BAB I

PENDAHULUAN

A. Latar Belakang masalah

Penggunaan teknologi internet semakin hari terus semakin menngkat, dikarenakan internet sangat berperan penting dalam berbagai aspek kehidupan. Pengguna internet dan penyedia layanan di internet juga semakin beragam. Banyak penyedia layanan multimedia yang menyediakan konten multimedia maupun layanan multimedia khususnya audio dan video melalu teknologi streaming seperti layanan live streaming maupun Video on Demand. Banyak juga produsen yang menyediakan layanan atau aplikasi multimedia video conference / teleconference yang dapat melakukan komunikasi 2 arah secara real-time. Teknologi tersebut tersebut dapat dimanfaatkan dalam berbagai macam bidang misalnya pada bidang kesehatan, pendidikan, hiburan dan lain sebagainya. Pada bidang pendidikan, teknologi streaming maupun video conference dapat dimanfaatkan sebagai metode pembelajaran jarak jauh (distance learning). Video Conference merupakan salah satu metode penyampaian informasi yang cukup diminati untuk komunikasi jarak dimana data yang disampaikan berbentuk video dan audiovisual yang dapat memudahkan client di berbagai tempat berbeda untuk saling bertukar informasi secara real time.

WebRTC atau Web Real-Time Communications merupakan sebuah proyek open source yang memungkinkan dilakukannya komunikasi real-time lintas web browser. Komunikasi ini nantinya terdiri dari pemanfaatan suara, video dan konektifitas menggunakan library javascript tanpa sebuah plugin tambahan.

Dengan WebRTC semuanya akan berubah menjadi lebih mudah karena tidak memerlukan biaya besar dalam pengembangan layanan realtime (Mulki Rifky Akbar, 2014). Komunikasi WebRTC merupakan komunikasi real-time lintas web dimana komunikasi dilakukan melalui internet (Desy Damarsari Nazilah, M. Nanak zakaria, Aisah, 2017).

pada penelitian ini perencanaan penerapan aplikasi video conference dengan menggunakan WebRTC dalam melakukan komunikasi nyata dengan media browser. Pengaplikasian video conference ini bertujuan untuk dapat melakukan komunikasi nyata melalu web browser sebagai medianya tanpa harus memiliki plugin atau mengunduh aplikasi, sebagai pengguna dapat melakukan komunikasi dalam waktu bersamaan, meskipun ditempat yang berbeda. Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka penulis tertarik untuk mengambil judul skripsi ini dengan judul " Perancangan Aplikasi Video Conference Berbasis Web Menggunakan WebRTC" diharapkan mampu menangani permasalahan yang telah diuraikan di atas.

B. Rumusan Masalah

Berdasarkan latar belakang masalah diatas, pokok malasah dalam penelitian ini adalah "Perencanaan Aplikasi Video Conference Berbasis Web Menggunakan WebRTC". Dengan rumusan masalah :

Bagaimana merancang aplikasi video conference berbasis web dengan menerapkan teknologi WebRTC ? dan bagaimana membuat sambungan Peer-to-peer menggunakan WebRTC pada web browser?

C. Batasan Masalah

Untuk menghindari pembahasan diluar permasalahan, Berikut ini adalah batasan permasalahan:

1. Aplikasi yang dibuat, menyediakan dan membantu user berinteraksi secara langsung dengan user lainnya secara dua arah melalui video conference.
2. Sistem video conference ini dibangun menggunakan teknologi WebRTC API yang di akses menggunakan javascript dan firebase serta bahasa pemograman HTML dan CSS sebagai Tampilan web.
3. Back-End untuk sistem ini menggunakan server Node.js untuk menangani aplikasi real-time berbasis web dan socket.io untuk menangani proses signaling.
4. Uji coba dilakukan pada web browser yang mendukung pada web browser yang mendukung system layanan multimedia WebRTC yaitu Google Chrome 23 (atau versi yang lebih baru).
5. Video conference ini dilakukan dengan cara peer-to-peer untuk menghubungkan antara Client A dengan Client B dan seterusnya.
6. Memerlukan aplikasi whatsapp sebagai media sharelink untuk membagikan link meeting untuk join ke room yang tersedia.
7. Aplikasi ini hanya untuk melakukan video conference.
8. Pada penelitian ini hanya membahas bagaimana cara terbentuknya video conference dengan menggunakan WebRTC.

D. Tujuan

Tujuan dari penelitian ini yaitu merancang dan membuat aplikasi video conference dengan menggunakan protokol WebRTC berbasis web.

E. Manfaat Penelitian

1. Bagi Penulis

   Mendapatkan pengetahuan tentang perancangan dan implementasi Video Conference dengan memanfaatkan WebRTC

2. Bagi Pembaca

   Dapat menambah wawasan ilmu pengetahuan dan sebagai bahan acuan atau referensi bagi yang berminat dalam pembahasan yang sama.

3. Bagi Institusi

   Memberi sumber acuan (refrensi) dan bahan kajian bagi upaya pengembangan aplikasi serta menambah literatur perpustakaan dan bahan pertimbangan yang berhubungan dengan Tugas Akhir ini kelak.

F. Sistematika penulisan

Sistematika penulisan dibuat untuk memudahkan dalam penyusunan skripsi, maka perlu ditentukan sistematika yang baik dan benar. Sistematika penulisan ini dibagi menjadi dalam beberapa bab yaitu:

BAB I PENDAHULUAN

Bab pendahuluan mendeskripsikan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi uraian tentang teori-teori yang digunakan sebagai refrensi dalam Rancangan Bangun Aplikasi Video Conference Berbasis Web menggunakan WebRTC.

BAB III METODE PENELITIAN

Pada bab ini berisi mengenai metode perencanaan yang digunakan dalam membangun sistem ini.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang hasil yang diperoleh setelah melakukan penelitian dan datanya telah di implementasikan kedalam sebuah aplikasi. Serta berisi tentang analisis aliran data, detail aplikasi, perancangan database, serta pengujian system

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan yang diperoleh dari keseluruhan penelitian dan saran yang mana aplikasi ini bisa untuk lebih dikembangkan kedepannya.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan dari pertimban Penulisan dalam melakukan penelitian, beberapa penelitian terdahulu yang dilakukan peneliti lain :

Tabel 2. 1. Tabel Penelitian Terdahulu

Nama Peneliti	Judul Penelitian	Hasil penelitian
Yeni Lina, Melinda, Sebastian Karuna Alfasan	Analisis dan perancangan aplikasi *meeting* berbasis web	Penelitian ini meng-hasilkan aplikasi yang memungkinkan per-temuan secara maya tanpa harus menghadiri secara langsung
Perbedaan : Penelitian yang dilakukan yen lina, Meliana, Sebastian karuna Alfasan ini menggunakan *platform* video *conference open source* selain WebRTC		
Nama Penelitian	Judul Penelitian	Hasil Penelitian
Tarcius Wahyu Cahya-widi Sutanto (2011)	Pembangunan Aplikasi text *Chatting* dan video *chatting* berbasis *web*	Penelitian ini menghasil-kan aplikasi yang bisa

		menghubungkan komuni-kasi antara manusia sehingga terjalin baik.
Perbedaan : Penelitian yang dilakukan Tarcius Wahyu Cahyawidi Sutant o ini merupakan video *conference* yang hanya dapat digunakan oleh 2 orang pengguna yang mana satu sebagai host dan satu client sedangkan pada penelitian penulis dapat dilakukan oleh banyak orang .		
Nama Peneliti	Judul Penelitian	Hasil Penelitian
Desi Darmasari Nazilah (2017	Perencanaan dan imple-mentasi *protocol* video conference pada keluarga narapidana penghuni lembaga permasyarakatan menggunakan s*martph-one*	Penelitian ini sebagai penunjang *prototype* di lembaga permasyaraka-tan
Perbedaan : Penelitian ini dikhususkan untuk para narapidana di lapas untuk bertemu secara *virtual* dengan keluarganya, sedangkan aplikasi penulis ditujukan untuk umum.		

B. Tinjauan teori

1. Video Conference

Video *conference* merupakan teknologi yang seluruh penggunanya berapa pada tempat yang berbeda namun dapat melakukan pertemuan secara *virtual* tanpa bertemu secara langsung. Bisa dirasakan bahwa menggunakan video *conference* adalah cara yang nyaman dan praktis untuk menjalankan sebuah pekerjaan maupun proses belajar. Waktu, biaya, dan tenaga pastinya akan berkurang dengan memaksimalkan manfaat dari dilakukannya video *conference* ini. Sebagai contoh ketika menyelenggarakan rapat dalam skala besar, tidak harus mengeluarkan biaya untuk *meeting* rutin, berbincang masalah kesepakatan bisnis, *interview* kerja, melakukan penelitian.

Dengan wawancara, dan beberapa hal lainnya baik formal maupun informal.(ilham, 2014).

2. WEB

Web adalah bisnis teknologi dari aplikasi yang akan dibangun pada *proyek* akhir ini. Teknologi *web* dipilih untuk digunakan dalam *implementasi* aplikasi video *conference* yang dibangun dengan pertimbangan karena lebih mudah dalam penggunaan dan sudah umum digunakan oleh *user*, sehingga aplikasi yang berjalan pada *web browser* dapat dengan mudah digunakan oleh penggunaa *internet*.

3. WebRTC

WebRTC (*Web Real-Time Communication*) merupakan *framework open source* yang mengizinkan komunikasi secara *real-time* pada *web* browser secara peer-to-peer tanpa menggunakan berbagai macam *plug-in* (Virag et al, 2014). Komunikasi secara *real-time* pada WebRTC dapat diakses melalui

Javascript API. WebRTC mengijinkan penjelajah web dapat melakukan pertukaran data aplikasi dan juga melakukan performa *teleconferencing audio*/video secara *peer-to-peer*, tanpa melakukan pegintsalan *plug-in* atau perangkat lunak pihak ketiga (virag et al, 2014). WebRTC dirilis *Google* sebagai protikol *open source* yang telah distandarisasi oleh IETF (Internet *Engineering Task Force*) dan W3C (*World Wide Web Consortium*). *Google*, *Mozilla*, *Chrome*, dan *Opera* mendukung WebRTC dan terlibat dalam pengembangan WebRTC.

WebRTC bertujuan untuk menciptakan suatu aplikasi yang *rich* dan *real-time* dengan kualitas tinggi, WebRTC dibangun untuk dapat dijalankan padaberbagai penjelajah *web* dan juga berbagai perangkat meliputi pada komputer personal, perangkat bergerak / *smartphone*, maupun pada *IoT device*, yang mana semua perangkat tersebut dapat saling berkomunikasi satu sama lain menggunakan *protokol* umum yang sudah ada (WebRTC, 2016). Dengan adanya teknologi ini pengguna dapat secara langsung berkomunikasi dan bertukar data melalui penjelajahan *web* tanpa memikirkan penginstalan *plug-in* dan lain sebagainya. Teknologi ini sangat mempermudah dan menguntungkan pengguna yang dapat dilakukan secara langsung kapanpun dan dimanapun pengguna berada melalui beberapa peramban *web* dan juga beberapa *platform* yg berbeda.

Mengingat teknologi WebRTC sangat baik untuk mengembangkan aplikasi *real-time* yang *rich* dan berkualitas tinggi yang memungkinkan *peer-to-peer* dan juga tanpa membutuhkan penginstalan *plug-in* dan lain sebagainya,

WebRTC memiliki beberapa manfaat yang dapat menguntungkan penggunan aplikasi dan juga pengembangan aplikasi *real-time*. Beberapa keuntungan tersebut, yakni :

a. Mengurangi biaya pengembangan aplikasi dan juga mengurang biaya bagi pengguna apikasi. *Developer* atau pengembangan aplikasi tidak memerlukan biaya untuk membeli hak milik karena WebRTC bersifat gratis dan *open source* serta biaya penyebaran dan dukungan aplikasi dapat di kurangi karena tidak membutuhkan penyebaran dengan menggunakan prangkat lunak khusus kepada pengguna. Bagi pengguna aplikasi, mereka tidak perlu juga untuk mengeluarkan banyak biaya karena aplikasi dapat diakses langsung melalui peramban *web* yang umum di gunakan dan mengguanakan berbagai perangkat yang sudah ada sebelumnya seperti komputer, leptop, *smartphone* dan sejenisnya.
b. *Developer* atau pengembang aplikasi dapat dengan mudah membangun aplikasi komunikasi *real-time* menggunakan WebRTC karena fungsionalitas WebRTC dapat dengan mudah diintegrasikan dengan *web service* atau situs yang telah di bangun sebelumya. Fungsi-fungsi dari WebRTC dapat dikembangkan dengan menggunakan WebRTC API yang telah didefinisikan dan dapat diakses melalui Javascript API.
c. Developer hanya perlu membangunkan satu aplikasi untuk semua platfrom dan pengguna juga dapat menggunakan aplikasi dengan berbagai macam perangkat dan platfrom atau peramban *web* yang berbeda.

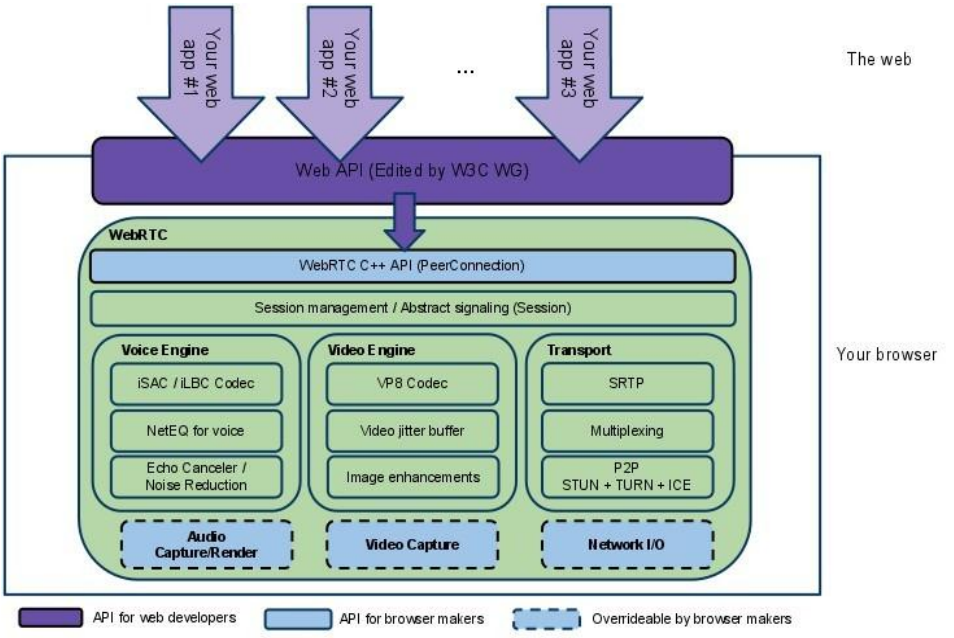
d. Lisensi WebRTC yang *open source* dan gratis memungkinkan sebuah komunitas WebRTC dapat menemukan kesalahan-kesalahan (*bugs*) baru dan dapat menyelesaikan secara efektif dan cepat.

WebRTC terdiri dari 3 API yang diakses menggunakan Javascript, yakni (Viraget al, 20140):

a. *MediaStream* (dikenal juga sebagai *getUserMedia*), yang memungkinkan penjelajah web untuk mendapatkan akses ke aliran data, seperti kamera dan *mikrofon* yang ada pada perangkat untuk menangkap media stream.

b. *RTCPeerConnection*, memungkinkan dua pengguna yang berbeda untuk berkomunikasi data *streaming* (*audio* dan video) secara langsung melalui *peer-to-peer*.

c. *RTCDataChannel*, memfasilitasi pertukaran jenis data selain *audio* dan video antara *peers* yang terhubung.



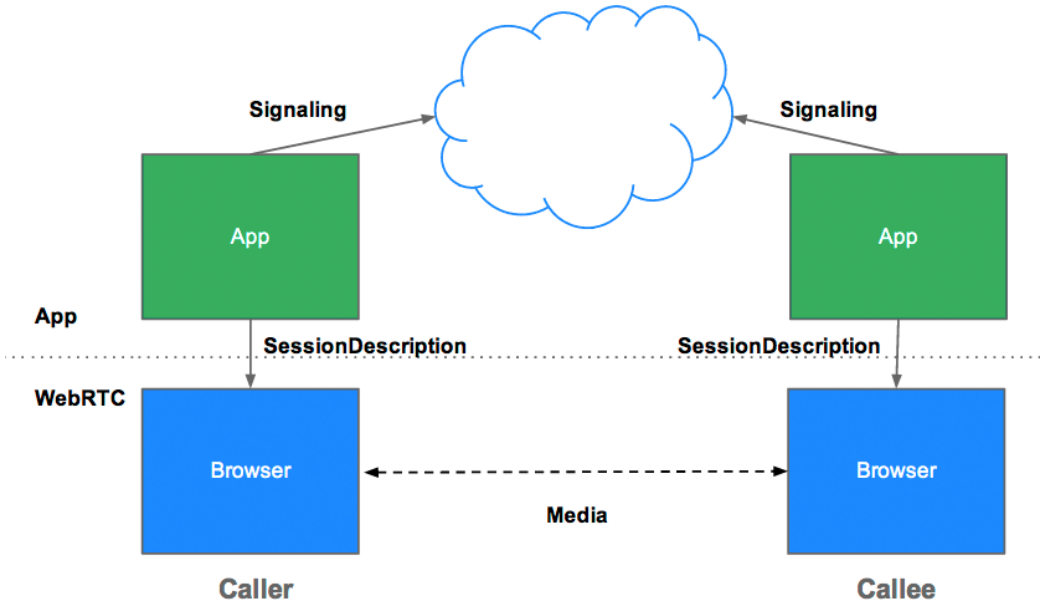
Gambar 2. 1. Arsitektur WebRTC

Arsitektur WebRTC memiliki 3 bagian yang terdiri dari WebRTC, Web API dan Aplikasi Web. Pada bagian WebRTC terdiri dari *Voice Engine* yang bertugas untuk menangani *echo cancellation* dan *noice reduction*. *Echo cancelation* merupakan inprovisasi kualitas suara, menghilangkan gema suara yang masuk pada perangkat *microphone* atau sejenisnya. Sedangkan *noice reduction* merupakan metode penghapusan kebisingan dari sinyal. Pada *voice Engine* terdapat codec iSAC dan iLBC. Pada bagian WebRTC selanjutnya terdapat Video *Engine* yang bertanggung jawab untuk menangani *image enhancement* dan video *jitter buffer*. *Image enhancement* merupakan proses untuk memperbaiki kualitas dari gambar digital dengan memanipulasi menggunakan perangkat lunak tertentu. Pada video *engine* terdapat codec VP8. Kemudian pada bagian WebRTC terdapat *mekanisme transport* yang bertanggung jawab untuk SRTP (*Secure Real-time Transport Protocol*), *multiplexing*, P2P (*Peer-to-Peer*), ICE (*Interactive Connectivity Establishment*), STUN (*Session Traversal Utilities for* NAT) dan TURN (*Traversal Using Relay* NAT). Pada bagian *Web* API bertanggung jawab untuk memudahkan pengembangan aplikasi dari kompeksitas dan manfaat fitur-fitur yang dimiliki oleh WebRTC untuk dikembangkan menjadi beragam aplikasi *real-time*.

WebRTC mengizinkan komunikasi *real-time* antara penjelajah *web* secara *peer-to-peer*, tetapi tetap membutuhkan proses untuk saling bertukar informasi antara *peers* yakni melakukan mekanisme pengiriman pesan kontrol (*control messege*) dan melakukan koordinasi antar *peers*. Proses tersebut dikenal sebagai proses *signaling* (Dutton. S. 2014). Berikut ini pertukaran informasi yang dilakukan pada saat proses *signaling* (Dutton. S, 2014):

a. *Session Control Messege,* digunakan untuk komunikasi sesi mulai dan selesai, serta melaporkan jika terjadinya kesalahan komunikasi.
b. *Network Configuration*, digunakan untuk mengkomunikasikan dan mengetahui alamat ip dan juga port yang digunakan untuk komunikasi.
c. *Media Capabilities*, bertujuan untuk membuat / menetapkan tipe codec dan resolusi yang dapat ditangani oleh penjelajah *web* pengguna dan dengan siapa pengguna / penjelajah *web* yang ingin berkomunikasi.

   Metode dan protokol *signaling* tidak didefinisikan oleh WebRTC karena setiap aplikasi yang berbeda memiliki karakter yang berbeda dan dapat memiliki untuk menggunakan protokol signaling yang berbeda sesuai dengan karakteristik aplikasi tersebut (Datton S., 2013). Setiap aplikasi belum tentu membutuhkan dan menggunakan protokol *signaling* yang sama. Oleh karena itu, WebRTC tidak menentukan protokol signaling dalam API atau standar WebRTC, informasi yang paling penting untuk saling dipertukarkan antara *peers* yakni *multimedia session description* yang menentukan *transport* yang diperlukan dan informasi *konfigurasi media* yang diperlukan untuk membuat media *plane*. Pendekatan proses *signaling* WebRTC dipaparkan oleh JSEP (*Javascript Session Establishment Protocol*) yang telah distandarisasi oleh IETF (*Internet Engineering Task Force*).



Gambar 2.2
Gambar 2. 2. Arsitektur JSEP

Sumber : Dutton S. (2013)

JSEP (*Javascript Session Establishment Protokol*) merupakan *protokol* yang mengontrol penuh proses *signaling* pada aplikasi *Javascript*. JSEP menghapus hampir seluruh inti dari *signalin flow*, yang mana tidak ditangani oleh *Javascript*, yakni melakukan passing terhadap *session descriptsion* secara local maupun *remote*, dan berinteraksi dengan ICE *state Machine* (Uberty et al, 2016). JSEP melakukan pertukaran *proses offer* dan *answer* yang saling berkomunikasi dengan melakukan pertukaran SDP (*Session Description Protocol*) (Dutton .S, 2013).

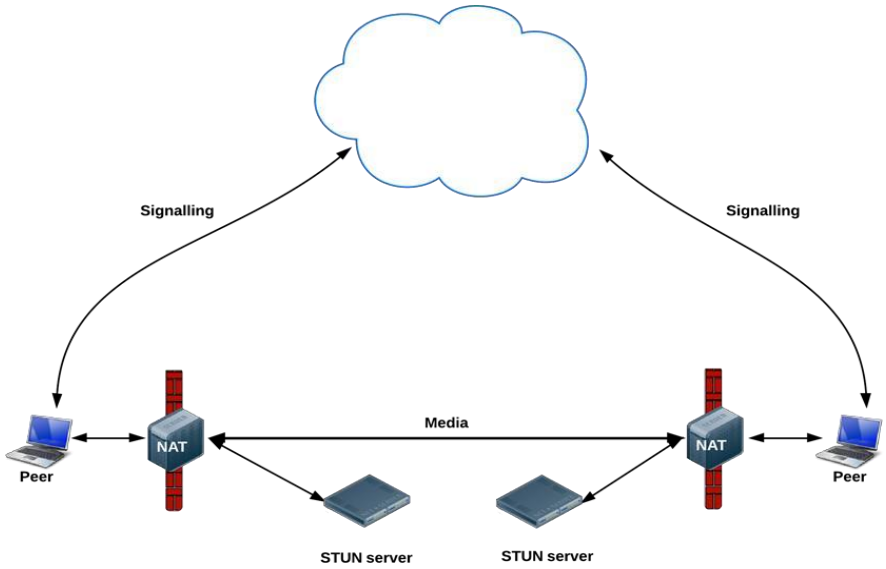
Protokol JSEP dapat menangani *session description* dengan mudah dan sederhana. Setiap kali pertukaran offer / answer dibutuhkan, pada sisi inisiasi membuat sebuah *offer* dengan memanggil *CreateOffer*() API. Kemudian aplikasi akan menggunakan hasil *offer* tersebut untuk melakukan konfigurasi secara lokal (*local config*) melalui *setLocalDescription*() API. Sebagai perlengkapan pertukaran

offer / *answer*, pada sisi *remote* menggunakan *createAnswer*() API untuk menghasilkan *answer* yang sesuai. Sisi remote akan menggunakan *setLoclDescription*() API dan mengirim *answer* kembali kepada initiator melalui *signaling channel*. Saat iniator mendapatkan *answer*, iniator akan melakukan proses instalasi menggunakan *setRemotedescription*() API. Setelah beberapa proses tersebut dilakukan, proses pertukaran *offer* / *answer* telah selesai dilakukan dan proses tersebut dapat dilakukan ulang pada pertukaran *offer* / *answer* tambahan (Uberti et al, 2016).

Komunikasi *Peer-to-Peer* pada WebRTC memerlukan beberapa protokol pendukung agar komunikasi antara *peer* bisa terlaksana dengan baik. Pada kenyataan sesungguhnya didunia nyata, komunikasi *peer-to-peer* pada WebRTC tidak bisa secara langsung berkomunikasi satu sama lain antara *peers*, karena idealnya suatu *client* di jaringan lokal untuk bisa berkomunikasi ke *internet* akan melalui NAT dan juga *firewall*. Oleh karena itu dibutuhkan metode atau *firework* pendukung agar komunikasi *peer-to-peer* bisa dilakukan, yakni dengan menggunakan *framwork* ICE (*Interactive Connectivity Establishment*), STUN (*Session Traversal Utilities for NAT*) dan TURN (*Traversal Using Relay NAT*).

ICE digunakan untuk menemukan informasi yang cukup terhadap topologi yang ada untuk mencari satu atau lebih path yang potensial antara satu sama lain pada *peer-to-peer*. ICE mencoba untuk mencari *path* terbaik untuk koneksi antar Peers. Pertama ICE mencoba untuk membuat koneksi menggunakan alamat *host* yang diperoleh dari sistem operasi perangkat lunak dan *network card*. Jika ICE gagal dalam menangani proses tersebut, ICE akan memperoleh alamat eksternal

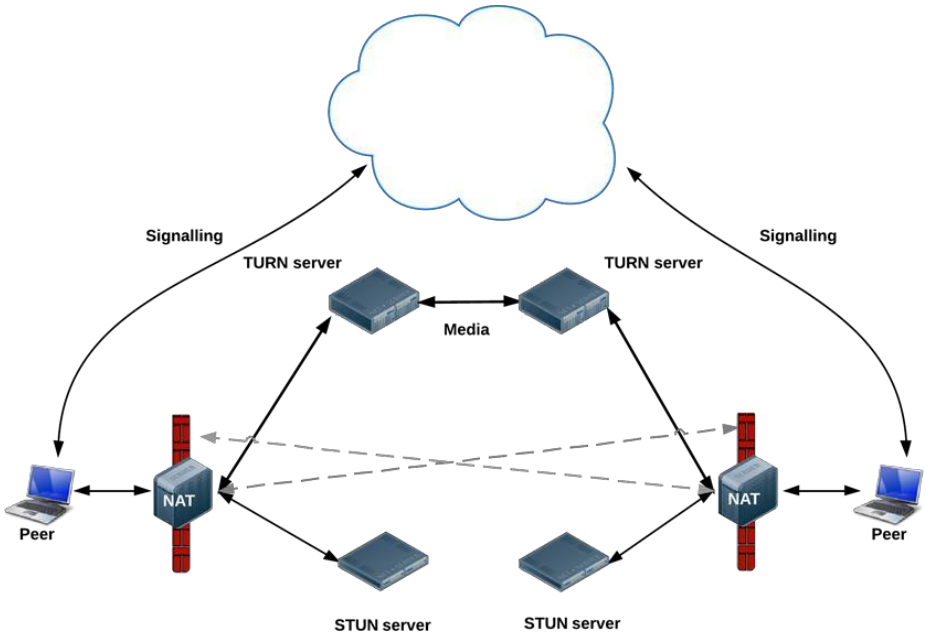
dari STUN *server*. STUN *server* berada pada *internet publik* dan memiliki tugas untuk mengecek alamat IP dan Port pada permintaan (*request*) yang masuk pada aplikasi yang berjalan dibalik NAT, dan mengirim alamat tersebut kembali menjadi sebuah response. Intinya aplikasi menggunakan STUN Server untuk mendapatkan alamat IP dan *port* dari perspektif publik.



Gambar 2. 3. STUN

Sumber : Dutton S. (2013)

Protokol TURN yang memiliki tugas untuk melakukan proses *relay* jika koneksi *peer-to-peer* tidak berhasi dilakukan. Server TURN memiliki alamat publik, sehingga server TURN Server menghubungi *peers* ketika peers berada dibelakang *firewall* atau *proxy*.



Gambar 2. 4. TURN

Sumber : Dutton S. (2013)

Teknologi WebRTC merupakan Teknologi yang masih sangat baru dan masih dalam proses pengembangan. Oleh karena itu, ada beberapa batasan terhadap peramban *web* dan juga platform yang belum mendukung atau hanya mendukung sebagian dari teknologi WebRTC, WebRTC didukung beberapa peramban *web*, Yakni (Sergiienko A, 2014) :

a. Google Chrome Versi 23 atau lebih tinggi
b. Mozzila Firefox Versi 22 atau lebih tinggi
c. Opera versi 18 atau lebih tinggi
d. Internet Explorer, tetapi dengan menambahkan Chrome *Component for IE*.

Pada perangkat *Mobile* atau *Smartphone* khususnya pada platform Android, WebRTC didukung oleh beberapa peramban *web*, yakni () :

a. Google Chrome Versi 29 atau lebih tinggi

b. Mozilla Firefox Versi 24 atau lebih tinggi
c. Opera Mobile Versi 12 atau lebih tinggi
d. Google Chrome OS.

4. NodeJS

NodeJS adalah *Open Source*, cross-platform *runtime environtment* pada sisi server dan aplikasi jaringan. NodeJS merupakan aplikasi yang ditulis dalam Javascript dan dapat dijalankan pada OS X, Microsoft Windows, Linux, *FreeBSD, NonStop,* dan IBM i.

Merupakan sebuah platform yang dibangun di atas Chrome's *Javascript* runtime dengan teknologi V8 yang mendukung proses server yang bersifat long-running. Tidak seperti platform moderen yang mengandalkan *multithreading*. NodeJS memilih menggunakan asynchronous I/O eventing. Karena inilah NodeJS mampu bekerja dengan konsumsi memori rendah (Joyent, Inc, 2015)(S. Tilkov and S. Vinoski, 2010).

Teknologi yang tidak memanfaatkan *multi-thread* ini memudahkan pengembang yang terkadang kesulitan mengatur sumberdaya yang terkunci karena thread yang berjalan. Akhirnya banyak yang memanfaatkan kemampuan dasar NodeJS sebagai *web* server.



```
var http =   require('http');
const PORT=8080;

function handleRequest(request, response)
    {
 response.end('It Works !! Path Hit: '+
     request.url);
}

var server = http.createServer(
   handleRequest);

server.listen(PORT, function(){
 console.log("Server is running on port:
     %s", PORT)});
```

Gambar 2. 5. Contoh Penggunaan NodeJS sebagai Web Server

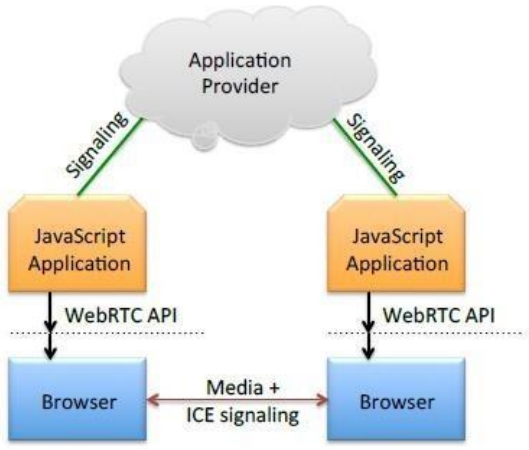
Dengan adanya *callback* untuk setiap penggunaan fungsi, memungkinkan setiap pemanggilan fungsi yang tidak menghasilkan apapun, NodeJS akan menuju mode *sleep*.

5. PeerConnection

PeerConnection adalah sebuah komponen website *realtime communication* yang menangani komunikasi *streaming* data antara *peers* yang saling terhubung. Kemampuan komponen *PeerConnection* dilakukan melalui objek *javascript* yang diperuntukkan ke pengembang. Objek ini mengabstraksi sejumlah detail dan kompleksitas saat ini terkait dengan cara kerja dalam video dan *audio* termasuk dalam penyembunyian *packet loss*, pembatalan echo, adaptasi *bandwidth*, kontrol *gain* otomatis, kontrol ICE Candidate untuk gelombang network address *translate*, dan masih banyak lagi (Javier Lopez, 2013)

6. *MediaStream*

MediaStream adalah representasi media yang ada pada protokol *web realtime* communication dan dilengkapi dengan dua bagian, yaitu *local* dan *remote*. Local *mediastream* digunakan untuk menangani *audio* dan video yang ditangkap secara lokal (contohnya melalui *browser* dalam koneksi lokal) dalam sebuah *webcam* ataupun *mikrofon*. Sebuah *mediastream* lokal dapat dirender dengan menggunakan *tag* <video> standar yang ada pada HTML5. Dengan cara yang sama, *remote mediastream* juga dapat membawa saluran video dan *audio* dan dapat dirender pula mengunakan *tag* <video>. Namun dalam hal ini media yang ada bukan berasal dari kamera ataupun *mikrofon* lokal, akan tetapi dari remote (orang lain) yang ada di ujung komunikasi. Dari perspektif protokol, stream melintasi jaringan mengunakan format dan ICE *candidate* yang sudah dinegosiasikan. Untuk mengamankan jaringan komunikasi, SRTP digunakan menjadi DTLS yang salah satu diantaranya memunkinkan terjadi pertukaran kunci. (Javier Lopez 2013).



Gambar 2. 6. Arsitektur WebRTC saat ini yang didasarkan pada pemisahan antara peroses pensinyalan dan media.

7. 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 mengindentifikasi arti dari kata data *channel* (IETF, 2015). Realisasi dari sebuah data *channel* merupakan pasangan antara satu *incoming stream* dengan outgoing SCTP stream yang sama. Penanda SCTP stream dipilih tergantung pada protokol dan implementasinya. Hal ini memungkinkan terjadinya komunikasi dua arah

8. Peer

Peer adalah modul dari NodeJS yang dibangun untuk membantu koneksi client peer.js pada aplikasi website peer-to-peer. Peer digunakan pada session metadata dan candidate signaling. Ada dua koneksi yang bisa dibuat oleh modul peer tersebut yaitu koneksi data dan koneksi media seperti audio dan video (Peer.js, 2015).

Sebelum koneksi dilakukan modul peer harus dijalankan terlebih dahulu pada server. Sebuah koniksi membutuhkan identitas seperti IP pada komputer, ketika peer ingin berhubungan, peer membutuhkan identitas peer yang dituju.

Peer akan menerima event connection untuk koneksi atau Call untuk koneksi media ketika peer menerima permintaan untuk saling berhubungan . Data dari masing-masing peer dikirim melalui web-socket.

9. Backbone JS

Ketika bekerja dalam lingkungan aplikasi yang menjalankan banyak Javascript, salah satu hal penting yang harus diperhatikan adalah bagaimana menghentikan ikatan data kedalam DOM (*Document Object Model*). Dengan Backbone JS, semua data akan dipresentasikan sebagai model yang bisa dibuat, divalidasi, dihapus, dan disimpan kedalam server.

Secara Harfiah, Backbone merupakan upaya untuk mengatur struktur (model dan collection) dan antarmuka pengguna (*view* dan *URL*) secara primitif yang mana akan berguna dalam pembuatan aplikasi berbasis *web* dengan *Javascript" (*BackboneJS, 2015).

Backbone JS memberikan struktur untuk aplikasi web dengan menyediakan model dengan key-value binding dan custom event, collection dengan API untuk fungsi - fungsi enumerable, view dengan event handling deklaratif, dan menghubungkan itu semua menjadi API melalui antarmuka RESTful JSON.

10. Handlerbar JS

Handlerbar JS merupakan modul *Javascript* yang berfungsi sebagai proses templat yang dinamis yang akan menghasilkan halaman HTML untuk menghemat waktu dan penyusunan kode program secara manual dan berulang-ulang." (HandlebarJS. 2015). Memanfaatkan *Javascript*. *Handlebat* JS mengkomunikasikan *file* JSON dengan HTML untuk pemrosesan templat.

```
<div class="entry">
  <h1>{{title}}</h1>
  <div class="body">
    {{body}}
  </div>
</div>
```

Gambar 2. 7. Contoh Penggunaan Handlebar JS untuk Templat

11. Javascript

Javascript adalah bahasa pemrograman yang populer. *Javascript* adalah bahasa pemrograman digunakan untuk HTML dan *web*, untuk *server*, PC, laptop, tablet dan lebih banyak lagi. Kode pemrograman *Javascript* mulai diperkenalkan di *browser Netscape Navigator* 2. Namun waktu itu namanya bukan *Javascript*, yaitu *LiveScript*. Mengingat pada waktu itu teknologi Java sedang panas-panasnya atau sedang *tren*, maka pihak *Netscape* memutuskan untuk mengganti menjadi *Javascript*, yang seperti nama tersebut lebih *marketible* dibandingkan *Livescript*. Selanjutnya pihak *microsoft* (*rival Netscape*) pun mulai ikut-ikutan memfasilitasi *web* browser buatanya, "*Internet Explorer*". Supaya bisa mendukung *javascript*. Namun mungkin karena gengsi, pihak *Microsoft* memberi nama bahasa lain, yaitu *Javascript*. Mulai saat itu. *Netscape* dan *Microsoft* mulai berlomba-lomba mengembangkan bahasa tersebut dalam versi yang lainnya. Oleh sebab itu persaingan itulah terkadang suatu *Javascript* mungkin bisa bekerja dengan baik di *browser Netscape*, tidak demikian halnya di *Internet Explorer*, begitu pula sebaliknya (Permana, 2016).

