haya, 2009). Berbagai teknologi di bidang pembelajaran terus di kembangkan. Salah satunya pembelajaran berbasis elektronik atau biasa disebut Electronic *Learning* (disingkat *E-Learning*). Menurut Jaya Kumar (2002) *E-learning* adalah pembelajaran yang menggunakan perangkat elektronik (LAN, WAN, dan Internet) untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan. *E-learning* merujuk pada pemanfaatan teknologi internet untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan keterampilan. Horton (2003) juga menyebutkan bahwa *E-learning* sebagai pembelajaran berbasis *website*.

Secara historis, ada dua model *e-learning*, yaitu pembelajaran jarak jauh dan pembelajaran berbasis komputer. Model pembelajaran berbasis komputer menggunakan komputer untuk membantu pengiriman paket multimedia yang berdiri sendiri untuk belajar dan mengajar, di mana multimedia yang digunakan terdiri dua atau lebih media, seperti teks, gambar, animasi, audio maupun video yang diakses oleh pelajar melalui komputer. Salah satu teknologi yang dapat membantu model pembelajaran *e-learning* ini adalah *Web Realtime Communication* (disingkat WebRTC). Menurut Abdulghani dan Gozali (2019), WebRTC adalah teknologi web yang memungkinkan terjadinya komunikasi antar

browser secara *real time* dengan melalui berbagai media, seperti suara, teks, dan video. WebRTC juga dapat bekerja pada web browser antar *platform* atau sistem operasi, baik komputer maupun *mobile device.* Hadirnya teknologi ini juga memberikan banyak keuntungan untuk user.

Dalam pengimplementasiannya, aplikasi-aplikasi pembelajaran jarak jauh yang dibangun dengan menggunakan WebRTC, terkhusus aplikasi yang tidak menggunakan media server sering mengalami kendala-kendala seperti adanya keterbatasan jumlah *client* atau partisipan yang dikarenakan oleh kemampuan dari komputer *client* yang terbatas. Hal ini dikarenakan seluruh pekerjaan dilakukan secara langsung oleh komputer *client* mulai dari proses tukar menukar data antara *client* hingga proses *decoding* dan *encoding* video / audio.

Dengan perkembangan teknologi yang semakin inovatif, kini konten pendidikan, metode pengajaran dan pembelajaran juga dapat mengalami perubahan besar. Kebutuhan untuk belajar dan cara orang belajar telah berubah menjadi lebih cepat dari sebelumnya, kondisi pandemi seperti saat ini juga memaksa kita untuk melakukan proses belajar mengajar secara daring. Hal ini tentunya membutuhkan perangkat pembelajaran yang mendukung salah satunya adalah aplikasi *video conference*. Aplikasi ini dibangun dan memanfaatkan Raspberry Pi untuk server untuk dapat menghemat biaya pembangunan sehingga mudah implementasikan oleh sekolah-sekolah. Oleh karena itu, penulis mengangkat penelitian sesuai dengan judul "**Implementasi Web dan Signaling Server Untuk *Web Real-Time Communication* (*WebRTC)* Pada Raspberry PI**".

1.2. Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana penerapan WebRTC dalam perancangan kelas virtual?
2. Bagaimana penerapan WebRTC *peer to peer* dengan menggunakan raspberry pi?
3. Bagaimana unjuk kerja WebRTC *peer to peer* pada raspberry pi?

1.3. Tujuan Penelitian

1. Untuk menghasilkan WebRTC *peer to peer* dengan menggunakan raspberry pi.
2. Untuk mengetahui kinerja WebRTC *peer to peer* pada raspberry pi.

1.4. Batasan Masalah

1. Implementasi WebRTC *peer to peer* menggunakan Raspberry Pi;
2. Raspberry Pi yang digunakan untuk pembuatan media server ini adalah 1 buah;
3. Perangkat yang digunakan dalam membuat system adalah laptop ASUS K401UQK dengan memori RAM 8 Gb dan *prosesor* intel Core i5 generasi ke-7.

1.5. Manfaat Penelitian

Manfaat dari tugas akhir ini adalah :

1. Bagi Peneliti

   Penelitian ini diharapkan mampu menambah pengetahuan peniliti dalam menerapkan WebRTC dalam mengembangkan *learning menegement system.*

2. Bagi Masyarakat

   Penelitian ini diharapkan mampu memberikan nilai tambah dalam pelaksanaan *e-learning.*

3. Bagi Pendidikan

   Penelitian ini diharapkan mampu menjadi acuan bagi penelitian selanjutnya terutama dalam system *e-learning.*

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan teori-teori yang menunjang percobaan yang dilakukan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi analisis kebutuhan sistem, perancangan sistem, dan scenario pengujian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian dan pembahasan penjabaran dari penelitian yang dilakukan.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian dan saran.

BAB II

TINJAUAN PUSTAKA

2.1 Raspberry Pi

Raspberry Pi adalah sebuah komputer papan tunggal (*single-board computer*) atau SBC seukuran kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation dengan tujuan untuk belajar pemrograman. Raspberry Pi pertama kali dikembangkan di laboratorium Komputer Universitas Cambridge oleh Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycrof. Mereka kemudian mendirikan yayasan Raspberry Pi bersama dengan Pete Lomas dan David Braben pada tahun 2009. Pada tahun 2012, Raspberry Pi Model B memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit. Pada bulan Februari 2016, Raspberry Pi Foundation mengumumkan bahwa mereka telah menjual 8 juta perangkat Raspi, sehingga menjadikannya sebagai perangkat paling laris di Inggris. Nama Raspberry Pi diambil dari nama buah, yaitu buah *Raspberry*, sedangkan Pi diambil dari kata Python, yaitu nama dari sebuah bahasa pemrograman. Python dijadikan bahasa pemrograman utama dari Raspberry Pi, namun tidak tertutup kemungkinan untuk menggunakan bahasa pemrograman lain pada Raspberry Pi. Keunggulan python dibanding dengan bahasa pemrograman yang lain adalah kode kode lebih mudah ditulis dan dibaca, dan juga banyak terdapat modul modul yang beragam. Adapun kekurangannya adalah tidak *realtime*, sehingga untuk akan kesusahan untuk

melakukan pekerjaan yang mempunyai *delay*, akibatnya tingkat presisi juga tidak tinggi.

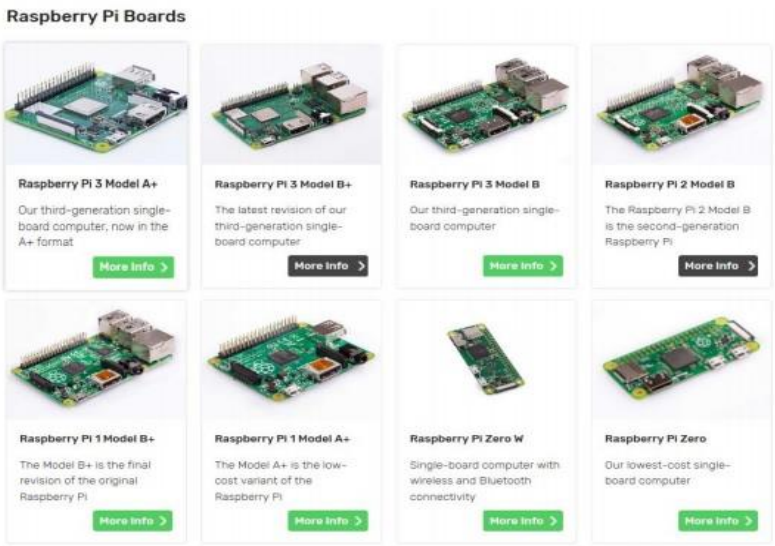
Nama Raspberry Pi diambil dari nama buah, yaitu buah Raspberry, sedangkan Pi diambil dari kata Python, yaitu nama dari sebuah bahasa pemrograman. Python dijadikan bahasa pemrograman utama dari Raspberry Pi, namun tidak tertutup kemungkinan untuk menggunakan bahasa pemrograman lain pada Raspberry Pi. Keunggulan python dibanding dengan bahasa pemrograman yang lain adalah kode kode lebih mudah ditulis dan dibaca, dan juga banyak terdapat modul modul yang beragam. Adapun kekurangannya adalah tidak *realtime*, sehingga untuk akan kesusahan untuk melakukan pekerjaan yang mempunyai *delay*, akibatnya tingkat presisi juga tidak tinggi.

Raspberry Pi memiliki komponen yang hampir serupa dengan komputer pada umumnya. Seperti CPU, GPU, RAM, Port USB, *Audio Jack*, HDMI, *Ethernet*, dan GPIO. Untuk tempat penyimpanan data dan sistem operasi Raspberry Pi tidak menggunakan *harddisk drive* (HDD) melainkan menggunakan *Micro* SD dengan kapasitas paling tidak 4 GB, sedangkan untuk sumber tenaga berasal dari *micro* USB *power* dengan sumber daya yang direkomendasikan yaitu sebesar 5V dan minimal arus 700 mA.

Raspberry Pi juga memiliki beberapa kekurangan jika dibandingkan dengan komputer pada umumnya. Salah satunya Raspberry Pi mempunyai arsitektur berupa *ARM* yang berbeda dengan komputer pada umumnya yaitu arsitektur x86 yang mengakibatkan beberapa *package* ubuntu tidak dapat di install pada Raspberry

Pi. Raspberry Pi juga memiliki spesifikasi CPU dan RAM yang cenderung lebih rendah disbanding komputer pada umumnya.

Raspberry Pi dapat digunakan layaknya PC konvensional, seperti untuk mengetik dokumen atau sekedar *browsing*. Namun Raspberry Pi juga dapat digunakan untuk membuat ide-ide inovatif seperti membuat robot yang dilengkapi dengan Raspberry Pi dan kamera, atau mungkin dapat membuat sebuah super komputer yang dibuat dari beberapa buah Raspberry Pi. Kelengkapan Raspberry Pi di antaranya memiliki *port* atau koneksi untuk *display* berupa TV atau monitor serta koneksi USB untuk *keyboard* serta mouse (Muchlisin Riadi,2020).



Gambar 2.1 Beberapa jenis raspberry pi

2.2 Pengertian Website

Pengertian *website* adalah halaman atau kumpulan halaman pada sebuah domain di internet yang dibuat dengan tujuan tertentu. *Website* umumnya berisi tampilan halaman berupa teks, gambar, animasi, audio, video atau gabungan satu

dengan lainnya. Dalam pengertian lain, *website* adalah kumpulan halaman web yang saling terkoneksi dan memiliki data informasi yang saling terkait. *Website* dibuat untuk dapat diakses secara luas melalui sebuah aplikasi peramban menggunakan URL (*Uniform Resource Locator*). Contoh URL yang paling umum adalah http://www.detik..com. Dalam suatu halaman web, biasanya terdapat berbagai macam jenis informasi dalam bentuk teks, video, gambar, suara, dan lain-lain.

Website pertama di dunia lahir di tahun 1991 pada tanggal 6 Agustus oleh Tim Berners-Lee. Ia adalah seorang ahli komputer dari Inggris yang bekerja di laboratorium fisika Swiss. Pada awalnya, *website* itu dibuat untuk memudahkan para peneliti untuk saling bertukar informasi. Namun, siapa yang menyangka bahwa kejadian sederhana ini akan menjadi sebuah lompatan besar pada sistem komunikasi seluruh umat manusia. Tonggak berdirinya sejarah website secara terbuka adalah pembuatan *website* bernama Mosaic oleh NCSA tahun 1993. Setelah itu, CERN, sebagai laboratorium tempat Tim Berners-Lee bekerja, membantu membuat perangkat lunak web itu secara cuma-cuma.

Seiring berjalannya waktu, *website* semakin berkembang karena sifatnya yang open *source*. Di akhir tahun 1993, terdapat lebih dari lima ratus situs *website*. Sebagai perbandingan, di akhir tahun 2018, jumlah situs *website* yang tercatat ialah 1,94 miliar di seluruh dunia dengan jumlah pengguna internet sebanyak 41 miliar jiwa (putra,2020).

2.3 *Web Real-time Communication* (WebRTC)

WebRTC merupakan sebuah upaya untuk menambahkan kemampuan komunikasi *realtime* ke dalam semua browser dan membuat kemampuan ini dapat diakses pengembang web melalui *tag* standar HTML5 dan API JavaScript. Dengan mengikuti standar-standar yang ditentukan dalam WebRTC, browser menjadi berkemampuan untuk berinteraksi secara langsung dengan browser yang lain. Untuk saat ini browser yang mendukung penggunaan teknologi WebRTC adalah Google Chrome, Mozilla Firefox dan Opera baik untuk desktop maupun *mobile*. (Bartholomeus Bismo Baruno, Alif Subardono, Sri Lestari,2015)

WebRTC (*Web Real Time Communication*) adalah Proyek *open source* di mana memungkinkan setiap pengguna melakukan komunikasi terhadap pengguna lainnya secara *real time* melalui browser. Proyek ini memanfaatkan kemampuan web browser modern di mana komunikasi dalam hal ini meliputi suara dan video dengan memanfaatkan API Javascript yang ada tanpa bantuan *plugin* lain.

WebRTC tentu sangat berguna untuk memberikan kemudahan dalam berkomunikasi antar pengguna. Kalian pastinya pernah menggunakan fitur *call* yang ada pada aplikasi Whatsapp pada *smartphone*, atau kalian mungkin pernah melakukan video call dengan orang lain. Kedua hal tersebut termasuk dalam kategori RTC (*Real Time Communication*). Perkembangan WebRTC tentu akan memberikan sebuah cara pandang baru terhadap cara berkomunikasi pada masyarakat.

Komunikasi *realtime* bukan lagi sesuatu hal yang baru dalam dunia komunikasi teknologi, Kita pastinya sudah tahu aplikasi Skype yang memang sudah ada sejak rilis perdananya pada Agustus 2003, di mana skype sendiri adalah program komunikasi yang menggunakan teknologi *peer to peer* yang menawarkan komnikasi *realtime* yang dibuat secara gratis untuk memberikan fasilitas komunikasi murah berbasis internet yang dapat melintasi negara.

Penerapan RTC saat ini sudah mulai berkembang sebagai fasilitas yang bisa digunakan dalam sebuah *website*. Saat ini aktivitas berkomunikasi seperti *video call* sudah bisa dilakukan secara langsung dari sebuah *website* tanpa harus meng-*install* aplikasi lain sebagai pendukungnya atau yang biasa disebut WebRTC. (Maykhel David,2017).

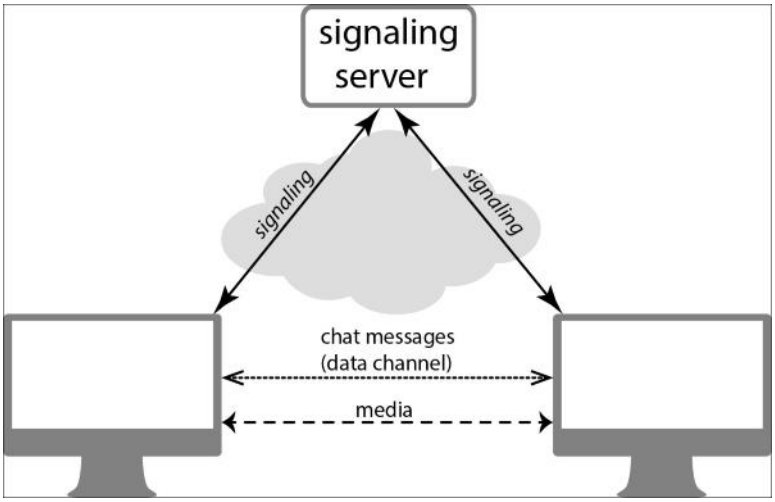
2.3.1 *Peer Connection*

Peer Connection adalah komponen WebRTC yang menangani komunikasi *streaming* data antar *peer*. Kemampuan komponen ini dilakukan melalui objek JavaScript ke pengembang. Objek ini mengabstraksi sejumlah besar detail dan kompleks yang terkait dengan cara kerja bagian dalam video dan audio termasuk penyembunyian *packet loss*, pembatalan gema, adaptasi *bandwidth*, kontrol gain otomatis, kontrol ICE untuk *traversal* NAT, dll (Fernandez dkk, 2013).

2.3.2 RTC *Data Channel*

Data channel didefinisikan sebagai API pada tingkatan aplikasi yang bisa mencerminkan API untuk WebSocket, yang menjelaskan *stream* dua arah dari data dan bidang tekstual yang disebut ”label” dan digunakan untuk

mengidentifikasi arti dari data *channel*. Realisasi dari sebuah data channel merupakan pasangan antara satu *incoming stream* dengan *outgoing* SCTP (*Stream Control Transmission Protocol*) *stream* yang memiliki penanda SCTP *stream* yang sama. Penanda SCTP *stream* dipilih tergantung pada protokol dan implementasinya. Hal ini memungkinkan terjadinya komunikasi dua arah (Rohman, 2019).



Gambar 2.2 Data Channel

2.4 Javascript

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis yang sangat cocok untuk gaya pemrograman berorientasi objek dan fungsional. JavaScript menurunkan sintaksnya dari Java, fungsi kelas satu dari Skema, dan pewarisan berbasis prototipe dari *Self*. JavaScript adalah bahasa pemrograman Web. Sebagian besar situs web modern menggunakan JavaScript, dan semua browser web modern di desktop, konsol *game*, tablet, dan ponsel pintar, menjadikan JavaScript sebagai bahasa pemrograman yang paling banyak digunakan dalam

sejarah. JavaScript adalah bagian dari tiga serangkai teknologi yang harus dipelajari oleh semua pengembang Web: HTML untuk menentukan konten halaman web, CSS untuk menentukan presentasi halaman web, dan JavaScript untuk menentukan perilaku halaman web (Flanagan, 2011).

JavaScript dibuat di Netscape pada hari-hari awal Web, dan secara teknis, "Java-Script" adalah merek dagang yang dilisensikan dari Sun Microsystems (sekarang Oracle) yang digunakan untuk menggambarkan implementasi bahasa Netscape (sekarang Mozilla). Netscape mengirimkan bahasa untuk standarisasi ke ECMA—Asosiasi Pabrikan Komputer Eropa—dan karena masalah merek dagang, versi standar bahasa tersebut terjebak dengan nama aneh “ECMAScript.” Untuk alasan merek dagang yang sama, versi bahasa Microsoft secara resmi dikenal sebagai "JScript." Dalam praktiknya, hampir semua orang menyebutnya bahasa JavaScript (Flanagan, 2011).

2.5 *Hypertext Markup Language* (HTML)

Hypertext Markup Language (HTML) adalah bahasa pemrograman yang digunakan untuk menampilkan sebuah *website*. HTML termasuk dalam bahasa pemrograman gratis, artinya tidak dimiliki oleh siapa pun, pengembangannya dilakukan oleh banyak orang di banyak negara dan bisa dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global. Dokumen HTML adalah dokumen teks yang dapat diedit oleh editor teks apa pun. Dan disimpan dengan *file extension* .html . Dokumen HTML punya beberapa elemen yang dikelilingi oleh tag-teks yang dimulai dengan *symbol* “ <”dan berakhir dengan sebuah *symbol*“>” (Sari dkk, 2019).

HTML pada awalnya dirancang untuk berbagi dokumen statis berbasis teks di Internet. Seiring waktu, karena pengguna web dan desainer menginginkan lebih banyak interaktivitas dalam dokumen HTML mereka, mereka mulai meningkatkan dokumen ini, dengan menambahkan fungsionalitas formulir dan kemampuan tipe "portal" awal. Sekarang, kumpulan dokumen statis ini, atau situs web, lebih seperti aplikasi web, berdasarkan prinsip aplikasi desktop klien/server yang kaya. Aplikasi web ini digunakan di hampir semua perangkat: laptop, ponsel pintar, tablet—semuanya (Wang dkk, 2013).

2.6 *Cascading Style Sheet* (CSS)

Cascading Style Sheets (CSS). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup / markup language*. Jika kita berbicara dalam konteks web secara bebas, CSS bisa di artikan secara bebas sebagai bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML (Sari dkk, 2019).

CSS telah berkembang selama bertahun-tahun dari CSS1 ke CSS3. Ekstensi CSS yang lebih baru juga telah diusulkan seperti pra-prosesor CSS. CSS1 adalah versi sederhana dengan sekitar 50 properti dan sebagian besar digunakan untuk presentasi berbasis layar. CSS2 mencakup semua properti CSS1 ditambah sekitar 70 properti tambahannya sendiri. Properti tambahan misalnya mengaktifkan CSS2 untuk menggambarkan presentasi aural dan jeda halaman yang tidak dapat dilakukan sebelumnya. CSS 2.1 yang disempurnakan juga dirilis yang menambahkan lebih banyak fitur seperti kemampuan untuk menggambarkan bagian-bagian yang didukung oleh dua atau lebih browser. Akhirnya, CSS3 diatur

ke dalam modul seperti latar belakang dan batas, pemilih, efek teks, model kotak, nilai gambar, dan konten yang diganti, animasi, transformasi 2 dan 3 dimensi, tata letak kolom ganda, dan antarmuka pengguna. Tujuan dari modularisasi adalah untuk memiliki beberapa spesifikasi, di mana setiap spesifikasi memiliki jalur perkembangannya sendiri (Ndia, 2019).

Cascading Style Sheets (CSS) menggunakan bahasa yang terpisah dari HTML. CSS memungkinkan Anda untuk menerapkan gaya elemen yang konsisten di semua halaman di situs Anda, sehingga semua judul, daftar, dan paragraf terlihat dan bertindak sama di setiap halaman situs (Osborn, 2011).

2.7 NodeJS

Node.js adalah salah satu teknologi berbasis JavaScript paling populer saat ini. Itu dibuat pada tahun 2009 oleh Ryan Dahl dan sejak itu, kerangka kerja Node.js telah berkembang menjadi ekosistem yang berkembang dengan baik. Manajer paketnya penuh dengan modul yang berguna dan pengembang di seluruh dunia telah mulai menggunakan Node.js di lingkungan produksi mereka (Tsonev, 2015).

Arsitektur Node.js mirip dengan *loop* peristiwa yang dapat bekerja dengan operasi *input/output nonblocking*. Operasi ini akan mengizinkan aktivitas pemrosesan lainnya untuk melanjutkan sebelum tugas yang sedang berlangsung dapat diselesaikan. Karakteristik ini sangat penting jika kita ingin menangani ribuan permintaan secara bersamaan. Sebagian besar server Node.js yang ditulis dalam Java atau C menggunakan *multi-threading*. Dalam *single-threaded,* semua permintaan yang datang ke server diproses oleh satu utas. Ini mungkin terdengar

seperti solusi yang tidak dapat diskalakan, tetapi Node.js jelas dapat diskalakan dengan menjalankan proses Node.js yang berbeda dan menggunakan penyeimbang beban yang mendistribusikan permintaan di antaranya (Tsonev, 2015).

JavaScript sebagai bahasa tidak memiliki mekanisme untuk mendefinisikan kelas nyata. Faktanya, semua yang ada di JavaScript adalah objek. Biasanya Node.js mewarisi properti dan fungsi dari satu objek ke objek lainnya. Untungnya, Node.js mengadopsi konsep yang didefinisikan oleh CommonJS, proyek yang menetapkan ekosistem untuk JavaScript (Tsonev, 2015).

2.8 Websocket

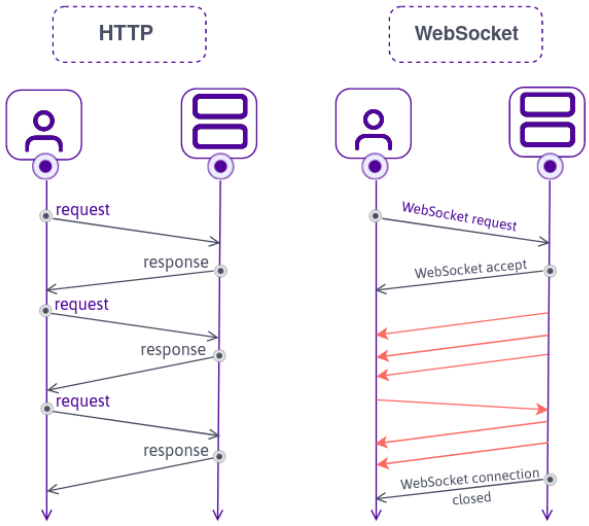
WebSocket adalah koneksi soket tunggal dupleks penuh, dua arah, dan alami. Dengan WebSocket, permintaan HTTP menjadi satu permintaan untuk membuka koneksi WebSocket (baik WebSocket atau WebSocket melalui TLS (*Transport Layer Security*, sebelumnya dikenal sebagai SSL)), dan menggunakan kembali koneksi yang sama dari klien ke server, dan server ke klien. WebSocket mengurangi latensi karena setelah koneksi WebSocket dibuat, server dapat mengirim pesan saat tersedia. Misalnya, tidak seperti *polling*, WebSocket membuat satu permintaan. Server tidak perlu menunggu permintaan dari klien. Demikian pula, klien dapat mengirim pesan ke server kapan saja. Permintaan tunggal ini sangat mengurangi latensi selama *polling*, yang mengirimkan permintaan secara berkala, terlepas dari apakah pesan tersedia (Wang, 2013).

WebSocket adalah protokol, tetapi ada juga WebSocket API, yang memungkinkan aplikasi Anda mengontrol protokol WebSocket dan merespons

kejadian yang dipicu oleh server. API dikembangkan oleh W3C (*World Wide Web Consortium*) dan protokol oleh IETF (*Internet Engineering Task Force*). WebSocket API sekarang didukung oleh browser modern dan menyertakan metode dan atribut yang diperlukan untuk menggunakan koneksi WebSocket dua arah dupleks penuh. API memungkinkan Anda untuk melakukan tindakan yang diperlukan seperti membuka dan menutup koneksi, mengirim dan menerima pesan, dan mendengarkan peristiwa yang dipicu oleh server. Protokol WebSocket memungkinkan komunikasi dupleks penuh antara klien dan server jauh melalui Web, dan mendukung transmisi data biner dan *string* teks (Wang, 2013).

Di WebSocket, hanya perlu melakukan tindakan "*handshake*" sederhana antara browser dan server, kemudian jalur cepat terbentuk antara browser dan server. Setelah terhubung, WebSocket sebagai bingkai data akan dikirim dan diterima dalam mode dualchannel. Antara klien dan server, koneksi WebSocket dibuat pada "*handshake*" pertama melalui protokol WebSocket. Dan protokol tersebut didasarkan pada protokol TCP/IP yang mendasarinya. Protokol WebSocket sederhana, klien seperti browser biasa melalui 80 atau 443 port permintaan server. Server menurut HttpHeader mengidentifikasi apakah sambungan adalah permintaan WebSocket, dan jika demikian, akan ditingkatkan sebagai sambungan. Setelah jabat tangan berhasil, koneksi memasuki fase transfer data koneksi dua arah yang panjang. Transfer data soket web didasarkan pada mode bingkai: 0x00 menunjukkan bahwa data dimulai, 0xff menunjukkan akhir data, data adalah pengkodean utf-8. Dibandingkan dengan permintaan HTTP tradisional, dalam proses jabat tangan WebSocket, informasi *header* permintaan sangat kecil

antara server dan klien, hanya 2 byte, sangat mengurangi permintaan *bandwidth* yang ditempati dan sumber daya server (Zhang dan Shen, 2013).



Gambar 2.3 Perbandingan *life cycle* HTTP dan websocket.

2.9 Socket.IO

Socket.IO adalah alat untuk membuat *real-time application* dengan komunikasi dua arah antara sisi server dan sisi klien. Ini memanfaatkan kekuatan WebSockets bersama dengan beberapa *fallback*, termasuk *long polling* JSON dan *long polling* JSONP melalui satu API terpadu. Ini dapat digunakan untuk membuat interaksi dua arah, seperti *real-time dashboard*, aplikasi obrolan, dan *multiplayer game* (Cadenhead, 2015).

Socket.IO adalah *opensource library* yang dibuat oleh Guillermo Rauch. Dibangun dengan Engine.IO, yang merupakan abstraksi tingkat rendah di atas teknologi WebSocket. Socket.IO digunakan untuk berkomunikasi dua arah antara sisi server dan sisi klien dalam sintaks yang terlihat seolah-olah Anda hanya

memicu dan mendengarkan peristiwa. Protokol API WebSocket di standarisasi pada tahun 2011. Ini adalah Protokol Kontrol Transmisi (TCP) yang hanya mengandalkan HTTP untuk jabat tangan awalnya. Setelah *handshake* selesai, koneksi dibiarkan terbuka sehingga server dan klien dapat meneruskan pesan bolak-balik sesuai kebutuhan (Cadenhead, 2015).

Socket.IO lebih dari sekadar menyediakan API yang mudah digunakan dan lebih kuat di atas WebSockets. Socket.IO juga menyediakan kemampuan untuk menggunakan protokol *real-time* lainnya dengan mulus jika WebSockets tidak tersedia. Jadinya, itu akan kembali pada JSON *long polling* tanpa adanya dukungan WebSocket. *Long polling* pada dasarnya adalah trik untuk meniru perilaku WebSocket di browser yang tidak mendukung WebSocket. Setelah permintaan *polling* yang lama dibuat, permintaan tersebut ditahan oleh server alih-alih segera merespons seperti permintaan HTTP tradisional. Saat data tersedia, permintaan *long polling* diselesaikan, menutup *loop* dari siklus permintaan yang panjang. Pada titik ini, permintaan *long polling* yang baru biasanya akan dibuat. Ini memberikan ilusi koneksi berkelanjutan yang disediakan WebSockets (Cadenhead, 2015).

2.10 PeerJS

Peer adalah modul dari NodeJS yang dibangun untuk membantu koneksi client peer.js pada aplikasi *website peer-to-peer*. *Peer* digunakan pada metadata sesi dan pensinyalan kandidat. Ada dua koneksi yang bisa dibuat oleh modul *peer* tersebut yaitu koneksi data dan koneksi media seperti audio dan video. Sebelum koneksi dilakukan, modul *peer* harus dijalankan terlebih dahulu pada server. Sebuah koneksi membutuhkan identitas seperti IP pada komputer. Ketika teman

ingin berhubungan, teman membutuhkan identitas teman yang dituju. *Peer* akan menerima *event Connection* untuk koneksi dan atau *Call* untuk koneksi media ketika *peer* menerima permintaan untuk saling berhubungan. Data dari masing-masing *peer* dikirim melalui WebSocket (Rohman, 2019).