**PENGEMBANGAN LAYANAN KONFERENSI VIDEO JITSI**

**UNTUK PT. TELEKOMUNIKASI INDONESIA TBK**

**PT. PADEPOKAN TUJUH SEMBILAN**

**LAPORAN PKL**

Diajukan untuk memenuhi syarat kelulusan Mata Kuliah PKL

****

Disusun oleh:

**Sophia Gianina Daeli NIM 171524029**

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI DIPLOMA IV TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2020**

**PENGEMBANGAN LAYANAN KONFERENSI VIDEO JITSI**

**UNTUK PT. TELEKOMUNIKASI INDONESIA TBK**

**PT. PADEPOKAN TUJUH SEMBILAN**

Disusun oleh:

**Sophia Gianina Daeli NIM 171524029**

Laporan PKL ini telah diperiksa dan disahkan  
di Bandung, … Desember 2020.

|  |  |
| --- | --- |
| Pembimbing JTK Polban,  Joe Lian Min, M.Eng  NIP. 19661018 199512 1 001 | Pembimbing Industri,  Arif Syaripudin, S.Si., M.T  NIP. 010912001 |
| Ketua Program Studi D IV Teknik Informatika JTK POLBAN,  Santi Sundari, S.Si., M.T.  NIP 197109031999032001 | |

# DAFTAR ISI

[DAFTAR ISI i](#_Toc64571726)

[DAFTAR TABEL v](#_Toc64571727)

[DAFTAR GAMBAR viii](#_Toc64571728)

[DAFTAR LAMPIRAN ix](#_Toc64571729)

[BAB I PENDAHULUAN 1](#_Toc64571730)

[I.1. Latar Belakang 1](#_Toc64571731)

[I.2. Perumusan Masalah 2](#_Toc64571732)

[I.3. Ruang Lingkup 3](#_Toc64571733)

[I.4. Tujuan 3](#_Toc64571734)

[I.5. Tahapan 4](#_Toc64571735)

[I.6. Lokasi dan Waktu 4](#_Toc64571736)

[I.6.1. Lokasi 5](#_Toc64571737)

[I.6.2. Waktu 5](#_Toc64571738)

[I.7. Sistematika Penulisan 5](#_Toc64571739)

[BAB II PROFIL PERUSAHAAN 7](#_Toc64571740)

[II.1. Tinjauan Umum Perusahaan 7](#_Toc64571741)

[II.2. Struktur Organisasi 9](#_Toc64571742)

[II.3. Deskripsi Kerja 12](#_Toc64571743)

[BAB III PENGETAHUAN DAN PERANGKAT PENDUKUNG 13](#_Toc64571744)

[III.1. Konsep Dasar 13](#_Toc64571745)

[III.1.1. Jitsi 13](#_Toc64571746)

[III.1.2. Bandwidth 14](#_Toc64571747)

[III.1.3. Last N 15](#_Toc64571748)

[III.1.4. Pagination 15](#_Toc64571749)

[III.1.5. WebRTC 15](#_Toc64571750)

[III.1.6. RESTful API 16](#_Toc64571751)

[III.2. Bahasa Pemrograman dan Teknologi 17](#_Toc64571752)

[III.2.1. XMPP 17](#_Toc64571753)

[III.2.2. Java 17](#_Toc64571754)

[III.2.3. JavaScript 18](#_Toc64571755)

[III.2.4. Lua 18](#_Toc64571756)

[III.2.5. React 18](#_Toc64571757)

[III.2.6. AJAX 18](#_Toc64571758)

[III.2.7. Axios 19](#_Toc64571759)

[III.2.8. Secure Shell 19](#_Toc64571760)

[III.3. Perangkat Pendukung 19](#_Toc64571761)

[III.3.1. Node.js 20](#_Toc64571762)

[III.3.2. IntelliJ IDEA 20](#_Toc64571763)

[III.3.3. Visual Studio Code 20](#_Toc64571764)

[III.3.4. Windows Subsystem Linux 21](#_Toc64571765)

[III.3.5. Ubuntu 21](#_Toc64571766)

[III.3.6. GitLab 21](#_Toc64571767)

[III.3.7. AnyDesk 22](#_Toc64571768)

[III.3.8. GlobalProtect 22](#_Toc64571769)

[III.3.9. Google Chrome 23](#_Toc64571770)

[III.3.10. Google Docs 23](#_Toc64571771)

[III.3.11. WhatsApp 23](#_Toc64571772)

[III.3.12. Google Meet 24](#_Toc64571773)

[III.3.13. AirTabel 24](#_Toc64571774)

[BAB IV ANALISIS DAN EVALUASI 25](#_Toc64571775)

[IV.1. Analisis 25](#_Toc64571776)

[IV.1.1. Timeline Analisis 26](#_Toc64571777)

[IV.1.2. Teknik Analisis 28](#_Toc64571778)

[IV.1.3. Kendala Analisis 29](#_Toc64571779)

[IV.2. Hasil Analisis 29](#_Toc64571780)

[IV.2.1. Log Prosody 29](#_Toc64571781)

[IV.2.2. Last N 30](#_Toc64571782)

[IV.2.3. Alur Pengiriman Paket di Jitsi Videobridge (JVB) 31](#_Toc64571783)

[IV.2.4. Media Komunikasi Berbasis XMPP dengan Kostumisasi 34](#_Toc64571784)

[IV.2.5. Metode Pemanggilan API dari Jitsi Meet 36](#_Toc64571785)

[IV.2.6. Partisipan Octo 38](#_Toc64571786)

[IV.2.7. Alur Pengiriman Paket di Jitsi Conference Focus (Jicofo) 39](#_Toc64571787)

[IV.2.8. Penerapan *Pagination* di Jitsi Meet 45](#_Toc64571788)

[IV.3. Evaluasi Analisis 46](#_Toc64571789)

[BAB V PERANCANGAN 48](#_Toc64571790)

[V.1. Perancangan pada Jitsi Videobridge (JVB) 48](#_Toc64571791)

[V.1.1. Pemberhentian Pengiriman Paket ke Partisipan (Non-Octo) 48](#_Toc64571792)

[V.1.2. Fungsi RESTful API 52](#_Toc64571793)

[V.1.3. Pengambilan Data ID Konferensi 54](#_Toc64571794)

[V.2. Perancangan pada Jitsi Conference Focus (Jicofo) 55](#_Toc64571795)

[V.3. Perancangan pada Jitsi Meet 55](#_Toc64571796)

[BAB VI IMPLEMENTASI 60](#_Toc64571797)

[VI.1. Implementasi di Jitsi Videobridge (JVB) 60](#_Toc64571798)

[VI.1.1. Pemberhentian Pengiriman Paket ke Partisipan (Non-Octo) 60](#_Toc64571799)

[VI.1.2. Fungsi RESTful API 64](#_Toc64571800)

[VI.1.3. Pengambilan Data ID Konferensi 67](#_Toc64571801)

[VI.2. Implementasi di Jitsi Conference Focus (Jicofo) 68](#_Toc64571802)

[VI.3. Implementasi di Jitsi Meet 71](#_Toc64571803)

[BAB VII PENUTUP 78](#_Toc64571804)

[VII.1. Kesimpulan 78](#_Toc64571805)

[VII.2. Lesson Learned 79](#_Toc64571806)

[VII.3. Saran 81](#_Toc64571807)

[VII.3.1. PT. Padepokan Tujuh Sembilan 81](#_Toc64571808)

[VII.3.2. Jurusan Teknik Komputer dan Informatika Polban 82](#_Toc64571809)

[DAFTAR PUSTAKA 83](#_Toc64571810)

[LAMPIRAN 87](#_Toc64571811)

# DAFTAR TABEL

[Tabel IV.1. *Timeline* Proses Analisis 26](#_Toc60222172)

[Tabel IV.2. *Method* sendOut() 31](#_Toc60222173)

[Tabel IV.3. *Method* wants() 33](#_Toc60222174)

[Tabel IV.4. Fungsi sendIq() 34](#_Toc60222175)

[Tabel IV.5. Fungsi createNonAnonymousRoom() 35](#_Toc60222176)

[Tabel IV.6. *Method* createOffer() 40](#_Toc60222177)

[Tabel IV.7. *Method* muteParticipant() 41](#_Toc60222178)

[Tabel IV.8. Jenis MediaDirection 43](#_Toc60222179)

[Tabel IV.9. *Method* updateMediaDirection() 43](#_Toc60222180)

[Tabel IV.10. Potongan *Method* handleMuteIQ() 44](#_Toc60222181)

[Tabel IV.11. Potongan Kode *Class* MuteIq 44](#_Toc60222182)

[Tabel IV.12. *Method* claimSources() 45](#_Toc60222183)

[Tabel IV.13. Fungsi addRemoteParticipantContainer() 46](#_Toc60222184)

[Tabel V.1. Rancangan *Method* wantsPacketFromEndpoint() 50](#_Toc60222185)

[Tabel V.2. Rancangan *Method* addEndpointToAcceptedList() 50](#_Toc60222186)

[Tabel V.3. Rancangan *Method* removeEndpointFromAcceptedList() 50](#_Toc60222187)

[Tabel V.4. Rancangan *Method* clearWantsToRecieveFromEndPoints() 50](#_Toc60222188)

[Tabel V.5. Rancangan Fungsi removeWantedRemoteAudioStream() 52](#_Toc60222189)

[Tabel V.6. Rancangan Fungsi removeWantedRemoteVideoStream() 52](#_Toc60222190)

[Tabel V.7. Rancangan Fungsi unmuteAllRemoteAudioStream() 53](#_Toc60222191)

[Tabel V.8. Fungsi getConferenceID() 54](#_Toc60222192)

[Tabel V.9. Rancangan Fungsi getConferences() 55](#_Toc60222193)

[Tabel V.10. Rancangan Fungsi getBridgesParticipants() 55](#_Toc60222194)

[Tabel V.11. Rancangan Fungsi getRemoteEndpointPage() 56](#_Toc60222195)

[Tabel V.12. Rancangan Fungsi unmuteParticipant() 57](#_Toc60222196)

[Tabel V.13. Rancangan Fungsi muteParticipant() 58](#_Toc60222197)

[Tabel V.14. Rancangan Fungsi muteAllParticipant() 58](#_Toc60222198)

[Tabel V.15. Rancangan Fungsi addParticipantId() 59](#_Toc60222199)

[Tabel V.16. Rancangan Fungsi removeParticipantId() 59](#_Toc60222200)

[Tabel VI.1. Fungsi wantsPacketFromEndpoint() 61](#_Toc60222201)

[Tabel VI.2.Fungsi addEndpointToAcceptedList() 61](#_Toc60222202)

[Tabel VI.3. Fungsi removeEndpointToAcceptedList() 62](#_Toc60222203)

[Tabel VI.4. Fungsi clearWantsToRecieveFromEndPoints() 62](#_Toc60222204)

[Tabel VI.5. Fungsi sendOut() dengan Modifikasi 62](#_Toc60222205)

[Tabel VI.6. Fungsi removeWantedRemoteAudioStream() 64](#_Toc60222206)

[Tabel VI.7. Fungsi removeWantedRemoteVideoStream() 65](#_Toc60222207)

[Tabel VI.8. Fungsi getRemoteEndpointPerPage() 66](#_Toc60222208)

[Tabel VI.9. Fungsi getConferenceId() 67](#_Toc60222209)

[Tabel VI.10. Fungsi getConferences() 68](#_Toc60222210)

[Tabel VI.11. Fungsi getBridgesParticipants() 70](#_Toc60222211)

[Tabel VI.12. Modifikasi penambahan *button* di Jitsi Meet 71](#_Toc60222212)

[Tabel VI.13. Fungsi unmuteParticipant() 72](#_Toc60222213)

[Tabel VI.14. Fungsi muteParticipant() 73](#_Toc60222214)

[Tabel VI.15. Fungsi muteAllParticipant() 74](#_Toc60222215)

[Tabel VI.16. Fungsi addParticipantId() 75](#_Toc60222216)

[Tabel VI.17. Fungsi removeParticipantId() 75](#_Toc60222217)

[Tabel VI.18. Fungsi addRemoteParticipantContainer() dengan Modifikasi 76](#_Toc60222218)

[Tabel VI.19. Fungsi addRemoteParticipantContainer() dengan Modifikasi 76](#_Toc60222219)

[Tabel VI.20. Fungsi hideMutedRemoteVideo() 77](#_Toc60222220)

[Tabel VI.21. Fungsi showRemoteVideo() 77](#_Toc60222221)

# DAFTAR GAMBAR

[Gambar II.1. Logo PT. Padepokan Tujuh Sembilan 7](#_Toc60222306)

[Gambar II.2. Struktur Organisasi PT. Padepokan Tujuh Sembilan 9](#_Toc60222307)

[Gambar III.1 Arsitektur Jitsi 14](#_Toc60222308)

[Gambar IV.1. Potongan *Log* Prosody 30](#_Toc60222309)

[Gambar IV.2. Ilustrasi Partisipan Octo dan Non-Octo pada Jitsi 38](#_Toc60222310)

[Gambar V.1. Alur Pengiriman Paket di JVB Setelah Dimodifikasi 51](#_Toc60222311)

[Gambar V.2. Alur fungsi removeWantedRemoteAudioStream() 53](#_Toc60222312)

[Gambar V.3. Alur fungsi removeWantedRemoteVideoStream() 54](#_Toc60222313)

# DAFTAR LAMPIRAN

[Lampiran 1 88](#_Toc60132559)

# DESKRIPSI APLIKASI

## Permainan Monopoli secara Umum

Monopoli merupakan permainan genre papan yang berbasis strategi. Tujuan para pemain monopoli yaitu mengumpulkan uang sewa dari lawan mereka agar lawannya bangkrut. Setiap pemain melempar dadu secara bergantian untuk memindahkan bidaknya. Hasil lemparan dadu ini akan mengarahkan bidak pemain ke satu petak, dimana pemain akan membeli petak itu atau mebayar uang sewa. Pemain monopoli diberikan sejumlah uang untuk dapat melakukan transaksi jual beli dan membayar uang sewa. Selain itu, pemain dapat pula memperdagangkan properti. Adapun, papan dan perlengkapan lainnya yang dipakai pada permainan monopoli pada Gambar I.1.



Gambar I.1. Papan, kartu, uang, dan bidak pada permainan Monopoli.

## Skenario

Permainan monopoli memiliki banyak versi yang tergantung pada *storyboard* dan aturan-aturan yang berlaku. Maka dari itu, program kami menerapkan *storyboard* beserta peraturannya sebagai berikut.

### