

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Meningkatnya mobilitas masyarakat membuat kebutuhan komunikasi virtual yang dapat dilakukan tanpa harus bertatap muka secara langsung ikut meningkat agar tetap terhubung satu sama lain. Solusi yang dapat dilakukan adalah dengan mengadakan *e-meeting*. *E-meeting* merupakan sebuah kegiatan pertemuan yang dilakukan melalui media elektronik menggunakan perangkat lunak khusus (Ainanda et al, 2020).

Video conference adalah komunikasi audio dan video untuk menghubungkan pengguna satu dengan yang lain meskipun tidak berada di tempat dan waktu yang sama (Azzam et al, 2019). Dengan teknologi *video conference*, kegiatan rapat, konsultasi, atau kegiatan apapun menghimpun dua orang atau lebih dapat dilakukan dengan mudah.

Kebutuhan komunikasi virtual juga terjadi di bidang pendidikan khususnya di tingkat perguruan tinggi. Perbandingan jumlah mahasiswa dengan dosen pembimbing berdampak pada kurang optimalnya bimbingan yang diberikan dan tingkat produktivitas prodi yang menurun (Abdulghani et al, 2019). Konsultasi dan bimbingan daring menjadi salah satu solusi untuk menangani permasalahan

tersebut. Komunikasi *real-time* dapat dilakukan tanpa perlu bertatap muka secara langsung dengan fitur *live video stream* dan *live chatting*.

Teknologi *streaming* dapat dimanfaatkan untuk mendukung *distance learning* seperti kelas virtual (Rahmanda et al, 2018). Dengan pemanfaatan *WebRTC* pada pengembangan kelas virtual dapat mempermudah komunikasi antara dosen dengan mahasiswa selama kegiatan perkuliahan. Aktivitas seperti *broadcast*, *file sharing*, dan *text-chat* dapat dengan mudah dilakukan sehingga kegiatan perkuliahan dapat terlaksana dengan lebih efektif dan efisien. Hal ini juga memungkinkan untuk dilaksanakan *distance learning* sehingga perkuliahan tetap bisa terlaksana meskipun dosen dan mahasiswa tidak bertatap muka secara langsung.

Dengan perkembangan teknologi internet, muncul teknologi-teknologi yang berkembang yang mampu mengirimkan sinyal video secara langsung (Nashuha, A. 2018). Salah satu teknologi yang dapat dimanfaatkan adalah *WebRTC* yang memungkinkan komunikasi secara langsung berupa audio, video, atau teks melalui *web browser*. Kegiatan *live streaming* dapat memudahkan pengguna dalam berinteraksi dengan orang lain secara langsung melalui media video.

Menulis merupakan sebuah kegiatan menuangkan isi pikiran, gagasan, atau perasaan seseorang dalam sebuah tulisan (Baskara, E. C. 2018). Kegiatan menulis yang dilakukan oleh dua orang atau lebih secara bersamaan membutuhkan sebuah alat bantu yang berfungsi untuk menggabungkan tulisan-tulisan tersebut secara langsung dan bersamaan. Penerapan teknologi *WebRTC* pada aplikasi *live editor* dapat memungkinkan kegiatan menulis dilakukan oleh 2 orang atau lebih dapat dilakukan dengan mudah. Berikut ini adalah tabel perbandingan pustaka yang menjadi tinjauan dalam skripsi ini:

Tabel 2.1.1. Tabel Perbandingan Pustaka

No.	Peneliti	Tujuan	Target Pengguna	Hasil
1.	Ainanda, R., Haryanto, E. V. (2020)	Alternatif <i>meeting</i> konvensional	Masyarakat umum	Aplikasi <i>E-meeting</i>
2.	Azzam, F. N., Kartikasari, D. P., Bakhtiar, F. A. (2019)	Mengatasi perbedaan sistem operasi dan kebutuhan instalasi aplikasi untuk melakukan <i>video conference</i>	Masyarakat umum	<i>Video conference</i> dengan <i>file sharing</i>
3.	Abdulghani, T., Gozali, M. M. H. (2020)	Mengoptimalkan waktu bimbingan antar dosen dengan mahasiswa	Dosen dan mahasiswa	Sistem konsultasi dan bimbingan daring berbasis web
4.	Rahmanda, R. Y., Pramukantoro, E. S., Yahya, W. (2018)	Mengatasi permasalahan pembelajaran jarak jauh	Dosen dan mahasiswa	Kelas virtual
5.	Nashuha, A. (2018)	Mengembangkan teknologi <i>WebRTC</i> untuk aplikasi video secara langsung	Tenaga pendidik	Aplikasi web <i>video streaming</i>
6.	Baskara, E. C. (2018)	Mengembangkan alternatif aplikasi <i>editor text</i> menggunakan <i>WebRTC</i>	Masyarakat umum	Aplikasi <i>live editor text</i>

2.2. Dasar Teori

2.2.1. *WebRTC*

WebRTC (*Web Real-Time Communication*) merupakan sebuah framework yang bersifat *open source*. *WebRTC* dirilis pertama kali oleh Google dan telah

distandarisasi oleh *Internet Engineering Task Force (IETF)*, *Request for Command (RFC)*, dan *World Wide Web Consortium (W3C)*.

WebRTC memungkinkan pengguna web untuk melakukan komunikasi *real-time* berupa suara, video, dan teks secara *peer to peer* tanpa perlu instal berbagai macam plug-in dan perangkat lunak khusus (Rahmanda *et al*, 2018). *WebRTC* dapat diakses menggunakan *JavaScript Application Programming Interface (API)* yang bersifat *open source* dan dapat digunakan secara gratis (Azzam *et al*, 2019). *WebRTC* dapat menghubungkan berbagai macam platform dan perangkat dalam berkomunikasi menggunakan protocol yang tersedia di website resmi *WebRTC* (www.webrtc.org). Teknologi *WebRTC* masih dapat terus dikembangkan karena terbilang teknologi yang masih sangat baru. Belum semua platform dan penjelajah web dapat mendukung *WebRTC* (Abdulghani *et al*, 2019).



Gambar 2.1. Arsitektur WebRTC

Gambar 2.1. menggambarkan mengenai rancangan arsitektur sistem *WebRTC*. *Signalling server* bertugas sebagai koordinator dalam penginisialisasian komunikasi antar pengguna. Ketika pengguna A memulai sebuah panggilan, maka

pengguna A akan menghasilkan objek *session description* atau biasa yang disebut *SDP Offer* dan mengirimkan SDP tersebut ke *signalling server*. SDP itu sendiri berisi informasi tentang jenis media yang dikirim, format, *transfer protocol* yang digunakan, *endpoint* dari *IP address* dan *port*, serta informasi lainnya yang dibutuhkan untuk mendeskripsikan sebuah *media transfer endpoint*. Lalu, *signalling server* yang telah menerima *SDP Offer* dari pengguna A, akan mengirimkannya ke pengguna B. Setelah pengguna B menerima *SDP Offer* dari pengguna A, pengguna B akan menerima "*offer*" tersebut atau menerima panggilan dan menghasilkan objek *session description*-nya sendiri atau biasa yang disebut *SDP Answer*. *SDP Answer* yang sudah terbentuk tadi juga akan dikirim ke *signalling server*, dan *signalling server* akan mengirimkannya lagi ke pengguna A alias pemanggil, sehingga sekarang pengguna A dan B sudah bertukar SDP-nya masing-masing dan sebuah *peer-to-peer connection* sudah terbentuk untuk bertukar data. Selain bertukar informasi tentang media melalui SDP, kedua pengguna juga harus bertukar informasi tentang koneksi jaringan, dan ini dikenal dengan *ICE candidate*. Hal tersebut dilakukan melalui ICE yang menggunakan sebuah *third party server* yang berfungsi untuk mengambil *public IP Address*, dan server tersebut biasa dikenal dengan sebutan STUN dan TURN.

2.2.2. Kotlin

Kotlin adalah bahasa pemrograman modern berbasis *Java Virtual Machine* (JVM). Kotlin dikembangkan oleh JetBrains dan diadopsi oleh Google untuk pengembangan aplikasi Android menggunakan Android Studio. Kotlin menjadi salah satu bahas pemrograman kelas utama dan banyak digunakan oleh *developer*

karena cukup *powerfull* dan dapat digunakan untuk pengembangan aplikasi di berbagai platform dan sistem operasi. Keuntungan yang didapat dari penggunaan bahasa pemrograman Kotlin antara lain:

1. *Open Source*. Siapapun dapat menggunakan Kotlin secara gratis dan bahkan dapat memodifikasi dan berkontribusi
2. Bahasa Kotlin cukup sederhana dan ringkas sehingga mudah untuk dipelajari
3. Lebih ringan dibandingkan Java. Proses *debugging* atau *me-run* aplikasi pada emulator atau *real device* dapat dilakukan dengan cepat.
4. Bebas dan aman dari *NullPointerException* (NPE)
5. Terintegrasi dengan Android Studio
6. *Interoperable*. Kotlin dapat dikolaborasikan dengan Java. Tingkat *interoperability* dengan Java mencapai 100%. Apapun yang berbasis Java juga bisa menggunakan Kotlin.
7. Tidak hanya untuk Android. Kotlin dapat digunakan untuk pengembangan aplikasi server, iOS, dan web.

WebRTC dapat diimplementasikan di *Android Native* menggunakan bahasa pemrograman Kotlin. Implementasi dapat dilakukan dengan bantuan *library WebRTC* dengan *dependencies* yaitu `com.mesibo.api:webrtc:1.0.5`.

2.2.3. XML

Extensible Markup Language (XML) adalah markup language yang diciptakan oleh *World Wide Web Consortium* (W3C). XML digunakan untuk mendeskripsikan dan memanipulasi data secara terstruktur agar pertukaran dan penyimpanan data antarserver dapat dilakukan dengan lebih sederhana. XML dapat

digunakan pada berbagai sistem karena bersifat independent. XML terdiri dari tiga struktur pembentuk, yaitu:

1. Deklarasi. Menunjukkan versi XML yang digunakan
2. Atribut. Berisi keterangan objek seperti judul dan lainnya
3. Elemen. Mendeskripsikan objek dengan tag pembuka, isi, dan tag penutup

2.2.4. *Interactive Connectivity Establishment (ICE)*

Interactive Connectivity Establishment (ICE) adalah sebuah protokol yang berfungsi dalam pengecekan koneksi antar dua klien. ICE memungkinkan terbentuknya koneksi langsung tanpa server perantara. Mekanisme penentuan pasangan ditentukan dengan membuat nominasi semua rute yang memungkinkan untuk pertukaran data antar klien saat sesi sedang berlangsung (Azzam *et al*, 2019). Dua agen ICE berkomunikasi menggunakan paket ICE ping atau (pengecekan konektivitas) untuk membangun konektivitas. Setelah konektivitas terbentuk, maka pengiriman audio, video, ataupun data dapat dilakukan layaknya seperti menggunakan socket normal.

2.2.5. *Session Description Protocol (SDP)*

Session Description Protocol (SDP) merupakan sebuah bentuk standar informasi kebutuhan dalam pertukaran data meliputi nama sesi, tujuan sesi, waktu aktif sesi, informasi yang dibutuhkan media seperti *address*, *port*, format, dan lain sebagainya yang dibutuhkan untuk melakukan proses konten multimedia. Tujuan adanya SDP adalah agar sistem dapat digunakan dalam area cakupan yang luas dengan lingkungan jaringan yang berbeda (Azzam *et al*, 2019).

2.2.6. Signalling Server

Signalling Server bertugas untuk melayani pertukaran data antar klien. *Signalling Server* digunakan pada *WebRTC* untuk mengurus komunikasi setiap klien (Azzam *et al*, 2019). *Signalling Server* dilakukan dengan menggunakan API dari *websocket* yang mendukung koneksi dua arah. Mekanisme *signalling server* adalah klien mengirimkan pemberitahuan status terhubung ke server lalu server akan meneruskan ke klien lainnya. Permintaan notifikasi dilakukan melalui *websocket* atau protokol HTTP seperti REST untuk pengiriman dan *long polling* untuk penerimaan. *Websocket* harus dalam keadaan hidup dan tidak dapat digunakan sewaktu-waktu. *Long polling* perlu membuat HTTP request berkali-kali dalam waktu yang singkat yang menghabiskan banyak *bandwidth*.