Pada awalnya, *Javascript* mulai dipertampilkan di browser *Netscape* Navigator 2. Namun waktu itu namanya waktu itu namanya bukan *Javascript*,

namun *Livescript*. Mengingat pada waktu itu teknologi Java sedang panas-panasnya atau sedang *tren*, maka pihak *Nescape* memutuskan untuk mengganti namanya menjadi *javascript*, yang seperti nama tersebut lebih *marketible* dibandingkan dengan *Livescript*. Selanjutnya pihak *Microsoft* (rival Netscape) pun mulai ikut-ikutan memfasilitasi *web browser* buatannya, '*Internet Explorer'*, supaya bisa mendukung *Javascript*. Namun mungkin karena gengsi, pihak *Microsoft* memberi nama bahasa lain, Jscript. Mulai saat itu, *Netscape* dan *Microsoft* mulai berlomba-lomba mengembangkan bahasa tersebut dalam versi yang berlainan. Oleh sebab persaingan itulah terkadang suatu *Javascript* mungkin bisa bekerja dengan baik di *browser Netscape*, tapi tidak demikian halnya di *Internet Explorer*, begitu pula sebaliknya (Permana 2016).

Ada dua jenis bagaimana *javascript* dibuat, pertama *Javascript* ditulis dalam file yang terpisah dengan HTML, kedua *Javascript* ditulis dalam HTML. Javascript yang ditulis diluar HTML disebut eksternal *Javascript* dengan ekstensi *file* .js. Dalam HTML, penulisan *script* diawali dengan sebuah tag *script*. Selanjutnya script yang dijalankan harus diletakkan diantara <*script*> dan </*script*> *tag*. <*script*> memiliki beberapa atribut, namun yang terpenting adalah *atribut language* dan *type*. Karnea *javascript* bukan satu-satunya bahasa *scripting*, maka sangatlah perlu untuk memberitahukan kepada browser bahwa bahasa script yang digunakan adalah *javascript* dan selanjutnya *browser* akan menjalankan modul *Javascript* untuk memprosesnya (Permana 2016).

12. Video

Video adalah teknologi pemrosesan sinyal elektronik mewakilkan gambar bergerak. Aplikasi umum dari teknologi video adalah televisi, tetapi dia dapat juga digunakan dalam aplikasi teknik, saintifik, produksi dan keamanan" (Google, 2013)

MACAM-MACAM VIDEO

a) Video IP Adalah video yang dilewatkan melalui IP. Terdapat tiga kategori video pada saat mereka dipancarkan pada publik baik melewati satelit, melalui kabel, dan melalui IP atau format radio analog.
    1) Kategori pertama memancarkan video dari satu sumber ke berbagai tujuan. Pemancaran seperti terjadwal, sedangkan pemancaran lewat IP multicast kepada pengguna jaringan bisa secara langsung atau rekaman.
    2) Kategori kedua adalah video berdasarkan permintaan, jadi hanya satu pengguna yang ingin melihat transmisi file video yang disimpan dalam IP unicast. Video ini sebelumnya direkam dan disimpan di server.
    3) Kategori ke tiga adalah sebuah Konferensi video interaktif yang bisa berlangsung antara 2 atau lebih pengguna. Ini adalah tipe video paling kompleks, dan dibutuhkan baik transmisi IP unicast bahkan dibeberapa kasus multicast tergantung skenarionya masing-masing.
b) Video text istilah yang dibuat itu untuk menjelaskan peralatan TV yang digunakan untuk menampilkan data berbasis komputer, baik dikirimkan lewat telepon atau lewat kanal pemancar.

c) Video Out Fitur pada peragkat keras yang bisa menghubungkan kamera ke video in port pada televisi atau monitor dan menampilkan citra digital di layar video.

d) Video RAM disingkat dengan VRAM. Tipe spesial dari DRAM yang memungkinkan akses direct high speed memory melalui sirkuit video. Jenis memori ini mahal bila dibandingkan chip DRAM yang konvensional.

e) Video text suatu kemampuan untuk mengirimkan, mentansmisikan secara dua arah dari suatu gambar dan suara.

13. Audio

Audio dalam sistem komunikasi bercirikan video, sinyal elektrik digunakan untuk membawa unsur bunyi. Istilah ini juga biasa digunakan untuk menerapkan sistem-sistem yang diberkaitan dengan proses perekaman dan transmisi yaitu sistem pengembalian /penangkapan suara, sambungan transmisi pembawa bunyi, amplifier dan lainnya" (Google, 2013).

MACAM-MACAM AUDIO

a. Audio Streaming istilah yang dipergunakan untuk mendengarkan siaran secara live melalui Internet. Berbeda dengan cara lain, yakni mendownload file dan menjalankan di komputer kita bila download-nya sudah selesai, dengan streaming kita dapat mendengarnya langsung tanpa perlu mendownload file-nya sekaligus. Ada bermacam-macam audio streaming misalnya Winamp (mp30, RealAudio (ram) dan liquid radio.

b. Audio Response suara yang dihasilakan oleh komputer. Output pembicaraan yang dihasilakan komputer untuk menanggapi input jenis khusus, misalnya permintaan nomor telepon.

c. Audio Oscillator merupakan produk dari perusahaan Hewlett Packard yang pertama. Produk ini digunakan oleh Walt Disney Studios dalam pembuatan filmnya yang berjudul fantasia.

d. Audio Modem Riser sebuah kartu plug-in untuk motherboard Intel yang memuat sirkuit audio dan atau sirkuit modem. AMR memuat fungsi-fungsi analog (kode-kode) yang diperlukan untuk operasi modem dan audio.“(Google, 2013)

14. HTML

HTML merupakan kependekan dari Hyper Text Markup Language yang berisikan tag-tag markup guna memberitahukan browser bagaimana harus menampilkan sebuah halaman.

HTML (Hyper Text Markup Language) adalah suatu dokument text biasa, yang mengandung tanda-tanda (tag) tertentu yang digunakan untuk menentukan tampilan suatu text dan tingkatan kepentingan dari text tersebut dalam suatu dokumen untuk menulis sebuah halaman web. Kerangka dasar HTML adalah sebagai berikut :

```
<html>
<head>
<title>JudulHalaman</title
</head>
```

```
<body>
.....
.....
</body>
</html>
```

22 p2p-video-chat-webrtc/index.html

```
@@ -0,0 +1,22 @@
+ <!DOCTYPE html>
+ <html>
+   <head>
+     <meta charset="utf-8">
+     <title></title>
+   </head>
+   <body>
+
+     <label>Your ID:</label><br/>
+     <textarea id="yourId"></textarea><br/>
+     <label>Other ID:</label><br/>
+     <textarea id="otherId"></textarea>
+     <button id="connect">connect</button><br/>
+
+     <label>Enter Message:</label><br/>
+     <textarea id="yourMessage"></textarea>
+     <button id="send">send</button>
+     <pre id="messages"></pre>
+
+     <script src="bundle.js" charset="utf-8"></script>
+   </body>
+ </html>
```

Gambar 2. 8. contoh Tag HTML

(GitHub Shame, 2015)

15. PHP (Hypertext Pre Processor)

PHP (Hypertext Pre Processor) adalah sebuah bahasa pemrograman yang berbentuk scripting. Sistem kerja program ini adalah sebagai interpreter bukan sebagai compiler" (Nugroho : 2004). Bahasa interpreter adalah bahasa yang script-script program tidak harus diubah ke dalam bentuk source code.

Sehingga pada saat menjalankan bentuk program, kode dasar secara langsung

Awal	Akhir
<?	?>
<?php	?>
<script languange = “php”>	</script>

akan dijalankan tanpa harus melalui proses pengubahan ke dalam bentuk source code.

Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan di parsing (diuraikan) di dalam web server oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser. Karena pemrosesan program PHP dilakukan di lingkungan web server. PHP dikatakan sebagai bahasa sisi server (server-side). Oleh sebab itu, kode PHP tidak akan terlihat pada saat user memilih perintah “View Source” pada web browser yang digunakan.

PHP (Hypertext PreProcessor) adalah sebuah script yang bersifat server-side yang artinya semua proses akan dikerjakan di sisi server dan hasilnya saja yang akan dikirim ke browser. PHP pertama kali dibuat oleh Rasmus Lerdorf untuk mengetahui siapa yang telah mengunjungi situsnya. Adapun beberapa aturan dalam penulisan kode program PHP ditunjukan pada tabel berikut :

Tabel 2. 2. Kode Pembuka dan Penutup PHP

16. CSS (Cascading Style Sheet)

CSS (Cascading Style Sheet)merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemograman.

Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. SS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.”(wikipedia, 2021)

17. STUN, TURN, ICE

Merupaka suatu server yang dibutuhkan oleh WebRTC ketika melakukan Peer-to-peer Communication. Karena klien dari WebRTC membutuhkan pertukaran metadata untuk mengkoordinasikan suatu komunikasi yang disebut signaling. Selain itu STUN, TURN, ICE dibutuhkan untuk mengatasi masalah NAT dan firewalls pada tiap PC yang akan saling berkomunikasi agar

komunikasi peer-to-peer dapat dilakukan." (Ahson, Syed A., Mohammad Ilyas, 2009).

18. API (Application Programming Interface)

Merupakan sebuah perangkat lunak yang dapat menerima panggilan atau permintaan dari perangkat lunak lainnya seperti aplikasi dan website yang memberikan pelayanan. Dengan kata lain API merupakan sebuah perangkat lunak yang mengijinkan atau menghubungkan dua aplikasi tidak hanya itu API juga digunakan untuk membuat software dan aplikasi oleh para developer." (Alif Subardono, Sri Lestari, 2015)

19. Firebase

Firebase adalah layanan Google yang berguna untuk membantu para developer untuk mengembangkan aplikasi. Firebase atau BaaS alias Backend as a Service merupakan cara developer yang fokus mengembangkan aplikasi tanpa perlu memberikan effort besar. Sejak ditemukan oleh Andrew Lee dan James Tamplin tahun 2011, Firebase memberikan produk Realtime Database yang berguna untuk menyimpan lebih banyak data dan sinkronisasi ke banyak pengguna. Pada tahun 2014 lalu, pihak Google mengakuisisi layanan pengembang aplikasi ini. Seiring perkembangan teknologi dan informasi yang semakin canggih, kecepatan layanan menjadi prioritas utama bagi pengguna. Kini developer bisa menggunakan layanan Firebase untuk memudahkan pengembangan aplikasi baik gratis atau berbayar sesuai pemakaian layanan.

Berikut beberapa fungsi Firebase berdasarkan fitur yang dimiliki yaitu:

a. Firebase Analytics

Merupakan fitur yang berfungsi untuk mengoleksi data yang bervariasi dan reporting baik perangkat Android atau iOS. Berdasarkan fitur Firebase Analytics bisa memberikan kemudahan bagi pengembang aplikasi untuk membuat segmentasi pengguna berdasarkan fitur apa saja yang sering mereka pakai.

b. Firebase Cloud Messanging and Notifications

Salah satu kegunaan Firebase adalah menyediakan koneksi yang bisa diandalkan untuk semua pengguna baik Android, iOS dan website tanpa perlu biaya tambahan. Sangat keren, bukan? Kini pengembang aplikasi bisa menargetkan pesan sesuai dengan segmen pasar yang diinginkan.

c. Firebase Authentication

Berkat Firebase Authentication, pengembang aplikasi bisa mengautentikasi pengguna ke aplikasi lewat tampilan interfaces. Nantinya autentifikasi bisa memudahkan pengguna untuk masuk ke aplikasi yang terintegrasi dengan penyedia identitas gabungan populer lain.

d. Firebase Cloud firestore

Merupakan database fleksibel dan terukur yang memudahkan pengembang untuk membuat aplikasi yang powerfull, responsif dan bisa bekerja tanpa latensi koneksi jaringan internet. Layanan secara offline yang

bisa diakses pengguna lewat perangkat seluler atau website menjadi pilihan paling favorit di era digital ini.

e. Firebase Realtime Database

Merupakan database yang di-host melalui cloud sehingga pengembang bisa mengelola database dalam jumlah lebih besar. Pengguna aplikasi akan tetap menerima update terbaru saat ponsel yang offline terhubung ke jaringan internet berkat fungsi Firebase satu ini.

f. Firebase Hosting

Kegunaan Firebase hosting adalah menayangkan konten menggunakan koneksi aman, cepat dan mendukung pengembang mengimplementasikan aplikasi web dan menyajikan konten statis atau dinamis ke jaringan penayangan konten.

20. UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti

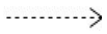
lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

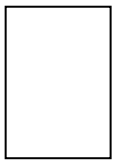
Adapun diagram UML yang digunakan dalam penelitian ini antara lain, sebagai berikut:

a. Use Case Diagram

Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah use case mempresentasikan sebuah interaksi antara actor dengan sistem. "Use case adalah sebuah gambaran dari fungsi sistem yang dipandang dari sudut pandang pemakai" (Schmuller, 1999:10).

Tabel 2. 3. Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri(independent).
3		Generalization	Hubungan dimana objek anak (descen-dent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
5		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

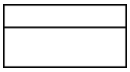
NO	GAMBAR	NAMA	KETERANGAN
6		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		Use Case	Deskripsi dari urutan aksi-aksi yang di-tampilkan sistem yang menghasilkan suatu hasil yang Terukur bagi suatu aktor
8		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

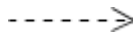
b. Activity Diagram

Menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, desicion yang mungkin terjadi, dan bagaimana alir aktivitas itu berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

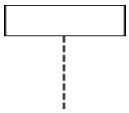
Tabel 2. 4. Simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing - masing kelas antar muka Saling berinteraksi satu sama lain.
2		Action	State dari sistem yang mencer-minkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek diben-tuk atau di awali.
4		Activity Final Node	Bagaimana objek dibentuk dan dihancurkan
5		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

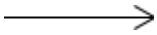
NO	GAMBAR	NAMA	KETERANGAN
1		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
2		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
NO	GAMBAR	NAMA	KETERANGAN

6		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (indepen-dent) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		Association	Apa yang menghubung-kan antara objek satu dengan objek lainnya

Tabel 2. 5. Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		LifeLine	Objek entity, antar muka yang saling berinteraksi.
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi- informasi tentang aktifitas yang terjadi

Tabel 2. 6. Simbol State Chart Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		State	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		Initial Pseudo State	Bagaimana objek dibentuk atau diawali
3		Final State	Bagaimana objek dibentuk dan dihancurkan
4		Transition	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		Node	Elemenfisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
7		Actifity Final Node	Bagaimana objek dibentuk dan dihancurkan

8		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

21. Implementasi Perangkat Lunak

a. Function Koneksi Peer

Gambar menunjukkan pseudocode implementasi dari function koneksi peer. Pada saat klien pertama kali membuka halaman, pengguna dihadapkan pada halaman login. Pengguna memasukkan username dan password agar bisa masuk pada halaman selanjutnya. Fungsi dari username adalah sebagai identitas inisialisasi koneksi peer. Dalam proses inisialisasi membutuhkan modul Peer yang terdapat dalam

Node.js serta port yang digunakan

Tabel 2. 7. Psudocode Inisialisasi Koniksi Peer

Input :-	
1	Load peer.js klien SEND id,
2	host modul peer, port
3	IF id sudah digunakan
4	Tidak bisa inisialisasi
5	ELSE
6	Koneksi peer terjadi
Output :-	

Penjelasan dari pseudocode inisialisasi koneksi peer adalah sebagai berikut :

1) Melakukan load peer.js klien yang ada pada aplikasi
2) Mengirimkan id, host modul peer, dan juga port yang digunakan untuk inisialisasi koneksi dengan modul peer yang ada pada server
3) Jika id yang dikirim kepada modul peer sudah digunakan, halaman web akan menampilkan pop-up bahwa id sudah digunakan.
4) Jika id belum digunakan, maka koneksi peer akan terjadi

b. Implementasi Mendapatkan Media Stream

Gambar menunjukkan pseudocode implementasi untuk mendapatkan media stream pengguna. Proses ini dilakukan setelah inisialisasi koneksi peer. Media stream yang didapat kemudian diletakkan pada halaman web. Tujuannya untuk menampilkan media stream yang ditangkap oleh kamera. Penjelasan pseudocode untuk mendapatkan media stream adalah sebagai berikut.

1) Memanggil fungsi getUserMadia() yang telah disediakan oleh peer.js klien.
2) Men-set konstrain media seperti resolusi dan frame rate.
3) Menampilkan stream pada halaman web.

Tabel 2. 8. Pseudocode Mendapatkan Media Stream

Input :-	
1	getUserMedia()

2	SET konstrain
3	IF sukses
4	Tampilkan dalam web
5	ELSE
6	Tampilkan pesan error
Output :-	

c. Implementasi Panggilan ke Peer Lain

Gambar menunjukkan pseudocode implementasi peer melakukan panggilan ke peer lain. Proses ini dilakukan untuk menghubungankan antara satu peer dengan peer lain. Saat peer melakukan panggilan ke peer lain, peer memerlukan identitas peer lain. Fungsi identitas yaitu agar STUN server dapat menemukan peer yang dimaksud dan agar tidak saling terturkar antara peer satu dengan yang lain.

Tabel 2. 9. Pseudocode Panggilan ke Peer Lain

Input :-	
1	Call (id, stream)
2	IF masih melakukan panggilan
3	Panggilan ditutup
4	Tampilkan stream dalam web
Output :-	

Penjelasan pseudocode untuk melakukan panggilan ke peer lain adalah sebagai berikut

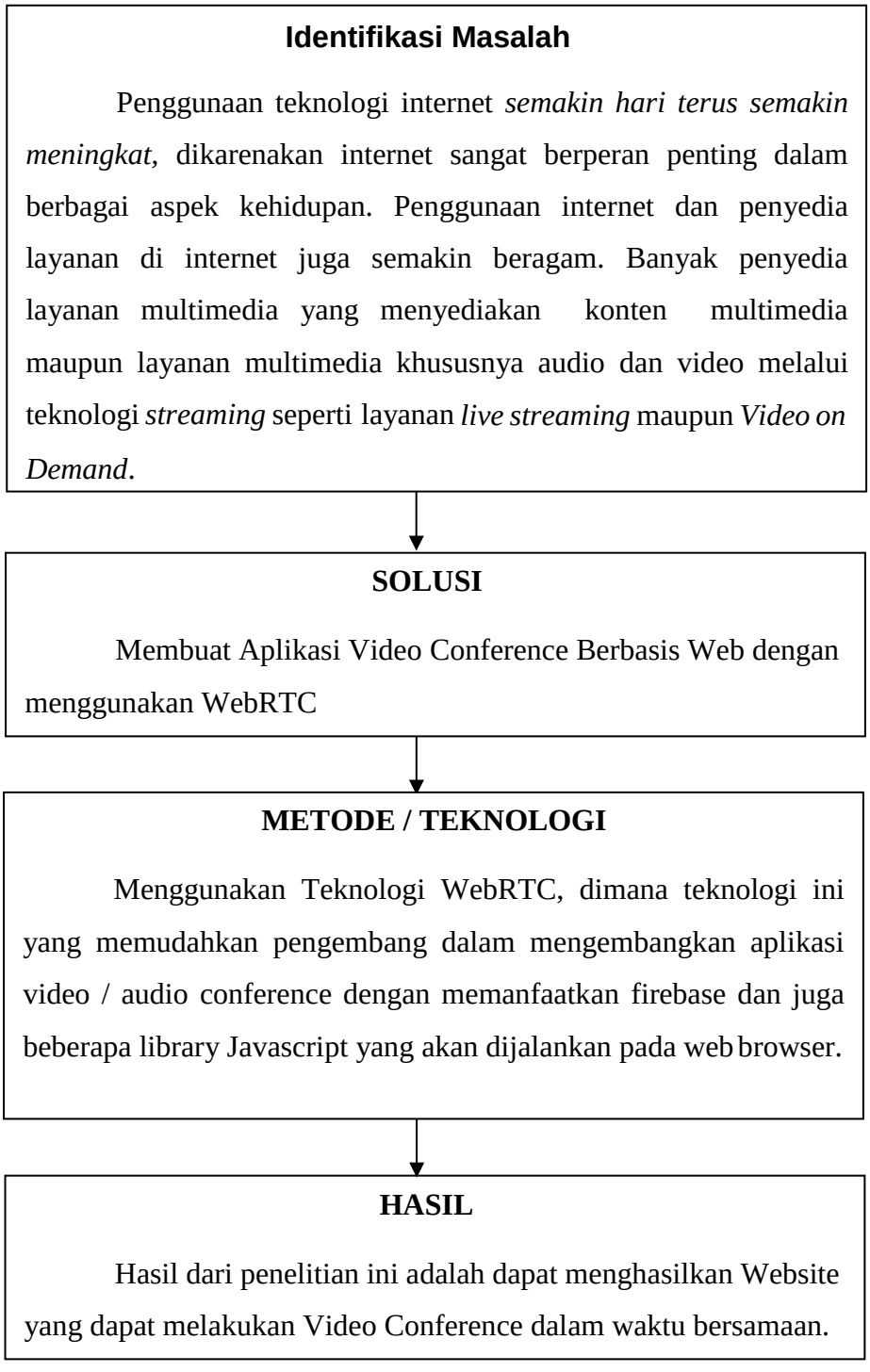
1) Memanggil fungsi call() yang mempunyai parameter id serta stream.

Id adalah identitas dari peer yang akan dituju. Stream adalah data yang akan dikirimkan kepada peer lain saat berhubungan.

2) Jika peer masih melakukan panggilan, maka panggilan yang sebelumnya akan ditutup.

22. Kerangka Pikir

Untuk memperjelas alur penelitian ini dibuatkan kerangka pikir sebagai berikut:



BAB III

METODOLOGI PENELITIAN

A. Lokasi Dan Waktu Penelitian

Kegiatan penelitian dilakukan di Kota Parepare di jalan Jendral Ahmad Yani KM 5 di kediaman Penulis. Waktu yang diperlukan penelitian ini ± 3 (tiga) bulan.

B. Jenis Penelitian

Metode yang digunakan dalam penelitian ini menggunakan metode penelitian dan pengembangan (Reasearch and Devlopment). Metode Penelitian dan Pengembangan adalah metode yang digunakan untuk menghasilkan suatu produk perangkat lunak dan menguji kelayakan perangkat lunak yang dibangun. Menurut" Sugiyono .2010, metode Penelitian dan Pengembangan adalah metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut.

C. Metode Pengumpulan Data

1. Sumber Data

   Adapun sumber data yang digunakan dalam penelitian ini adalah: studi pustaka, yaitu mencari informasi yang berasal dari buku-buku dan berbagai literature yang berkaitan dengan objek penelitian. Sumber tersebut digunakan sebagai landasan teori dan alat bantu dalam menganalisis.

   a. Data Primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber- sumber yang berada di lapangann yang berasal dari wawancara dengan pihak-pihak terkait dengan penelitian ini.

b. Data Sekunder

Data sekunder merupakan perlengkapan data primer, diperoleh dari sumber kepustakaan dan referensi berupa literatur, bahan kuliah, serta materi-materi yang berkaitan dengan penulisan ini

2. Metode Pengumpulan Data.

a. Metode observasi, dilaksanakan dengan melakukan pengamatan langsung dengan lembaga yang terkait untuk mengumpulkan informasi dan data yang dibutuhkan dalam menunjang permasalahan

b. Kepustakaan, melakukan penelitian pustaka dengan cara membaca, mempelajari, dan mengkaji literatur. Sejumlah referensi seperti buku-buku, dokumen, dan media lainnya yang berhubungan dengan pokok permasalahan.

D. Alat dan Bahan Penelitian

1. Alat Penelitian

a. Hardware (perangkat keras)

Perangkat keras yang digunakan untuk membuat aplikasi Video conference dapat dilihat pada table berikut :

Tabel 3. 1. Spesifikasi Perangkat Keras

Spesifikasi	
Merk	ASUS
Processor	Intel Core i5-8250U (1.6GHz)
Memory	4 GB
Harddisk	1000 GB

b. Software (perangkat lunak)

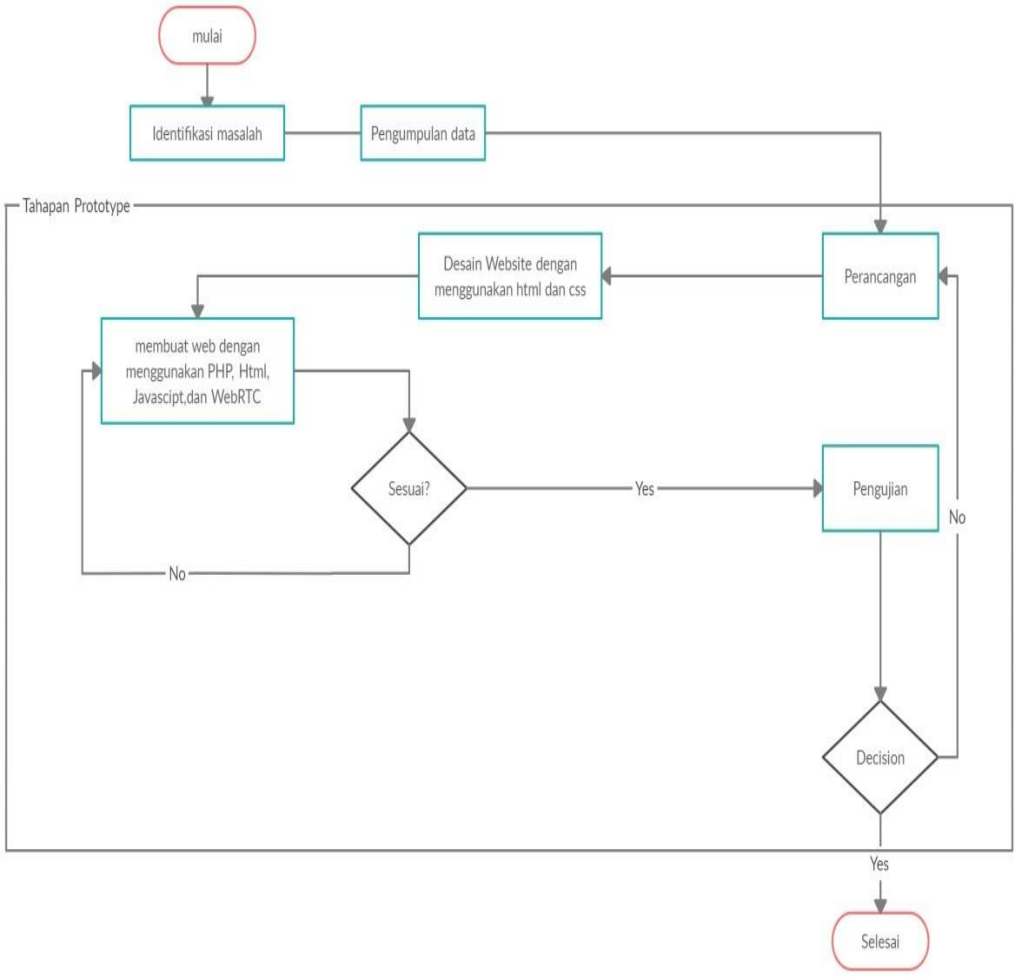
Perangkat lunak yang digunakan untuk membuat aplikasi Video conference yaitu :

Tabel 3. 2. Perangkat lunak

Spesifikasi	
Sistem Operasi	Windows 7 64 Bit
Aplikasi / Program	Visual Studio, Nodejs, webrtc, Firebase, GitHub.

E. Tahapan Penelitian

Tahapan- tahap penelitian yang dimaksud dalam penelitian ini adalh berkenan dengan proses pelaksanaan penelitian.



1. Tahap identifikasi masalah : Tahapan awal dari penelitian ini. Masalah yang di identifikasi adalah bagaimana membuat aplikasi Video conference berbasis web.
2. Tahap pengumpulan data : Mencari dan mengumpulan tahap ini dilakukan pengumpulan data yang didapat dari jurnal, artikel, internet dan penelitian lansung.
3. Tahap perancangan : Pada tahap ini akan di lakukan perencanaan atas aplikasi yang dibangun, dengan menganalisa kebutuhan berdasarkan skenario kebutuhan

pengguna terhadap aplikasi dan bagaimana itu semua dapat merespon interaksi dari pengguna.

4. Tahap mendesain web :pada tahap ini mendesain web dengan css dan HTML.
5. Tahap pembuatan program : pada tahap ini program di bangun dengan menggunakan text editor berupa Sublime Text 3 dan XAMPP,WebRTC, Javascript
6. Tahap Pengujian : Tahap ini merupakan tahap pengujian terhadap aplikasi yang dibuat agar dapat berjalan dengan semestinya. jika pada tahap pengujian ini aplikasi yang dibuat mengalami masalah, maka tahapan tersebut akan kembali ke tahap perencanaan. Untuk menganalisa kesalahan yang terjadi pada proses tersebut.

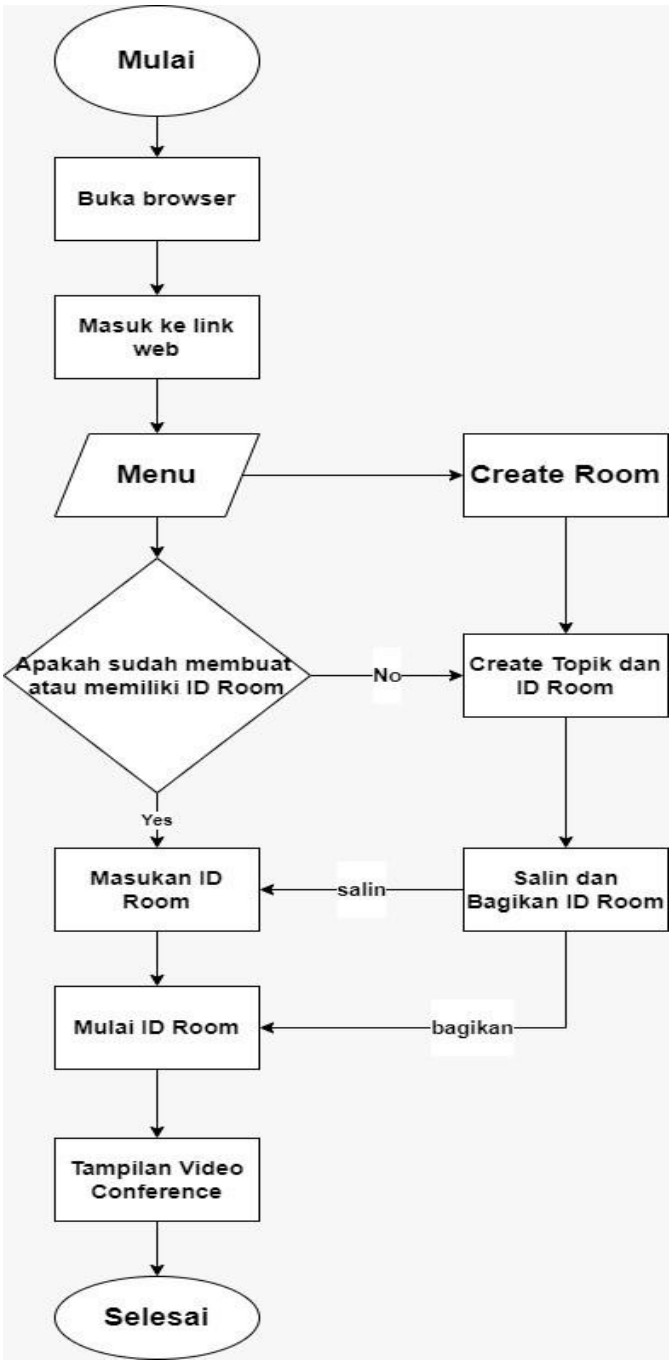
F. Metode Pengujian

Metode pengujian ini menggunakan metode Black box,yaitu pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dengan kata lain pengujian black box mengevaluasi hanya dari tampilan luarnya(interfacenya), fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui input dan output).

White box adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

G. Desain Sistem

1. Sistem yang di usulkan

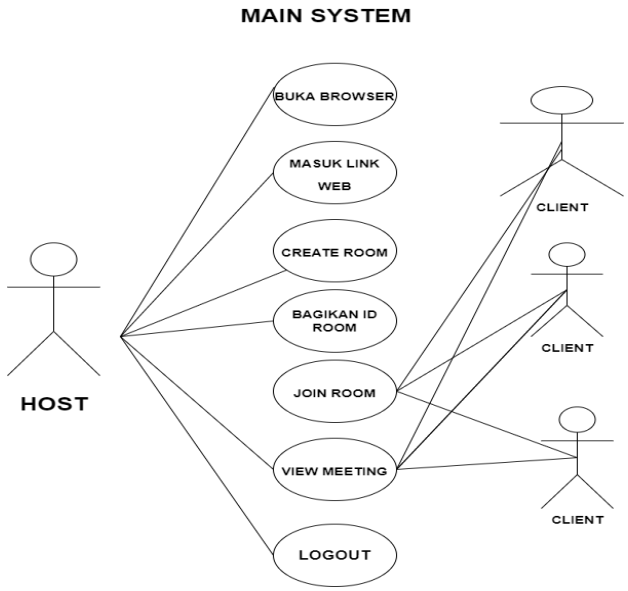


Gambar 3. 1. Flowchart Sistem yang diusulkan

Penjelasan Pada flowchart sistem pada user ini menjelaskan tentang bagaimana alur sistem yang akan dibuat penulis. Yang pertama membuka

browser lalu masuk ke situs web aplikasi di situs web aplikasi tersebut memiliki tiga menu yaitu Home dan create serta logout, pertama dimenu home itu adalah tempat jika kita ingin memulai atau join room meeting, jika sudah memiliki id room maka bisa langsung memasukan id room yang telah diketahui untuk memulai video conference sedangkan jika belum memiliki maka diharapkan membuat room pada menu create, didalam menu create kita diharuskan mengisi topik room , jika sudah membuat topik room maka id room pun akan tersedia lalu kita bisa menyalinnya dan memasukannya pada menu home yang dimana sebagai menu join atau memulai video conference. Didalam menu create room akan ada terdapat pilihan membagian id room ke client yang ingin join ke room yang telah dibuat setelah selesai masuk ke room yang telah dibuat maka akan muncul tampilan video conference.

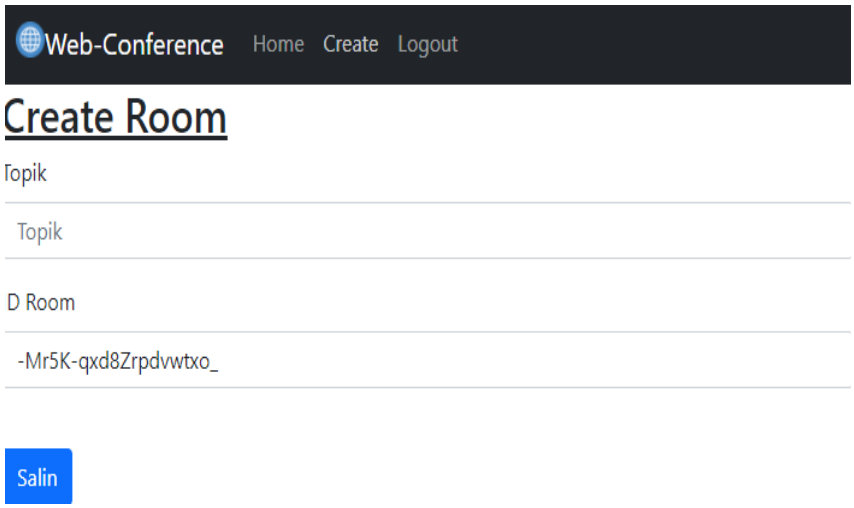
2. Sistem Yang berjalan



Gambar 3. 2. Diagram Use Case Sistem Yang Berjalan

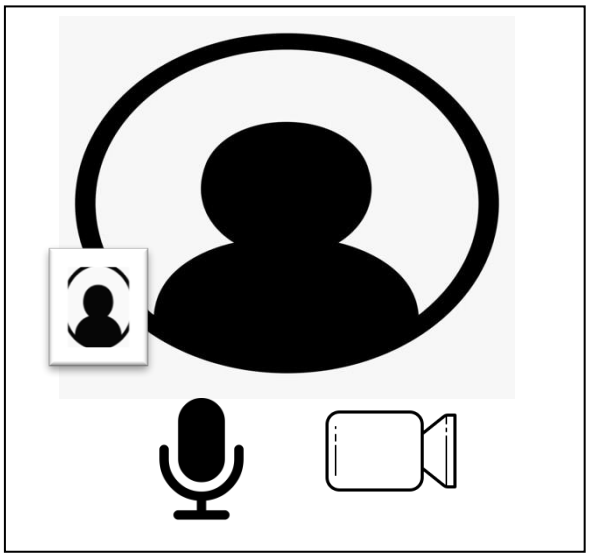
3. Rancangan Output

a. Menu Utama



Gambar 3. 3, Tampilan awal aplikasi

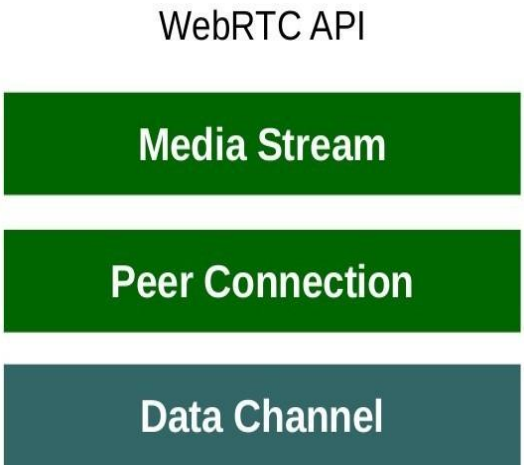
b. Video Call



Gambar 3. 4. Tampilan video call

4. Proses kerja WebRTC

   a. WebRTC API



Gambar 3. 5. Jenis API yg tersedia di WebRTC

   b. Media Stream



Gambar 3. 6.MediaStream yg digunakan

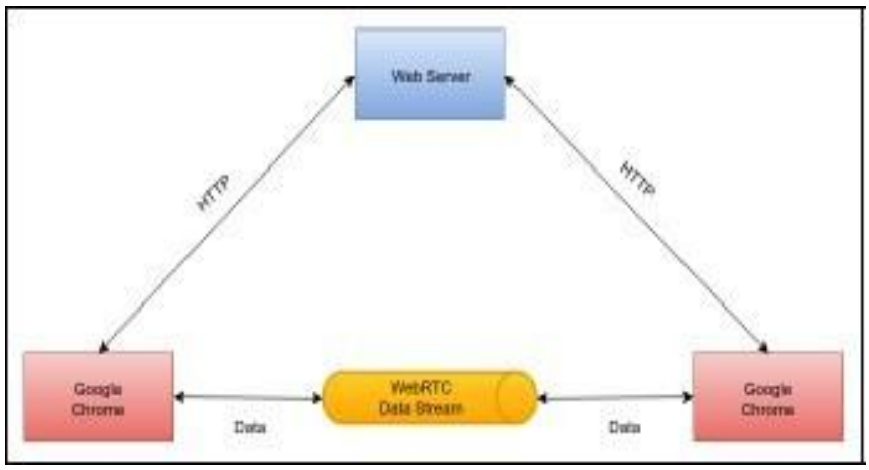
c. Cara mengkoneksikan media stream

```
var localVideo = document.querySelector('#local-video');


navigator.mediaDevices.getUserMedia({ video: true })
.then(function(stream) {
  localStream = stream;
  localVideo.srcObject = localStream;
})
.catch(function(error) {
  alert('Error: ', error);
});
```

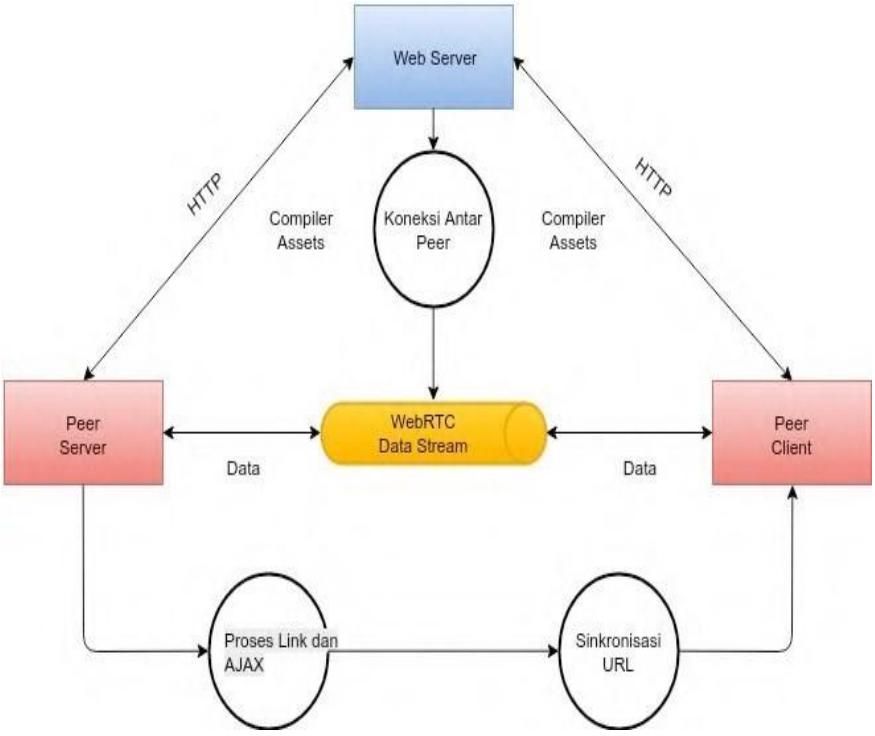
Gambar 3. 7. Untuk menghubungkan Media Stream

d. Diagram koneksi aplikasi



Gambar 3. 8. Diagram koneksi aplikasi

e. Diagram Alir data 0



Gambar 3. 9. Diagram Sistem Proses Awal

BAB IV

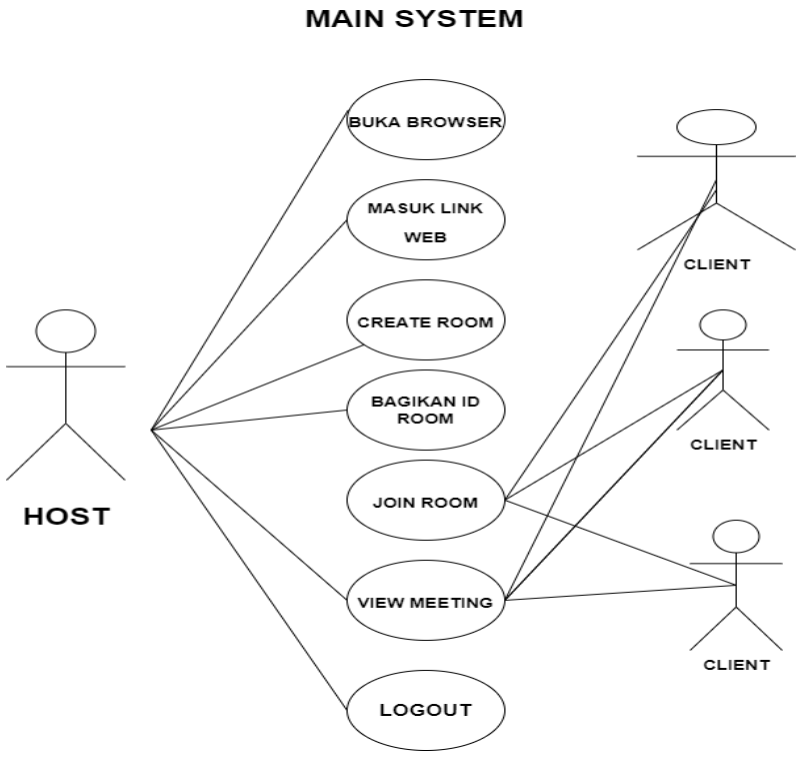
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data dengan UML

Analisis aliran data bertujuan mengetahui alur proses aplikasi video *conference* berbasis web ini. Dalam analisis sistem ini, penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar *system*.



Gambar 4. 1. Use Case Diagram

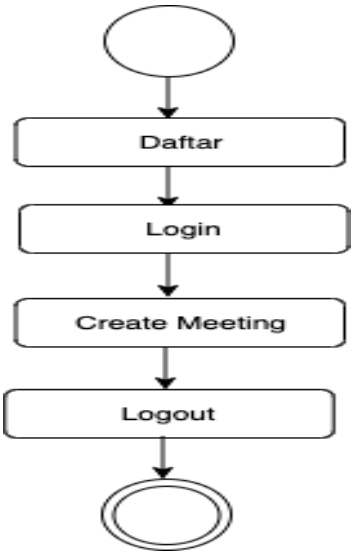
Tabel 4. 1. Penjelasan Usecase

Nama *Use Case*	Deskripsi *Use Case*
Daftar	Use case yang menjelaskan tentang proses host mendaftarkan jadwal meeting.
Login	*Use case* ini menjelaskan tentang proses *host login* dengan memasukkan Nama Lengkap.
Create Meeting	*Use case* ini merupakan proses *host* membuat sebuah jadwal untuk *meeting*.
Join Meeting	*Use case* ini menjelaskan prosen *client* bergabung di sebuah *meeting* yang telah dijadwalkan oleh *host*.
View Meeting	*Use case* ini menjelaskan prosen *client* saat telah bergabung di sebuah *meeting* yang telah dijadwalkan oleh *host*.
Kelola Data	*Use case* ini menjelaskan prosen admin mengelolah semua data *meeting*.
Logout	Use case ini menjelaskan prosen *host* mengakhiri *meeting*.

2. Activity Diagram

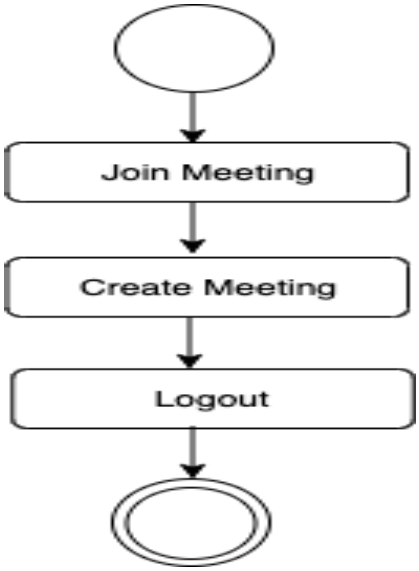
Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada video *conference* ini.

a. Activity Diagram Host



Gambar 4. 2. Activity Diagram Host

b. Activity Diagram Client

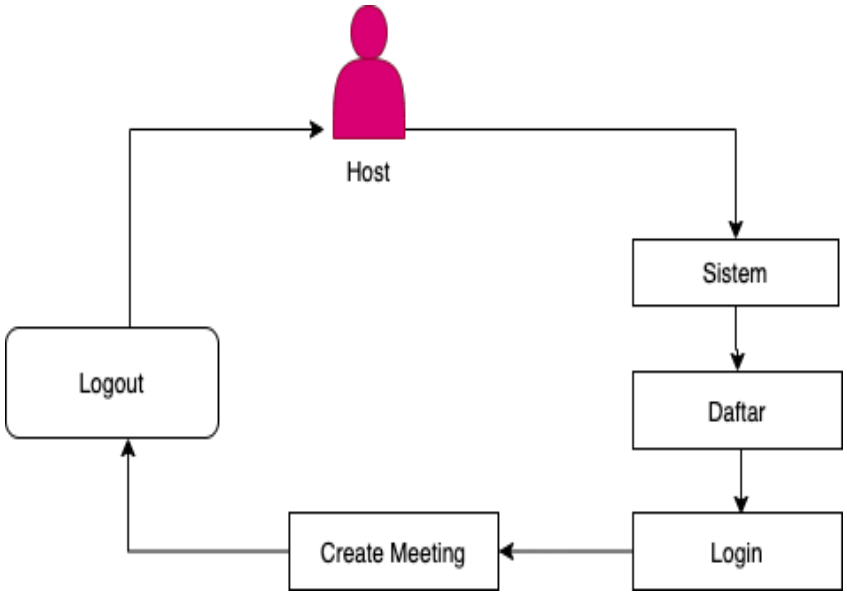


Gambar 4. 3. Activity Diagram Client

3. Sequence Diagram

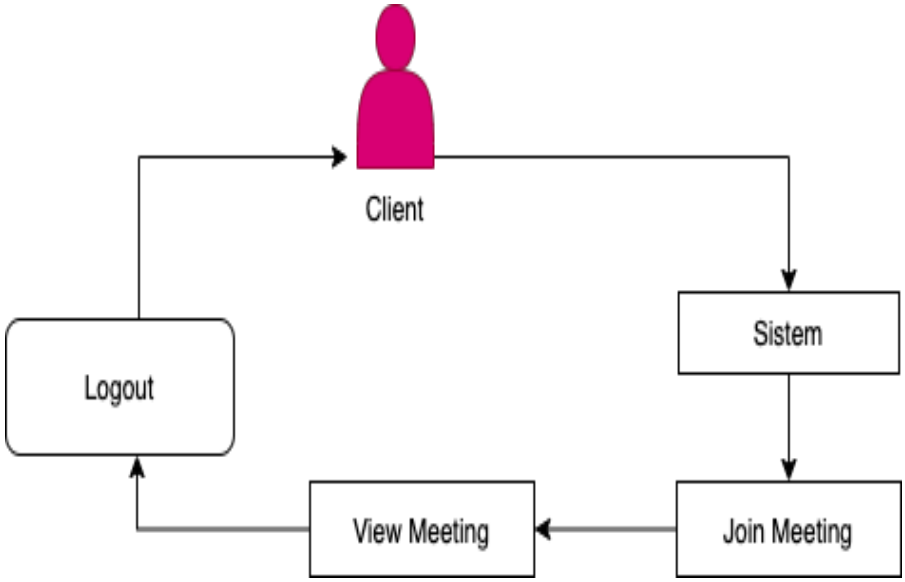
Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram sequencenya.

a. Sequence Diagram Host



Gambar 4. 4. Activity Diagram Client

b. Sequence Diagram Client

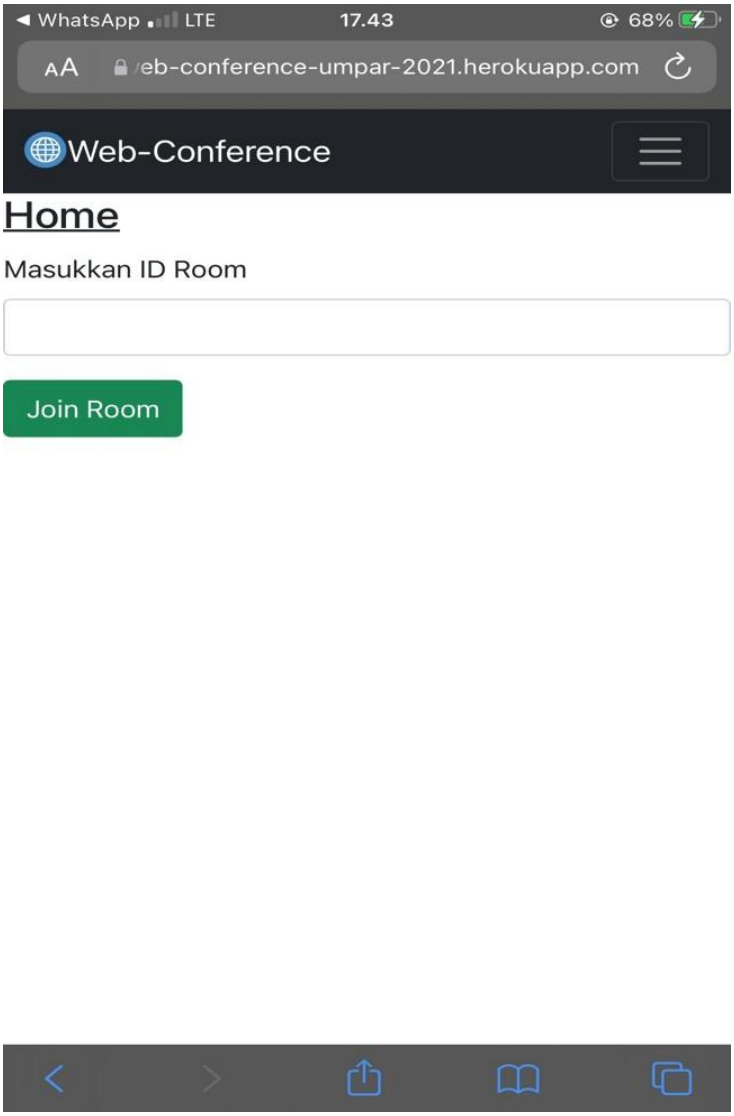


Gambar 4. 5. Sequence Diagram Client

B. Detail Aplikasi

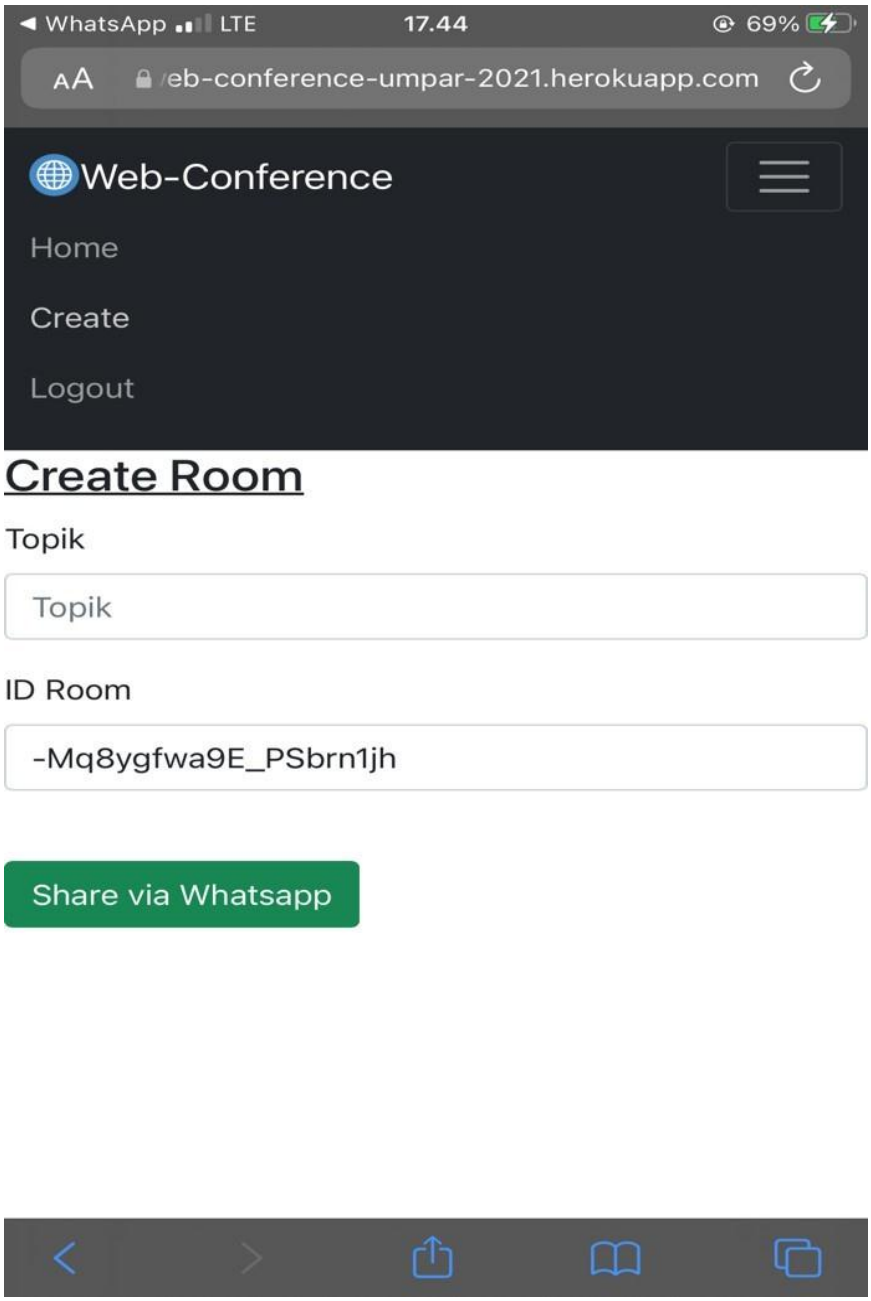
1. Halaman Awal Aplikasi

Merupakan halaman awal yang ditampilkan ketika membuka aplikasi, menampilkan judul aplikasi dan Masukkan Nama Lengkap.



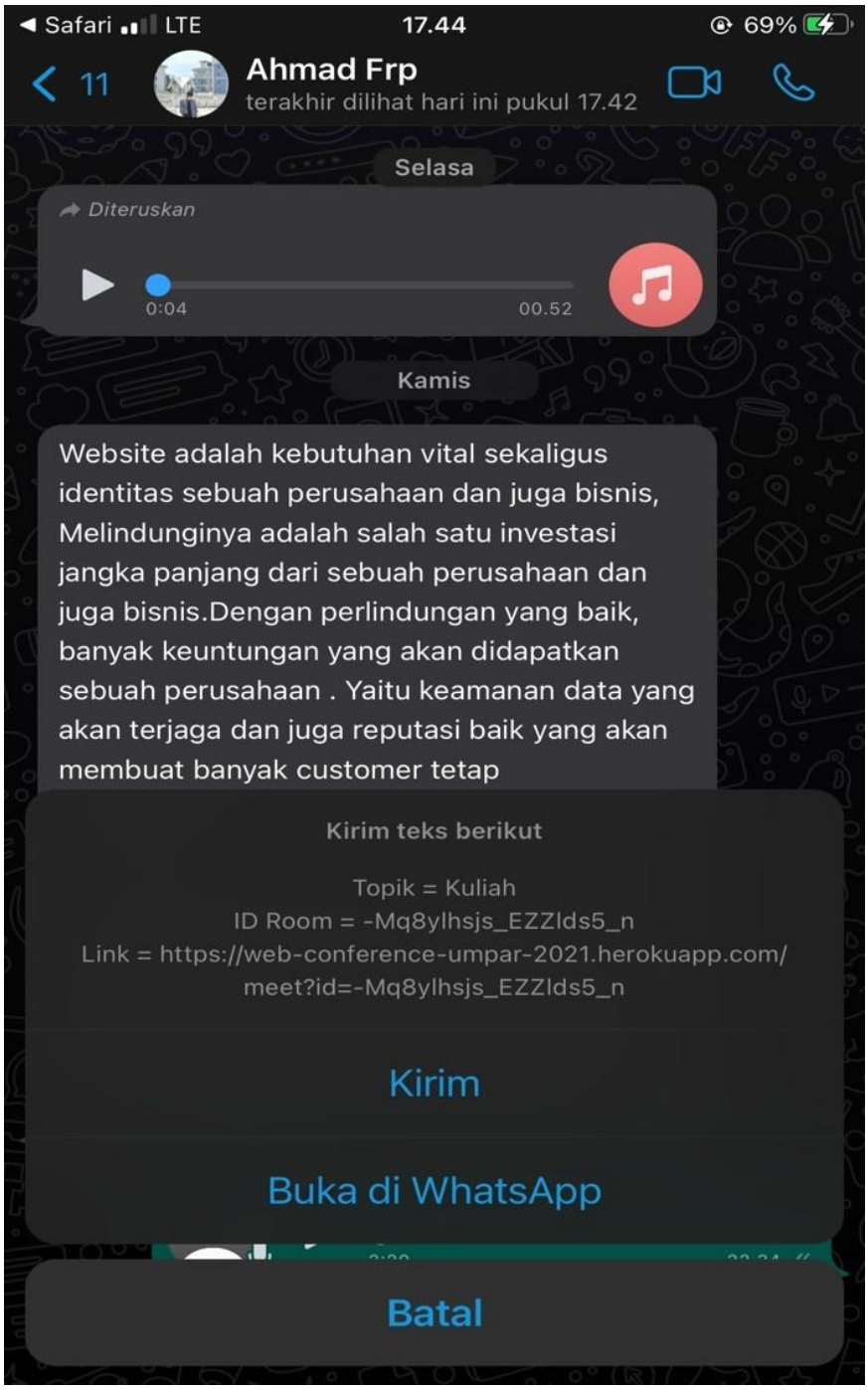
Gambar 4. 6. Halaman Awal Aplikasi

2. Halaman Create Meeting



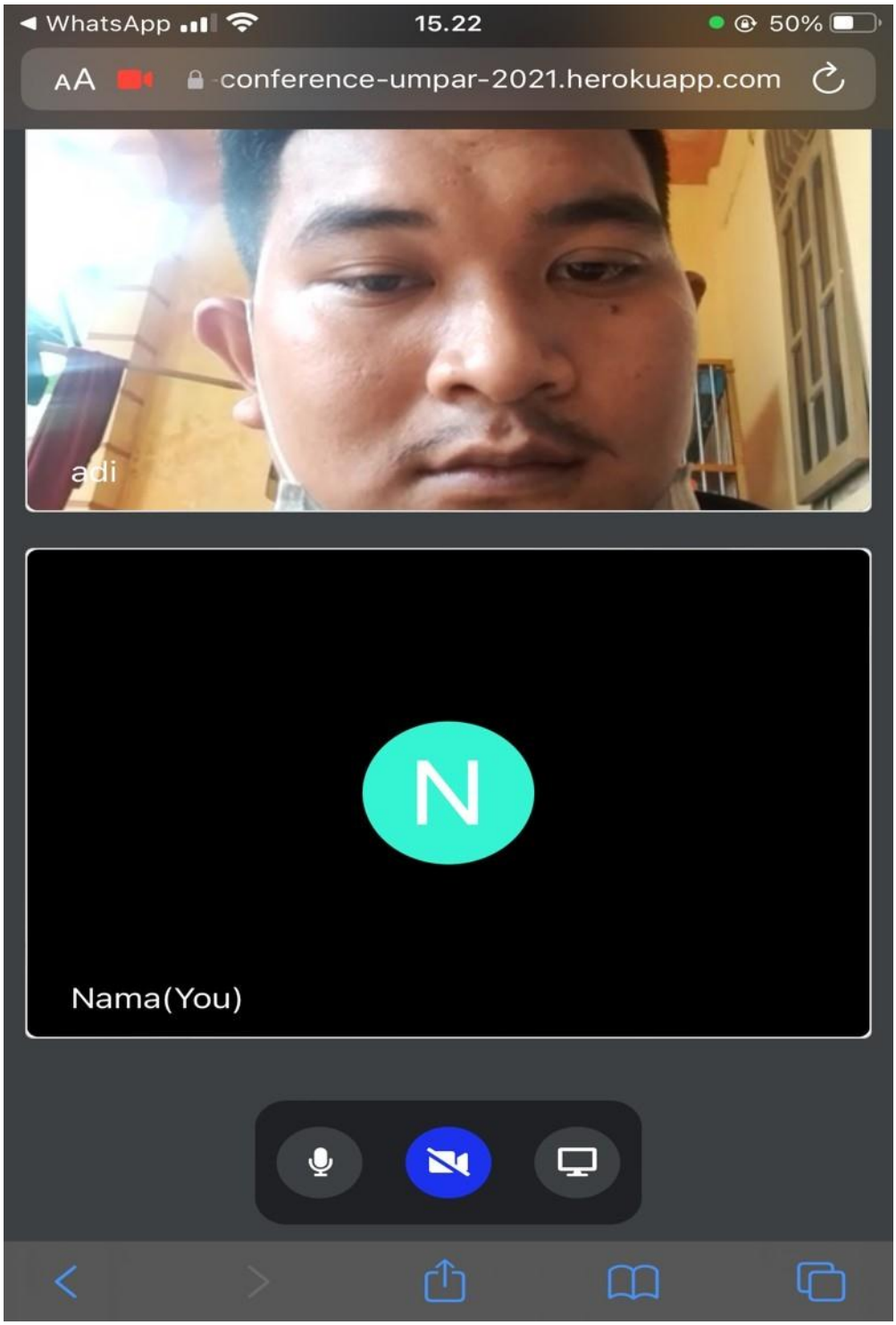
Gambar 4. 7. Halaman Create Meeting

3. Halaman Share Id Meeting



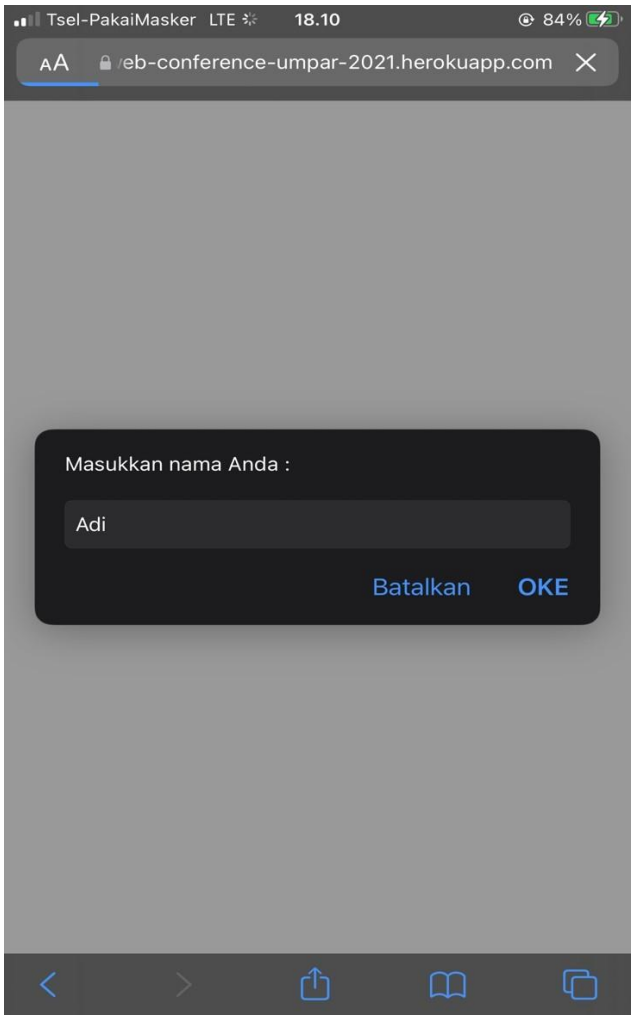
Gambar 4. 8. Halaman Share Id Meeting lewat Whatsaap

4. Halaman Meeting



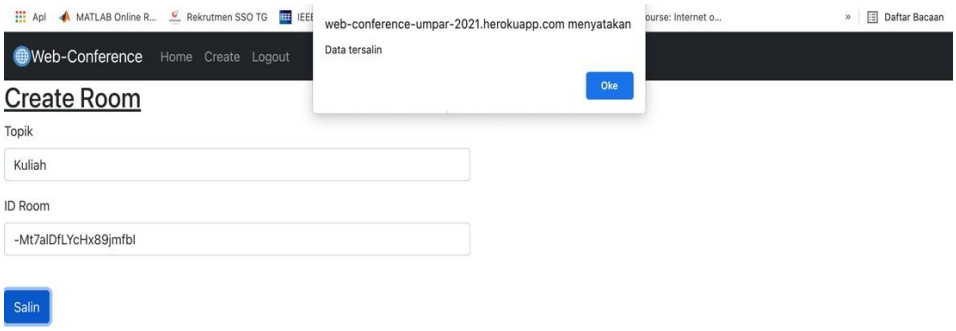
Gambar 4. 9. Halaman meeting

5. Halaman Join Meeting



Gambar 4. 10. Join Meeting

6. Halaman Menu Share Link Via PC



Gambar 4. 11. Halaman Menu Share Link Via PC

7. Halaman Menu Virtual Background



Gambar 4. 12. 1. Halaman Menu Virtual Background

C. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu bahasa pemrograman

Bahasa pemrograman yang digunakan untuk membuat aplikasi absen guru UPTD SD Negeri 68 Parepare ini adalah WEB RTC, PHP,CSS dan Javascript.

a. Kebutuhan perangkat keras

Spesifikasi minimum perangkat keras sebagai berikut :

Tabel 4. 2. Kebutuhan perangkat keras

Jenis	Spesifikasi
Notebook/Komputer	Acer Aspire E14
Processor	Intel Core i5-8250U (2.7 GHz)
Memory	8 GB RAM
Harddisk	1 TB

b. Kebutuhan perangkat lunak

Spesifikasi minimum perangkat lunak sebagai berikut :

Tabel 4. 3. Kebutuhan Perangkat Lunak

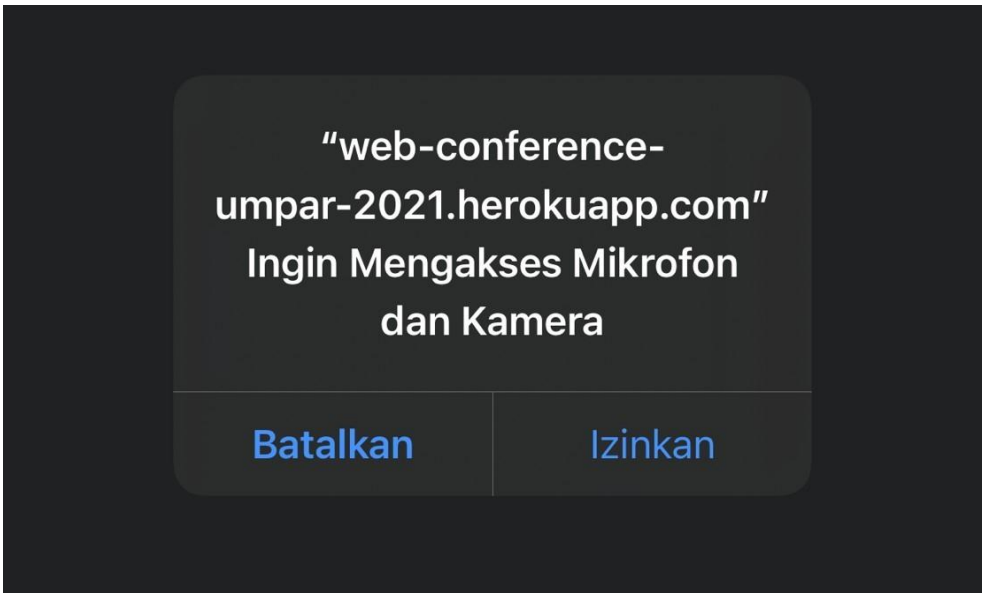
Sistem Operasi	Spesifikasi
Windows	Windows 10, x64-bit

D. PENGUJIAN SISTEM

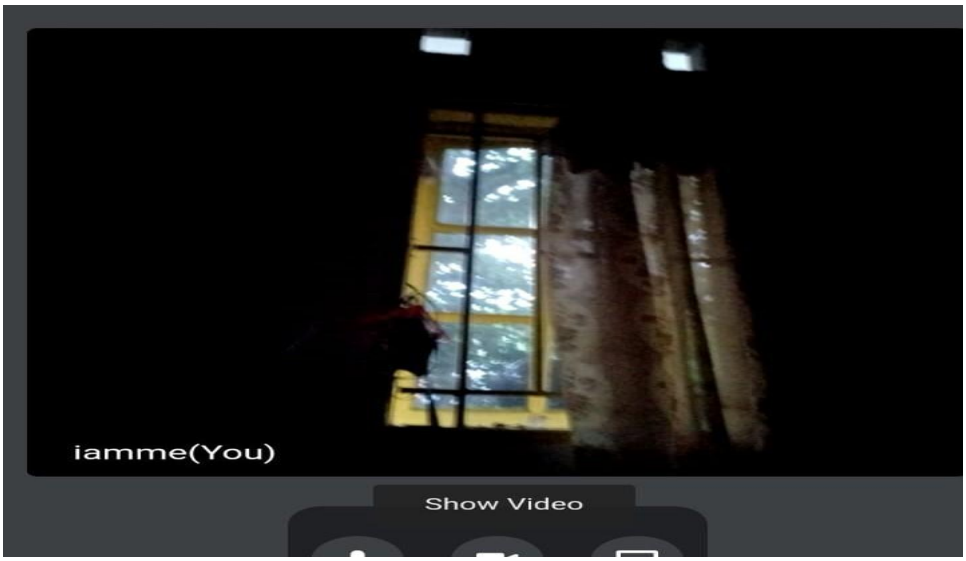
1. Black box

Pengujian aplikasi dilakukan dengan cara pengujian Black Box

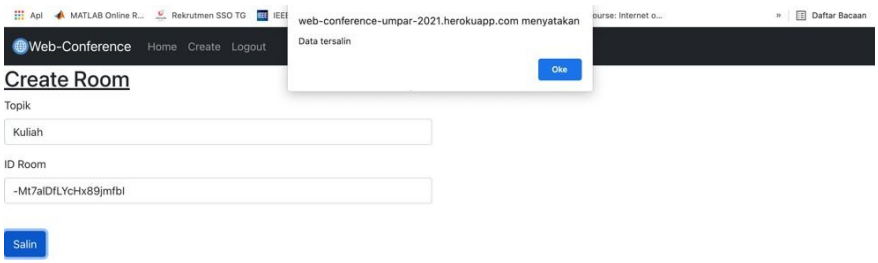
Tabel 4. 4. Pengujian Black Box Login

Test Faktor	Hasil	Kesimpulan
Halaman Login, masukkan nama pengguna.	√	Berhasil, muncul info aplikasi ingin mengakses mikrofon dan kamera.
Screen Shot		
		

Tabel 4. 5. Pengujian Black Box Menu Meeting

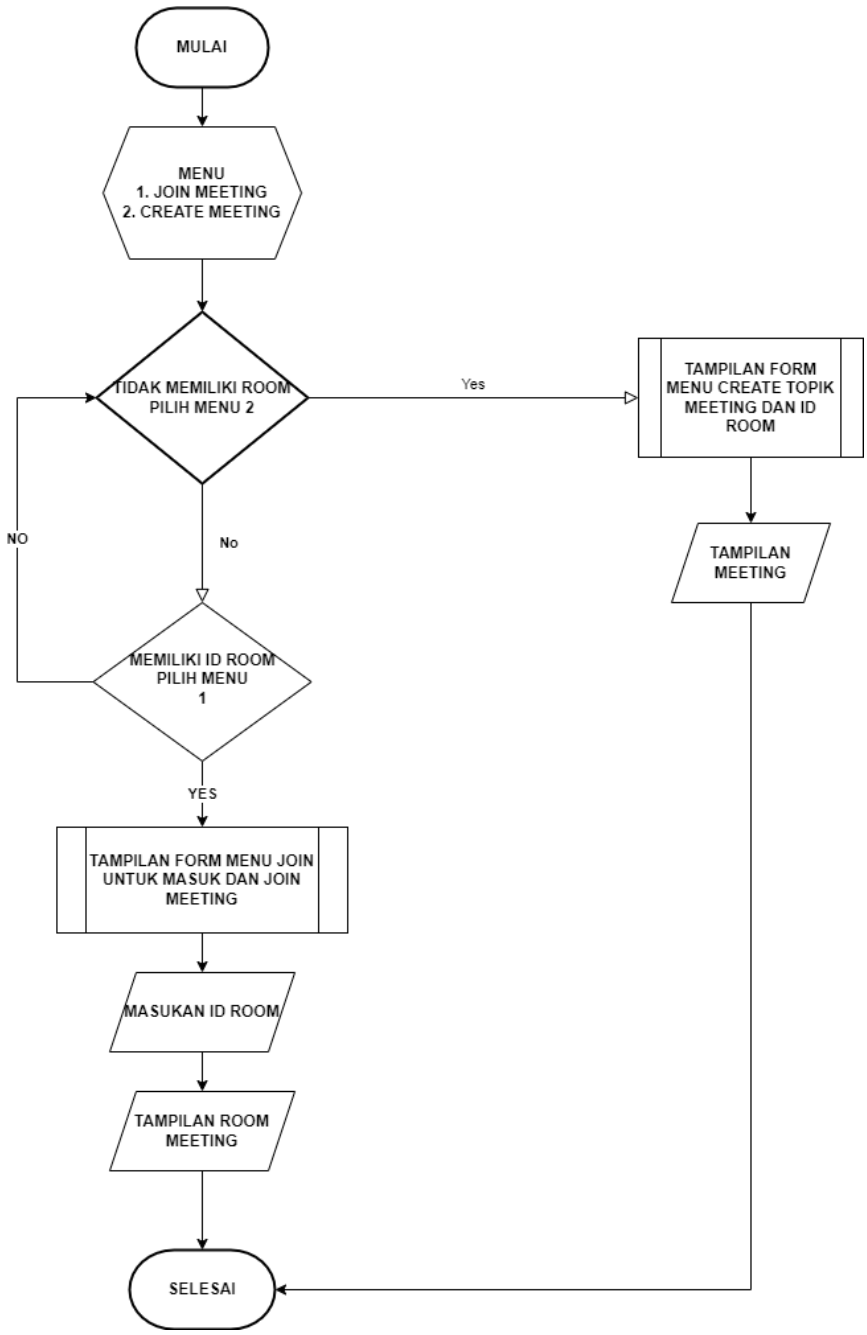
Test Faktor	Hasil	Kesimpulan
Halaman meeting, mengizinkan kamera dan mikrofon.	√	Berhasil, muncul gambar.
Screen Shot		
		

Tabel 4. 6. Pengujian Black Box Menu Share Link Via PC

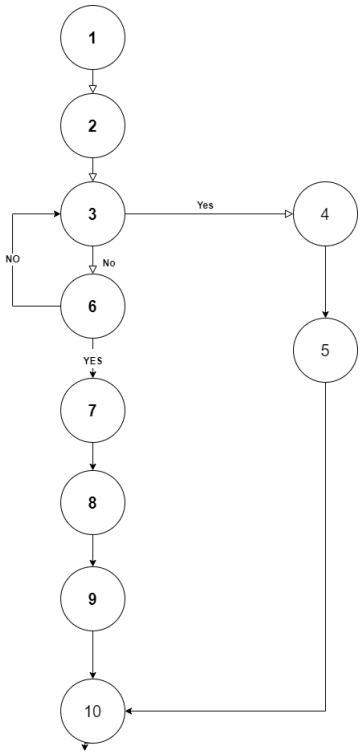
Test Faktor	Hasil	Kesimpulan
Halaman awal, membuat topik dan link.	√	Berhasil, muncul Data disalin.
Screen Shot		
		

2. White Box

a. Flowchart dan Flowgraph login Host



Gambar 4. 13. Flowchart Login Host



Gambar 4. 14. Flowgraph Login Host

Dari *flowgraph* menu login Host diatas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (edge) = 10

   N (Node) = 10

   P (Predikat Node) = 0

   Penyelesaian : V(G) = E – N + 2

   = 10 – 10 + 2

   = 2

   Predikat (P) = 1 + 1

   = 1 + 1

= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 2
3. Independent *path* pada *flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 5 – 10

   Path 2 = 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10

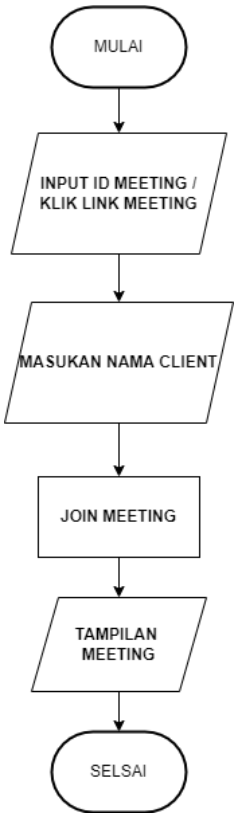
4. Grafik Matriks *Login Host*

Tabel 5. 1.

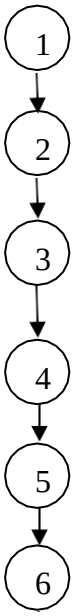
Tabel 4. 7. Grafik Matriks Login Host

	1	2	3	4	5	6	7	8	9	10	E –
1	1										1 – 1 = 0
2		1									1 – 1 = 0
3			1								1 – 1 = 0
4				1							1 – 1 = 0
5					1						1 – 1 = 0
6						1					1 – 1 = 0
7							1				1 – 1 = 0
8								1			1 – 1 = 0
9									1		1 – 1 = 0
10										1	1 – 1 = 0
											0 + 1 = 1

b. Flowchart dan Flowgraph Login Client



Gambar 4. 15. Flowchart Login Client



Gambar 4. 16 Flowgraph Login Client

Dari *flowgraph* menu login client diatas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (edge) = 6

   N (Node) = 6

   P (Predikat Node) = 0

   Penyelesaian : V(G) = E – N + 2

   = 6 – 6 + 2

   = 2

   Predikat (P) = 1 + 1

   = 1 + 1

   = 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 2

3. Independent *path* pada *flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 5 – 6

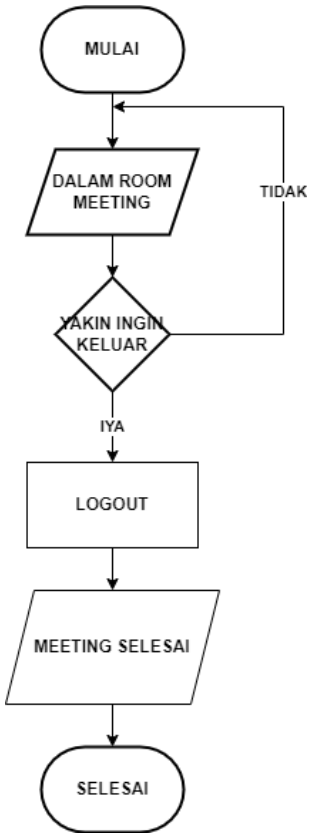
   Path 2 = 1 – 2 – 3 – 4 – 5 –3 – 4 – 5 – 6

4. Grafik Matriks *Login Client*

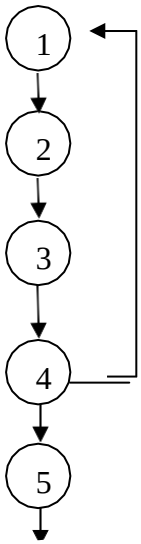
Tabel 4. 8. Grafik Matriks Login Client

	1	2	3	4	5	6	E – 1
1	1						1 – 1 = 0
2		1					1 – 1 = 0
3			1				1 – 1 = 0
4				1			1 – 1 = 0
5					1		1 – 1 = 0
6						1	1 – 1 = 0
SUM(E+1)							0 + 1 = 1

c. Flowchart dan Flowgraph Logout Host



Gambar 4. 17. Flowchart Logout Host



Gambar 4. 18. Flow6graph Logout Host

Dari *flowgraph* menu logout host diatas dapat dilakukan proses 7
perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (edge) = 5

   N (Node) = 5

   P (Predikat Node) = 0

   Penyelesaian : V(G) = E – N + 2

   = 5 – 5 + 2

   = 2

   Predikat (P) = 1 + 1

= 1 + 1

= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 2
3. Independent *path* pada *flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 5

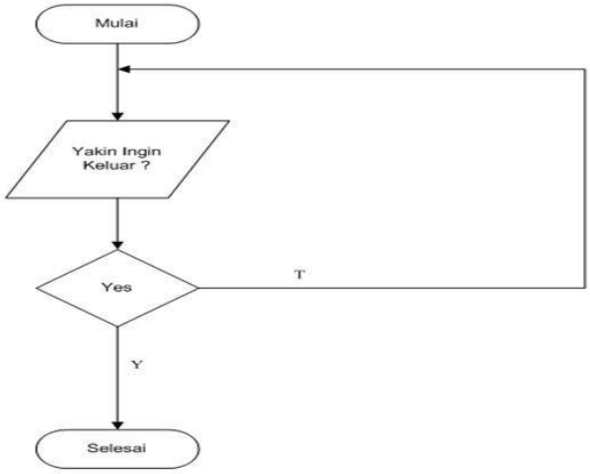
   Path 2 = 1 – 2 – 3 – 4 – 5 –3 – 4 – 5

4. Grafik Matriks *Logout Host*

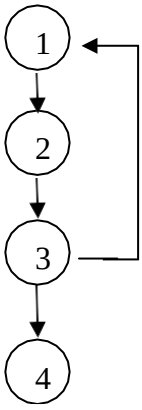
Tabel 4. 9. Grafik Logout Host

	1	2	3	4	5	E – 1
1	1					1 – 1 = 0
2		1				1 – 1 = 0
3			1			1 – 1 = 0
4	1			1		2 – 1 = 1
5					1	1 – 1 = 0
SUM(E+1)						1 + 1 = 2

d. Flowchart dan Flowgraph Log Out Client



Gambar 4. 19. Flowchart Log Out Client



Gambar 4. 20. Flowgraph logout client

Dari *flowgraph* menu logout client diatas dapat dilakukan proses perhitungan sebagai berikut:

5. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (edge) = 4

N (Node) = 4

P (Predikat Node) = 0

Penyelesaian : V(G) = E – N + 2

= 4 – 4 + 2

= 2

Predikat (P) = 1 + 1

= 1 + 1

= 2

6. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 2
7. Independent *path* pada *flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4

   Path 2 = 1 – 2 – 3 – 4 – 5 –3 – 4

8. Grafik Matriks *Logout client*

Tabel 4. 10. Grafik Logout Client

	1	2	3	4	E – 1
1	1				1 – 1 = 0
2		1			1 – 1 = 0
3	1		1		2 – 1 = 1
4				1	1 – 1 = 0
SUM(E+1)					1 + 1 = 2

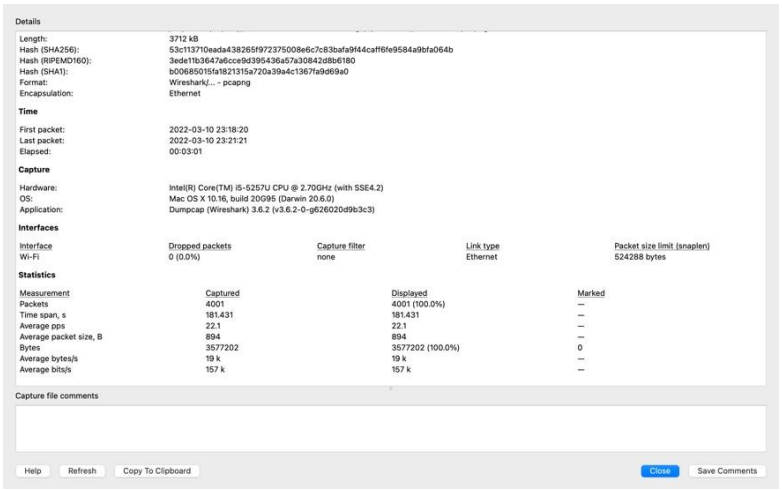
3. Hasil Pengujian

Tabel 4. 11. Tabel Hasil Pengujian

No	*Flowchart*	*Independent Path*	*Region*	*Cyclomatic Complexity*
1	*Login Host*	1	1	1
2	*Login Client*	1	1	1
3	*Logout Host*	2	2	2
4	*Logout Client*	2	2	2

4. Pengujian Parameter QoS *Video Conference* menggunakan *wireshark*

Pada hasil penelitian ini membahas mengenai hasil analisis dari pengujian parameter aplikasi video conference berdasarkan *Quality Of Service (QoS),* seperti *throughput, packet loss, delay* dan *Jitter* untuk mengetahui kinerja dari sistem tersebut.



Gambar 4. 21. Pengujian Quality Of Service (QOS) menggunakan wireshark

Berikut adalah cara pengambilan data beserta perhitungannya:

A. Throughput

Throughput yaitu kecepatan (rate) transfer data efektif, yang diukur dalam bps (bit per second). Throughput adalah jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut *(time span).*

Kategori *Throughput*	*Throughput*	Indeks
Bad	0 – 338 kbps	0
Poor	338 – 700 kbps	1
Fair	700 – 1200 kbps	2
Good	1200 kbps – 2,1 Mbps	3
Excelent	>2,1 Mbps	4

Gambar 4. 22. Kategori Throughput

$$\textbf{\textit{Throughput}} = \frac{\textit{(jumlah data yang dikirim)}}{\textit{(waktu pengiriman data)}}$$

Throughput = $\frac{3577202}{181.431.}$ $\frac{\text{bytes}}{\text{second}}$

= 19.716,5975 bytes/second

= 19.716,5975 KB

= 19.716,5975 x 8

Throughput = 157.733 KB/Second

B. Packet Loss

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena collision dan congestion pada jaringan.

Kategori *Packet Loss*	*Packet Loss*	Indeks
Poor	>25%	1
Medium	12 – 24%	2
Good	3 – 14%	3
Perfect	0 – 2%	4

Gambar 4. 23. Kategori Packet Loss

Adapun Rumusnya

$$Packet\ Loss = \frac{(Packet\ dikirim - Packet\ diterima)\ x\ 100\%}{Packet\ dikirim}$$

$$Packet\ Loss = \frac{(4001 - 3982)\ x\ 100\%}{4001}$$

$$= \frac{19\ x\ 100\%}{4001}$$

$$= \frac{1900\%}{4001}$$

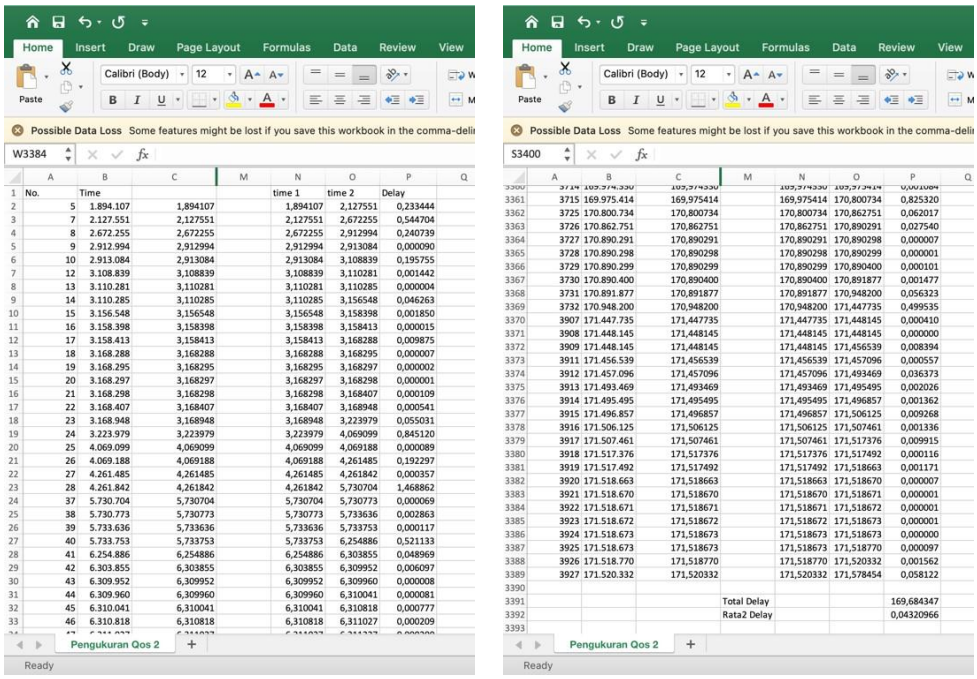
$$= 0{,}44988753\%$$

c. Delay

Delay (Latency) merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, congesti atau juga waktu proses yang lama.

Kategori Latency	Latency	Indeks
Poor	> 450 s	1
Medium	300 – 450 s	2
Good	150 – 300 s	3
Perfect	< 150 s	4

Gambar 4. 24. Kategori Delay (Latency)



No.	Time		time 1	time 2	Delay
5	1.894.107	1,894107	1,894107	2,127551	0,233444
7	2.127.551	2,127551	2,127551	2,672255	0,544704
8	2.672.255	2,672255	2,672255	2,912994	0,240739
9	2.912.994	2,912994	2,912994	2,913084	0,000090
10	2.913.084	2,913084	2,913084	3,108839	0,195755
12	3.108.839	3,108839	3,108839	3,110281	0,001442
13	3.110.281	3,110281	3,110281	3,110285	0,000004
14	3.110.285	3,110285	3,110285	3,156548	0,046263
15	3.156.548	3,156548	3,156548	3,158398	0,001850
16	3.158.398	3,158398	3,158398	3,158413	0,000015
17	3.158.413	3,158413	3,158413	3,168288	0,009875
18	3.168.288	3,168288	3,168288	3,168295	0,000007
19	3.168.295	3,168295	3,168295	3,168297	0,000002
20	3.168.297	3,168297	3,168297	3,168298	0,000001
21	3.168.298	3,168298	3,168298	3,168407	0,000109
22	3.168.407	3,168407	3,168407	3,168948	0,000541
23	3.168.948	3,168948	3,168948	3,223979	0,055031
24	3.223.979	3,223979	3,223979	4,069099	0,845120
25	4.069.099	4,069099	4,069099	4,069188	0,000089
26	4.069.188	4,069188	4,069188	4,261485	0,192297
27	4.261.485	4,261485	4,261485	4,261842	0,000357
28	4.261.842	4,261842	4,261842	5,730704	1,468862
37	5.730.704	5,730704	5,730704	5,730773	0,000069
38	5.730.773	5,730773	5,730773	5,733636	0,002863
39	5.733.636	5,733636	5,733636	5,733753	0,000117
40	5.733.753	5,733753	5,733753	6,254886	0,521133
41	6.254.886	6,254886	6,254886	6,303855	0,048969
42	6.303.855	6,303855	6,303855	6,309952	0,006097
43	6.309.952	6,309952	6,309952	6,309960	0,000008
44	6.309.960	6,309960	6,309960	6,310041	0,000081
45	6.310.041	6,310041	6,310041	6,310818	0,000777
46	6.310.818	6,310818	6,310818	6,311027	0,000209

3361	3715	169.975.414	169,975414		169,975414	170,800734	0,825320
3362	3725	170.800.734	170,800734		170,800734	170,862751	0,062017
3363	3726	170.862.751	170,862751		170,862751	170,890291	0,027540
3364	3727	170.890.291	170,890291		170,890291	170,890298	0,000007
3365	3728	170.890.298	170,890298		170,890298	170,890299	0,000001
3366	3729	170.890.299	170,890299		170,890299	170,890400	0,000101
3367	3730	170.890.400	170,890400		170,890400	170,891877	0,001477
3368	3731	170.891.877	170,891877		170,891877	170,948200	0,056323
3369	3732	170.948.200	170,948200		170,948200	171,447735	0,499535
3370	3907	171.447.735	171,447735		171,447735	171,448145	0,000410
3371	3908	171.448.145	171,448145		171,448145	171,448145	0,000000
3372	3909	171.448.145	171,448145		171,448145	171,456539	0,008394
3373	3911	171.456.539	171,456539		171,456539	171,457096	0,000557
3374	3912	171.457.096	171,457096		171,457096	171,493469	0,036373
3375	3913	171.493.469	171,493469		171,493469	171,495495	0,002026
3376	3914	171.495.495	171,495495		171,495495	171,496857	0,001362
3377	3915	171.496.857	171,496857		171,496857	171,506125	0,009268
3378	3916	171.506.125	171,506125		171,506125	171,507461	0,001336
3379	3917	171.507.461	171,507461		171,507461	171,517376	0,009915
3380	3918	171.517.376	171,517376		171,517376	171,517492	0,000116
3381	3919	171.517.492	171,517492		171,517492	171,518663	0,001171
3382	3920	171.518.663	171,518663		171,518663	171,518670	0,000007
3383	3921	171.518.670	171,518670		171,518670	171,518671	0,000001
3384	3922	171.518.671	171,518671		171,518671	171,518672	0,000001
3385	3923	171.518.672	171,518672		171,518672	171,518673	0,000001
3386	3924	171.518.673	171,518673		171,518673	171,518673	0,000000
3387	3925	171.518.673	171,518673		171,518673	171,518770	0,000097
3388	3926	171.518.770	171,518770		171,518770	171,520332	0,001562
3389	3927	171.520.332	171,520332		171,520332	171,578454	0,058122
3390							
3391				Total Delay			169,684347
3392				Rata2 Delay			0,04320966
3393							

Gambar 4. 25. Hasil CSV untuk menghitung delay

Adapun Rumusnya :

Delay = Time 1 – Time 2

Total Delay = 169,684347

Rumus Rata-rata Delay

Rata-rata delay = Total Delay / Total Paket yang diterima

Rata-rata delay = 169,684347/ 3927

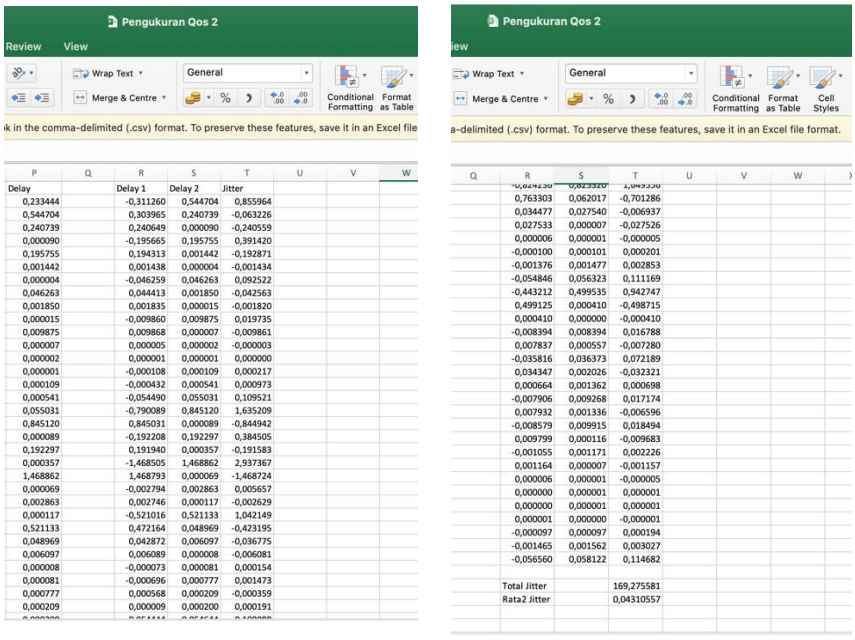
= 0,0432096631016043 second

d. Jitter

Jitter diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket diakhir perjalanan jitter.

Kategori *Jitter*	*Jitter*	Indeks
Poor	125 – 225 ms	1
Medium	75 – 125 ms	2
Good	0 – 75 ms	3
Perfect	0 ms	4

Gambar 4. 26. Kategori Jitter



Delay		Delay 1	Delay 2	Jitter
0,233444		-0,311260	0,544704	0,855964
0,544704		0,303965	0,240739	-0,063226
0,240739		0,240649	0,000090	-0,240559
0,000090		-0,195665	0,195755	0,391420
0,195755		0,194313	0,001442	-0,192871
0,001442		0,001438	0,000004	-0,001434
0,000004		-0,046259	0,046263	0,092522
0,046263		0,044413	0,001850	-0,042563
0,001850		0,001835	0,000015	-0,001820
0,000015		-0,009860	0,009875	0,019735
0,009875		0,009868	0,000007	-0,009861
0,000007		0,000005	0,000002	-0,000003
0,000002		0,000001	0,000001	0,000000
0,000001		-0,000108	0,000109	0,000217
0,000109		-0,000432	0,000541	0,000973
0,000541		-0,054490	0,055031	0,109521
0,055031		-0,790089	0,845120	1,635209
0,845120		0,845031	0,000089	-0,844942
0,000089		-0,192208	0,192297	0,384505
0,192297		0,191940	0,000357	-0,191583
0,000357		-1,468505	1,468862	2,937367
1,468862		1,468793	0,000069	-1,468724
0,000069		-0,002794	0,002863	0,005657
0,002863		0,002746	0,000117	-0,002629
0,000117		-0,521016	0,521133	1,042149
0,521133		0,472164	0,048969	-0,423195
0,048969		0,042872	0,006097	-0,036775
0,006097		0,006089	0,000008	-0,006081
0,000008		-0,000073	0,000081	0,000154
0,000081		-0,000696	0,000777	0,001473
0,000777		0,000568	0,000209	-0,000359
0,000209		0,000009	0,000200	0,000191

		0,763303	0,062017	-0,701286
		0,034477	0,027540	-0,006937
		0,027533	0,000007	-0,027526
		0,000006	0,000001	-0,000005
		-0,000100	0,000101	0,000201
		-0,001376	0,001477	0,002853
		-0,054846	0,056323	0,111169
		-0,443212	0,499535	0,942747
		0,499125	0,000410	-0,498715
		0,000410	0,000000	-0,000410
		-0,008394	0,008394	0,016788
		0,007837	0,000557	-0,007280
		-0,035816	0,036373	0,072189
		0,034347	0,002026	-0,032321
		0,000664	0,001362	0,000698
		-0,007906	0,009268	0,017174
		0,007932	0,001336	-0,006596
		-0,008579	0,009915	0,018494
		0,009799	0,000116	-0,009683
		-0,001055	0,001171	0,002226
		0,001164	0,000007	-0,001157
		0,000006	0,000001	-0,000005
		0,000000	0,000001	0,000001
		0,000000	0,000001	0,000001
		0,000001	0,000000	-0,000001
		-0,000097	0,000097	0,000194
		-0,001465	0,001562	0,003027
		-0,056560	0,058122	0,114682
		Total Jitter		169,275581
		Rata2 Jitter		0,04310557

Gambar 4. 27. Hasil CSV untuk menghitung jitter

Adapun Rumusnya :

Jitter = Delay 2 – Delay 1

Total Jitter = 169,275581

Rumus Rata-rata Jitter

Rata-rata Jitter = Total Jitter / Total Paket yang diterima

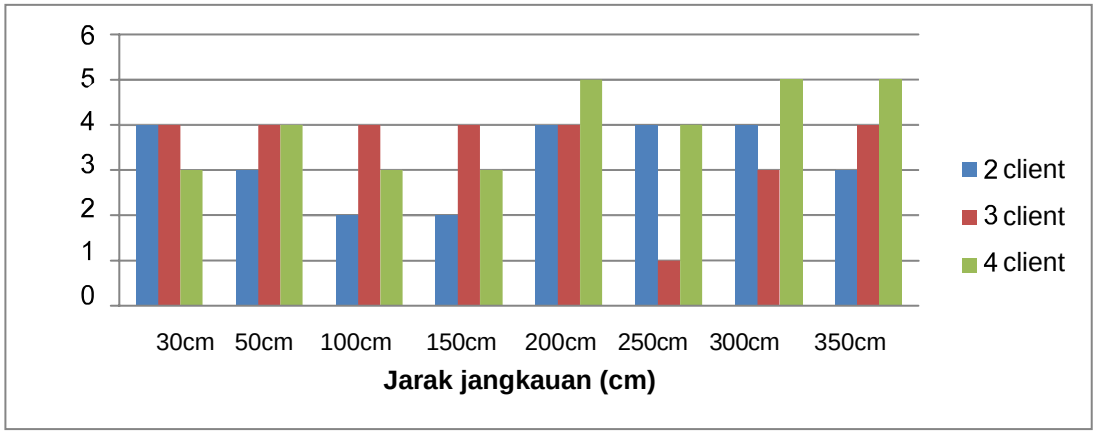
Rata-rata Jitter = 169,275581 / 3927

= 0,0431055719378661 ms

5. Pengujian kualitas percakapan pada *video conference*

Tabel 4. 12. Kategori MOS (Mean Opinion Source)

Kualitas percakapan	Nilai	index
Sangat baik	5	4
Baik	4	3
Cukup	3	2
Kurang baik	2	1
buruk	1	0



Gambar 4. 28. Grafik Perbandingan MOS Pada Banyaknya Client

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis untuk membuat aplikasi tersebut, maka ada beberapa kesimpulan sebagai berikut :

1) Aplikasi ini dapat memudahkan pengguna untuk melakukan video conference tanpa memikirkan batas waktu.
2) Aplikasi video conference in menggunakan sistem peer to peer yang dapat menghubungkan dua user atau lebih dalam satu room.
3) WebRTC dapat menyesuaikan real-time transfer untuk mengirim audio dan video yang ditransmisikan meskipun titik akses tidak menjamin QoS, dikarenakan kemampuan WebRTC membantu mengurangi *bandwidth* pensinyalan langsung menuju ke client tanpa routing proxy pada server
4) Aplikasi yang digunakan untuk mengakomodasi komuniksi video, audio , virtual background dan sharescreen
5) Pada aplikasi ini semakin banyak user yang bertukar data maka arsitektur atau bentuk networknya akan komplek sehingga Ketika salah satu user yang keluar pada saat meeting berlangsung maka user yang berada dibelakang user

yang keluar akan mengalami eror pada video meeting dan otomatis keluar dari meeting

6) Didalam aplikasi video conference ini memiliki fitur virtual background yang dapat mengganti latar belakang pada saat melakukan video conference.

B. Saran

Setelah selesai melaksanakan penelitian ini, maka penulis bisa memberikan saran demi perbaikan di masa mendatang yaitu :

Dalam penelitian ini penulis sendiri merasa banyak kekurangan dalam pembuatan aplikasi ini, seperti kurangnya sumberdaya pada sisi server yang terbatas, membutuhkan koneksi jaringan yang cepat dan stabil sehingga peneliti berharap apabila aplikasi ini akan dikembangkan agar di optimalkan disisi jaringan yang kurang stabil dan juga menambahkan fitur chat.

DAFTAR PUSTAKA

Atsani, K. L. G. M. Z. (2020). Transformasi media pembelajaran pada masa Pandemi COVID-19. Al-Hikmah, 1(1), 82-93.

A Sergiienko. (2014, May). WebRTC Blueprints: Develop your very own media applications and services using WebRTC. [On-line]. Available: http://4free- ebooks.com/ebook/WebRTC- blueprints/bibgphrgd

BackboneJS, ”Backbone.js”, [Online], https: //backbonejs.org/ diakses tanggal 24 Juni 2015.

Batra. S, ”AJAX - Asynchronous JavaScript and XML”, Salzburg: Information Technology and System Management, 2003.

Collins. N, McLean. A, Rohrhuber, and Ward. J, ”Live Coding in Laptop Performance”, Organized Sound 8(3), 2003, pp. 321-322

Edy Winarno, Ali Zaki dan SmitDev Community. 2014. Pemrograman Web Berbasis HTML 5, PHP, Dan JavaScript. Jakarta: Elex Media Komputindo

Escale Networks S.L, ”What is an API? Your Guide to the Internet Business (R)evolution”. 3scale, 4, 2011.

HandlebarJS, ”Handlebar.js”, [Online], http: //handlebarsjs.com/ diakses tanggal 24 Juni 2015.

Hidayatullah, S., Khouroh, U., Windhyastiti, I., Patalo, R. G., & Waris, A. (2020).Implementasi Model Kesuksesan Sistem Informasi DeLone And McLean Terhadap Sistem Pembelajaran Berbasis Aplikasi Zoom Di Saat Pandemi Covid-19. Jurnal Teknologi Dan Manajemen Informatika, 6(1), 44-52.

Huda, Miftakhul. 2010. Membuat Aplikasi Database Dengan JAVA, MySQL, dan Netbeans. PT. Elex Media Komput indo,

Ichwan M, 2011. “Pengukuran Kinerja Goodreads Application Programing Interface (Api) Pada Aplikasi Mobile Android (Studi Kasus Untuk Pencarian Data Buku)”. Institut Teknologi Nasional Bandung

Ilham, 2014 “Apa itu Video Conference?” Internet: http://in.vmeetsolutions.com/apa-itu-video-conference, 20 Nov, 2014

IETF, ”Web RTC Data Channel”, [Online] https://tools.ietf.org/html/ draft-ietf- rtcweb-data-channel-13# section-6.4, diakses tanggal 20 Juni 2015.

Joyent, Inc, ”About Node.js”, [Online], https://nodejs. org/en/about/, diakses tanggal 01 April 2015

K, Noviyanti, 2014. “Pengembangan Aplikasi Live Chat Dengan Menggunakan Webrtc Sebagai Pemanfaatan Teknologi Informasi Dan Komunikasi Untuk Media Pembelajaran Jarak Jauh (E-Learning)”.

Lina, Y., & Alfasan, S. K. (2013). Analisis dan Perancangan Aplikasi E-Meeting Berbasis Web. ComTech: Computer, Mathematics and Engineering Applications, 4(2), 1331-1339

Lina, Yeni, 2013. “Analisis Dan Perancangan Aplikasi E-meeting Berbasis Web”. Binus University

Loreto, S., & Romano, S. P, ”Real-Time Communication with WebRTC”, California: O’Reilly Media Inc,2014.

Mulki Rifky Akbar, “Pemanfaatan Web RTCuntuk membangun Aplikasi Video Conference”, Skripsi, 2014.

Ming Zhang, Jing Zhang, Wei Zheng, Feiran Hu, Ge Zhuang, ”A Self-description Data Framework for Tokamak Control System Design”, Fusion Engineering and Design, p. 5, 2015.

MongoDB, Inc, ”MongoDB”, [Online], https://www. mongodb.org/, diakses tanggal 08 April 2015.

Peerjs, ”Peejs-Server”, [Online],http://peerjs.com diakses tanggal 27 Mei 2015.

S. Tilkov and S. Vinoski, ”Node.js: Using JavaScript to Build High-Performance

Network Programs”, IEEE Computer Society Issue, vol. 6, pp. 80-83, 2010.

Sutanto, T. W. C. (2011). Pembangunan Aplikasi Text Chatting dan Video Chatting berbasis web. Program Studi Teknik Informatika. Fakultas Teknologi Industri. Universitas Atma Jaya Yogyakarta: Tugas Akhir.

Schmuller, Joseph. 1999. Sams Teach Yourself UML in 24 Hours. Pearson Education, Inc.

Sudarmin, Azrul, 2014. “Pengembangan Aplikasi Live Editor Menggunakan Webrtc Untuk Media Pembelajaran Jarak Jauh (E-Learning)”. UNHAS.

W3schools. 2019. “Pengertian HTML 5”. Tersedia dalam :<http://www.w3schools.com/html/html5_intro.asp> diakses tanggal 18 Februari 2020. Jakarta.

PeerJS enables WebRTC browser-to-browser. http://www.honline. com/open/news/item/PeerJS-enables-WebRTC-browser-tobrowser- banter-1804427.html. Diakses pada tanggal: 20 September 2011