

Implementasi Aplikasi Konferensi Video Berbasis WebRTC dengan Fitur Berbagi dan Rekam Layar

Beny Rahman Hakim, Abdullah Ardi, Budi Styawan, Yazid Aufa

Teknik Informatika, Politeknik Hasnur, Barito Kuala, Indonesia

Email: ¹benyrah@gmail.com, ²ardiofchemistry@gmail.com, ³budistyawan05@gmail.com, ^{4,*}yazid.aufar.ya@gmail.com

Email Penulis Korespondensi: yazid.aufar.ya@gmail.com

Abstrak—Konferensi video saat ini sudah lumrah digunakan pada berbagai aktivitas keseharian. Tersedia banyak sekali aplikasi untuk menyelenggarakannya, namun sebagian besar platform tersebut mengharuskan pengguna untuk berlangganan agar seluruh fitur dapat difungsikan. *Web Real-Time Communication* (WebRTC) adalah *framework* yang berguna untuk membangun komunikasi *real-time* (waktu nyata) melalui internet. Teknologi ini bekerja secara *peer-to-peer* dan dapat diterapkan dalam membangun aplikasi konferensi video *low cost* tanpa mengkhawatirkan limitasi fitur. Fasilitas berbagi dan rekam layar, yang pada kebanyakan aplikasi dibatasi penggunaannya, dapat dengan mudah dikerjakan oleh WebRTC. Penelitian ini bertujuan membangun aplikasi konferensi video berbasis kerangka kerja tersebut yang dapat membagikan dan rekam layar selama rapat berlangsung. Adapun fitur-fitur yang juga turut ditambahkan yakni kendali mikrofon/kamera, menampilkan nama pengguna, hitung jumlah partisipan, kirim pesan teks dan berbagi file. Pengembangan sistem mengacu pada metode *waterfall*, yaitu diawali tahap analisis hingga pengujian. Sistem kemudian diimplementasikan menggunakan bahasa HTML, JavaScript, CSS, dan kerangka kerja Node.js. Selanjutnya, dilakukan hosting terhadap sistem supaya bisa diakses via internet. Pengujian dengan metode *black box testing* menunjukkan bahwa semua fitur mampu dijalankan dengan baik. Selain itu, diketahui pula bahwa aplikasi mampu menyambungkan sebanyak 13 orang ke dalam rapat yang sama.

Kata Kunci: Konferensi Video; WebRTC; Berbagi Layar; Rekam Layar; Node.js

Abstract—Video conference is now commonly used in a variety of daily activities. There are many applications available to perform it, but most of these platforms require users to subscribe so that all features can be fully utilized. *Web Real-Time Communication* (WebRTC) is a framework that functions to establish a real-time communication over the internet. This technology works in a peer-to-peer basis and can be implemented in building low-cost video conference application without worrying about feature limitations. Screen sharing and recording facilities, which are restricted in most applications, can easily be accommodated by WebRTC. This research aims to build a video conference application based on WebRTC framework that is capable of sharing and recording screen during meeting. Additionally, some useful features are also added, such as microphone/camera control, displaying user names, counting the number of participants, sending text messages and sharing files. System development refers to waterfall method, which starts from analysis to testing steps. The language used for implementing the system include HTML, JavaScript, CSS and some Node.js libraries. After that, the application is deployed on a hosting server to make it accessible on the internet. The result of black box testing method shows that all features can run normally without problem. In addition, it is also known that the application is able to connect up to 13 people in the same meeting.

Keywords: Video Conference; WebRTC; Screen Sharing; Screen Recording; Node.js

1. PENDAHULUAN

Konferensi video (*video conference*) merupakan salah satu buah dari kemajuan teknologi informasi masa kini. Teknologi ini memungkinkan diadakannya rapat (*meeting*) secara virtual antara dua orang atau lebih guna membahas bermacam persoalan. Berbeda dari pertemuan (langsung) pada umumnya yang memerlukan penjadwalan waktu dan tempat, konferensi video menawarkan fleksibilitas terkait kapan dan di mana akan dilaksanakan. Asalkan ada koneksi internet, *meeting* siap diselenggarakan meski dengan zona waktu berlainan, serta dari belahan dunia manapun.

Saat ini telah banyak berkembang platform aplikasi untuk menyelenggarakan *video conference*. Beberapa diantaranya yang populer yakni aplikasi Zoom Meeting, Google Meet, dan Skype. Pada 2022, dari total 118 negara yang melakukan konferensi video, 80 diantaranya menggunakan Zoom, 28 negara memakai Google Meet dan 4 negara memanfaatkan Skype (Brandl, 2022). Di Indonesia sendiri, kenaikan pemakaian aplikasi ini sehari-hari meningkat sejumlah 31,7% di 2020 yakni selama era pandemi covid-19 (KumparanTech, 2020). Data-data tersebut mengindikasikan bahwa rapat virtual telah diterima dan diadopsi di berbagai sektor.

Terdapat fitur-fitur aplikasi konferensi video yang menjadikannya begitu digemari, dua diantaranya ialah *screen sharing* dan *recording*. Fitur *screen sharing* (berbagi layar) berguna untuk membagikan tampilan layar kepada pengguna lain, sedangkan *screen recording* (rekam layar) berfungsi untuk merekam seluruh aktivitas layar dan menyimpannya sebagai video. Hampir tiap-tiap platform *video conference* menyediakan kedua fasilitas tersebut, namun dengan beberapa batasan. Pada aplikasi Google Meet misalnya, hanya disediakan opsi untuk berbagi layar tetapi tidak ada menu perekaman. Lain halnya dengan Zoom versi *free*, kedua fitur tersebut sudah ada hanya saja penggunaannya dibatasi dengan durasi maksimal 40 menit per satu kali rapat. Kebanyakan aplikasi konferensi video pun tidak memberikan keleluasaan untuk dikustomisasi atau disesuaikan dengan kebutuhan pengguna. Oleh sebab itu, diperlukan sistem konferensi video yang bukan hanya andal dari segi fitur, tetapi juga dapat dimodifikasi menurut kebutuhan para pemakainya.

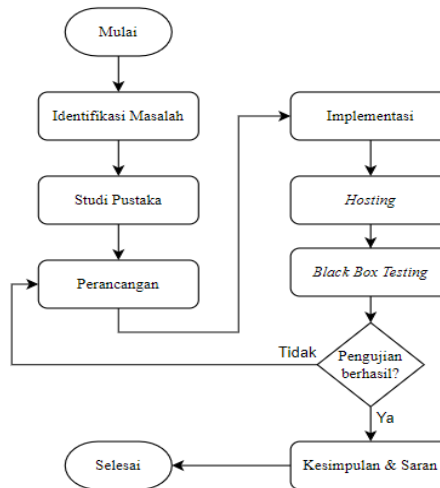
WebRTC (*Web Real-Time Communication*) berguna untuk membangun aplikasi *real-time* berbasis web. Dengan tanpa memungut biaya, kerangka kerja ini menawarkan banyak fitur yang diperlukan untuk pengembangan *video conference*. *Framework* ini menjamin keleluasan dalam kustomisasi sistem. Penelitian oleh Rahmanda dkk (2018)

mengembangkan kelas virtual berbasis WebRTC, dimana sistem tersebut mampu menghubungkan sebanyak 9 orang serta mengirim file dan pesan teks. Penelitian berikutnya dilakukan oleh Azzam dkk (2019) guna mengukur kinerja konferensi video berbasis WebRTC dalam melakukan *file sharing*. Selanjutnya, Amanda dan Haryanto (2020) melakukan penelitian tentang rancang bangun aplikasi *E-meeting* dengan WebRTC, dengan maksimum 2 pengguna bisa terhubung. Penelitian oleh Kasetwar, dkk. membahas tentang *video conference* berbasis WebRTC dengan fitur *screen sharing* (Kasetwar et al, 2022). Penelitian berikutnya dilakukan oleh Muath dan Naktal mengenai desain dan implementasi *peer-to-peer video and chat communication* antar 2 pengguna terhubung (Saed dan Edan, 2022). Berangkat dari permasalahan tersebut, penelitian ini dimaksudkan untuk mengembangkan sistem konferensi video berbasis WebRTC. Selain fitur berbagi dan rekam layar, pada sistem ditanamkan pula menu-menu pelengkap seperti kontrol mikrofon/kamera, menampilkan nama pengguna, hitung jumlah partisipan, kirim pesan teks dan berbagi file. Sebelum diuji fungsionalitasnya menggunakan metode *black box testing*, terlebih dahulu sistem di-*hosting* agar bisa diakses melalui internet. Diharapkan penelitian ini nantinya dapat menjadi acuan bagi siapapun yang berkeinginan membangun sistem *video conference* berbasis WebRTC.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilaksanakan dengan mengacu pada alur yang telah ditentukan. Hal ini dimaksudkan supaya penelitian menjadi lebih terarah dan memperoleh hasil optimal. Alur tersebut dapat dilihat pada Gambar 1 berikut.



Gambar 1. Tahapan Penelitian.

Pertama, dilakukan identifikasi masalah seputar aplikasi konferensi video berbasis WebRTC. Pada tahap ini, ragam masalah yang ditemukan kemudian dicarikan kemungkinan solusinya. Kedua, studi pustaka dikerjakan untuk mengumpulkan informasi-informasi relevan yang berguna sebagai solusi, diambil dari jurnal, buku dan situs-situs internet. Dari tahap ini, diketahui pula perbandingan penelitian dengan yang sebelumnya pernah dilakukan. Tahap ketiga ialah menentukan metode penelitian yang sesuai guna mencapai hasil akhir penelitian. Tahap perancangan, implementasi sistem, *hosting* dan *black box testing* dilaksanakan dengan maksud guna mempersiapkan sistem menjadi sematang mungkin.

2.2 Analisis Kebutuhan Fungsional

Kebutuhan fungsional sistem menggambarkan kemampuan sistem secara menyeluruh. Analisis ini berguna untuk mencantumkan seluruh fitur yang harus ada pada sistem. Adapun hasil analisis kebutuhan fungsional sistem *video conference* yang akan dibangun dapat diamati pada Tabel 1 berikut.

Tabel 1. Analisis Kebutuhan Fungsional

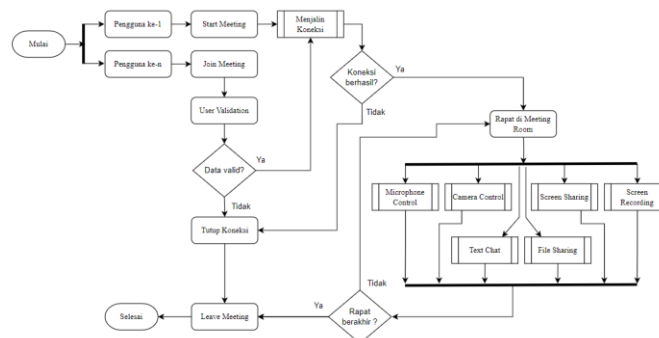
No.	Kebutuhan Fungsional	Keterangan	No.	Kebutuhan Fungsional	Keterangan
1.	<i>Start meeting</i>	Membuat rapat baru.	9.	<i>Screen sharing</i>	Berbagi layar kepada pengunjung lain.
2.	<i>Join meeting</i>	Bergabung ke <i>room</i> yang sudah ada.	10.	<i>Back to shared screen</i>	Mengembalikan <i>stream</i> dari main video menjadi <i>shared screen</i> .

3. <i>User validation</i>	Pemeriksaan <i>meeting invitation</i> dan <i>passcode</i> saat bergabung.	11. <i>Get shared screen on join meeting</i>	Mendapatkan layar yang dibagikan ketika pengguna baru bergabung di <i>meeting</i> .
4. <i>Show Mini video</i>	Video berukuran kecil yang menampilkan peserta rapat.	12. <i>Screen recording</i>	Merekam audio dan video pada layar komputer.
5. <i>Show User name</i>	Menampilkan nama para pengguna.	13. <i>Count participant</i>	Menampilkan jumlah seluruh peserta rapat.
6. <i>Display Main video</i>	Video ukuran yang lebih besar dan bisa <i>fullscreen</i> untuk melihat layar yang dibagikan.	14. <i>Text Chat</i>	Mengirim pesan teks di dalam <i>meeting</i> , baik secara publik (seluruh peserta rapat).
7. <i>Microphone control</i>	Menyalakan atau mematikan mikrofon.	15. <i>File sharing</i>	Berbagi file (dokumen, gambar, audio atau video) sehingga bisa diunduh peserta lain.
8. <i>Camera control</i>	Menyalakan atau mematikan <i>webcam</i> .	16. <i>Leave meeting</i>	Keluar meninggalkan <i>meeting</i> .

Berdasarkan Tabel 1 terlihat bahwa tersedia beragam fitur yang bisa digunakan pengguna dalam berkonferensi video. Pada sistem yang dibangun, aktor/user adalah siapapun yang mengakses sistem.

2.3 Diagram Alur Sistem

Diagram alur berguna untuk memahami cara kerja sistem yang akan dibangun. Gambar 2 berikut menunjukkan alur kerja sistem konferensi video berbasis WebRTC penelitian ini.



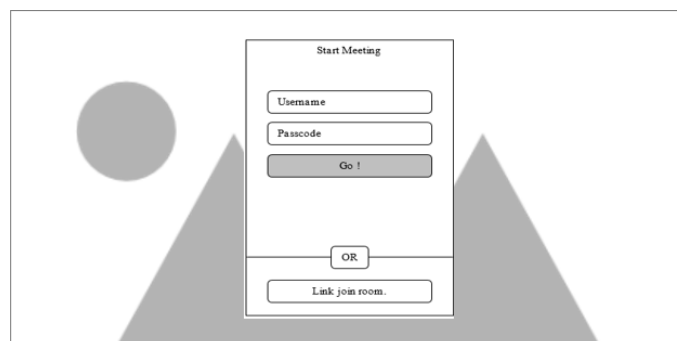
Gambar 2. Diagram Alur Sistem.

2.4 Desain Antarmuka Sistem

Sistem konferensi video yang dibangun mempunyai dua tampilan antarmuka, yaitu *frontpage* (halaman depan) dan *meeting room* (ruang rapat). Berikut dijelaskan mengenai peran dan rancangan dari kedua antarmuka tersebut.

1. *Frontpage*

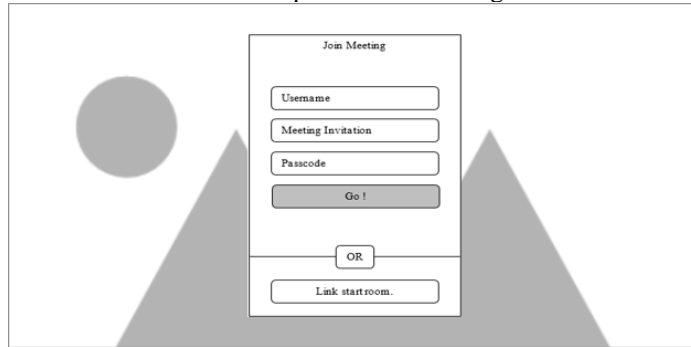
Saat pertama kali diakses, pengguna tidak langsung diarahkan kedalam rapat melainkan pada *frontpage*. Halaman ini berfungsi sebagai antarmuka yang menyediakan pilihan bagi user apakah ingin membuat rapat baru atau bergabung ke yang sudah ada. Gambar 3 di bawah ini mengilustrasikan desain antarmuka yang dimaksud.



Gambar 3. Formulir *Start Meeting*.

Secara *default*, formulir *start meeting* akan tampil ketika pengguna mengakses sistem. Antarmuka ini berfungsi untuk membuat rapat baru, dimana pengguna diharuskan input *username* (nama) dan *passcode* (kata sandi). Bila *user* bermaksud bergabung ke dalam suatu *meeting*, maka harus membuka formulir *join meeting* dengan menekan *link join room* yang tertera. Kedua formulir ini disajikan dalam satu halaman yang sama (*frontpage*), namun tetap ditampilkan satu per satu. Hal ini dimungkinkan dengan diterapkannya animasi *rotate* (berputar/rotasi), yang menyebabkan *start* dan *join meeting* saling berbelakangan. Animasi berjalan dan menampilkan *join room* saat tautannya diklik.

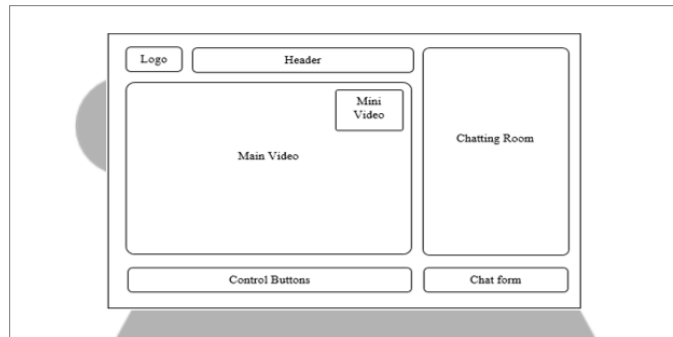
Formulir *join meeting*, seperti namanya, berfungsi untuk pengguna mengikuti rapat yang sudah ada. Perbedaannya dari *start meeting* ialah di dalamnya terdapat satu tambahan kotak input, yang disebut *meeting invitation* (undangan rapat). Undangan tersebut diperoleh saat pembuatan rapat baru. Dengan melakukan validasi terhadap kode undangan dan *passcode*, sistem dapat mengarahkan pengguna yang benar-benar valid ke dalam rapat atau menolak akses bagi yang tidak tervalidasi. Gambar 4 berikut memperlihatkan rancangan dari antarmuka *join room*.



Gambar 4. Formulir *Join Meeting*.

2. Halaman Meeting Room

Setelah berhasil membuat rapat baru atau bergabung ke yang sudah ada, pengguna akan diarahkan ke ruang rapat (*meeting room*). Di dalamnya tersaji beragam fitur (lihat Tabel 1) guna menunjang kegiatan konferensi video agar lebih interaktif.



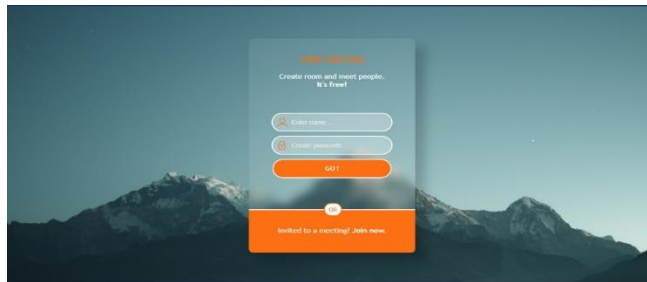
Gambar 5. Halaman *Meeting Room*.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi

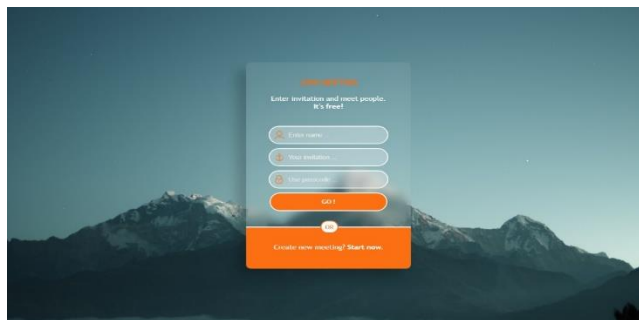
1. Halaman Frontpage

Bagian yang pertama diimplementasikan dari sistem konferensi video yang dibangun adalah *frontpage*, yang berfungsi sebagai antarmuka pembuka aplikasi. Setiap kali diakses, halaman inilah yang akan tampil. Sebagaimana diutarakan pada desain *interface*, pada halaman ini terdapat 2 formulir yakni *start meeting* dan *join meeting*. Keduanya ditampilkan secara berbelakangan, dan hanya akan terbuka bila pengguna mengakses *link* dari masing-masing formulir. Adapun hasil implementasi *start meeting* dapat dilihat pada Gambar 6 berikut ini.



Gambar 6. Implementasi *Start Meeting*.

Masih di halaman *frontpage*, hasil implementasi untuk formulir *join meeting* dapat dilihat pada Gambar 7 di bawah ini.



Gambar 7. Implementasi *Join Meeting*.

2. Halaman Meeting Room

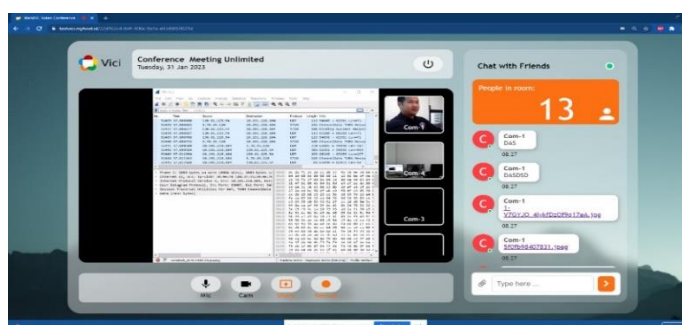
Meeting room merupakan halaman tempat para peserta rapat berkumpul dan berkomunikasi satu sama lainnya. Gambar 8 berikut memperlihatkan aktivitas video konferensi yang berlangsung antara dua orang. Terlihat pula menu-menu yang tersedia di dalamnya sebagai penunjang kegiatan *meeting*.



Gambar 8. Implementasi *Join Meeting*.

3.2 Pengujian Jumlah Pengguna

Pengujian berikutnya yang dilakukan adalah menentukan jumlah maksimum pengguna yang dapat terhubung di dalam rapat yang sama. Pengujian ini dilakukan secara online menggunakan 20 unit komputer yang dipinjam dari Laboratorium Komputer Politeknik Hasnur. Dari total keseluruhan komputer, ternyata aplikasi konferensi video hanya mampu menghubungkan sejumlah 13 orang pengguna saja. Hasil pengujian ini dapat dilihat pada Gambar 10 berikut.



Gambar 9. Hasil Pengujian Jumlah Pengguna.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa aplikasi konferensi video berbasis WebRTC mampu menjalankan seluruh fiturnya dengan baik. Sistem yang di-*hosting* sanggup menyambungkan 13 orang dalam satu rapat yang sama. Penelitian kedepannya dapat melakukan evaluasi kinerja dan analisis *Quality of Service* (QoS) guna mengetahui kelayakannya untuk pemakaian publik. Lebih lanjut, untuk *hosting* dapat digunakan server berspesifikasi lebih tinggi agar diketahui efeknya terhadap jumlah pengguna yang bisa terhubung. Selain itu, dapat pula dilakukan penambahan beberapa fitur pendukung, seperti deteksi pembicara, *virtual background*, *face recognition* dan *object detection*.

REFERENCES

- [1] Abdulghani, T. dan Gozali M. M. H., Sistem Konsultasi dan Bimbingan Online Berbasis Web Menggunakan WebRTC (Studi Kasus-Fakultas Teknik Universitas Suryakencana). Media Jurnal Informatika, Vol. 11 No. 2, 43-49. DOI: <https://doi.org/10.35194/mji.v11i2.1037>, 2019.
- [2] Ainanda, R. dan Haryanto E. V., Rancang Bangun Aplikasi E-meeting Menggunakan WebRTC (Web Real time Communication). Jurnal FTIK Vol. 1 No. 1, 220-228, 2020.
- [3] Alamsyah, F. dkk., Implementasi WebRTC pada Sistem Broadcast Pembelajaran untuk Menampilkan Bahasa Isyarat. JPTIHK, Vol. 3 No. 10, 10331-10336, 2019.
- [4] Azam, F. N. dkk., Implementasi Video Conference dengan File Sharing Menggunakan WebRTC. JPTIHK, Vol. 3 No. 10, 10102-10109, 2019.
- [5] Febrianda, dkk., Rancang Bangun Aplikasi Arsip Surat Menyurat Elektronik dengan Model Agile pada Kantor Desa Setia Karya Mandailing Natal, Jurnal BIT, Vol. 4 No. 1, Hal. 57 – 62, 2023.
- [6] Jansen, B., dkk., Performance Evaluation of WebRTC-based Video Conferencing. ACM SIGMETRICS Performance Evaluation Review, Vol. 45 No. 3. DOI: <https://doi.org/10.1145/3199524.3199534>, 2017.
- [7] Nawangsih, I. dan Sugeng B. R., Implementasi E-Desa untuk Meningkatkan Kualitas Pelayanan Masyarakat Teluk Buyung. Jurnal BIT, Vol. 3 No. 4, Hal. 380 – 384, 2022.
- [8] Rahmanda, R. Y. dkk., Perancangan dan Implementasi Kelas Virtual FILKOM Universitas Brawijaya dengan Memanfaatkan Teknologi WebRTC (Web Real-Time Communication). JPTIHK, Vol. 2 No. 7, 2721-2729, 2018.
- [9] Romadoni, J. dan Anjas S., Rancang Bangun Sistem Informasi Himpunan Mahasiswa Teknik Informatika Politeknik Hasnur Berbasis Web. Jurnal BIT, Vol. 3 No. 4, Hal. 419 – 424, 2022.
- [10] Saputra, R. A., Wilia I., Rancang Bangun Sistem Informasi Pengolahan Data Lapangan Futsal GOR Mampis Rungan Berbasis Web, Vol. 3 No. 4, Hal 301 – 307, 2022.
- [11] Yuniati, I. P. dkk., Analisis Kebutuhan Bandwidth Layanan Video Call pada WebRTC Berdasarkan Resolusi. Student Online Journal, Vol. 2 No. 1, 165-174, 2021.
- [12] Yulyanto, dan Rika N., Rancang Bangun Aplikasi Pembelajaran Desain Grafis Berbasis Multimedia Interaktif. Jurnal BIT, Vol. 3 No. 1, Hal. 43 – 47, 2022.
- [13] Loreto, S., dan Romano S. P., Real-Time Communication with WebRTC: Peer-to-Peer in The Browser. O'Reilly, USA, 2014.
- [14] Manson, R., Getting Started with WebRTC. Packt Publishing, Birmingham B3 2PB, UK, 2013.
- [15] Nakagawa, K., dkk., WebRTC-based Measurement Tool for Peer-to-Peer Applications and Preliminary Findings with Real Users, Proceedings of the 16th Asian Internet