2.2.7. Peer-to-peer (P2P)

Komunikasi *peer-to-peer* (P2P) adalah sebuah bentuk komunikasi dua arah yang dalam interaksinya terjadi pertukaran informasi. Bentuk informasi yang dapat dibagikan dengan *WebRTC* berupa video, audio, dan data teks. Interaksi P2P membutuhkan koneksi internet dan salinan perangkat lunak atau protokol untuk setiap perangkat yang terlibat.

2.2.8. Network Address Translation (NAT)

Network Address Translation (NAT) adalah proses pendistribusian alamat publik ke satu atau beberapa komputer jaringan di dalam jaringan pribadi yang dilakukan oleh perangkat jaringan seperti *firewall* dan *router* untuk diterjemahkan ke dalam alamat global untuk menyediakan akses ke *host* lokal. Sebelum paket didistribusikan, alamat IP pribadi diterjemahkan menjadi alamat yang legal dan

unik secara global. Jumlah alamat IP pribadi dibatasi dengan mengaktifkan dan menerjemahkannya ke alamat IP yang tidak terdaftar *online*. Cara kerja NAT adalah sebagai berikut:

1. NAT menerima permintaan dari klien yang ditujukan ke suatu *website*, *server remote*, atau jaringan lainnya di internet
2. Alamat IP klien dicatat dan disimpan ke dalam *address translation table*
3. Alamat IP pribadi diubah ke nomor IP publik
4. Paket dikirimkan ke tujuan awal dengan alamat IP publik yang telah berubah

2.2.9. Data Stream

Data streaming adalah proses mengumpulkan data yang dilakukan terus-menerus pada saat data dihasilkan dan memindahkan data tersebut ke tujuan. Umumnya data-data tersebut ditangani oleh perangkat lunak *stream processing* untuk dianalisis, disimpan, dan ditindaklanjuti. *Data streaming* yang dikombinasikan dengan *stream processing* dapat menghasilkan *real-time intelligence*.

2.2.10. Session Traversal Utilities for NAT (STUN)

Session Traversal Utilities for NAT (STUN) adalah protokol yang diciptakan untuk bekerja dengan NAT. Protokol STUN juga digunakan oleh ICE/TURN. STUN sangat berguna dalam pemetaan NAT secara terprogram untuk mengetahui IP dan port yang digunakan. STUN juga memberikan detail pemetaan NAT yang dapat dibagikan ke orang lain. Pemetaan yang telah dibuat dapat dimanfaatkan untuk pengiriman lalu lintas.

2.2.11. Traversal Using Relays around NAT (TURN)

Traversal Using Relays around NAT (TURN) merupakan sebuah protocol yang membantu traversal NAT atau firewall pada aplikasi WebRTC. TURN protokol adalah ekstensi untuk STUN. Dengan adanya TURN, proses pertukaran data dapat dilakukan oleh klien melalui server perantara.

2.2.12. Web Socket

Web socket merupakan sebuah teknologi yang berguna untuk membuat koneksi antara klien dengan server sehingga memungkinkan untuk dilakukannya komunikasi real-time. Web socket dapat mengurangi lalu lintas jaringan yang tidak penting dan juga latency dalam skala yang cukup besar.

2.2.13. Konsultasi Ahli

Dikutip dari KBBI konsultasi merupakan kegiatan bertukar pikiran yang menghasilkan sebuah kesimpulan (nasihat, saran, dan sebagainya) yang sebaik-baiknya. Konsultasi ahli dilakukan oleh setidaknya satu ahli dalam bidang tertentu dengan tujuan untuk bertukar pikiran atau mendapatkan nasihat dalam mencapai tujuan tertentu.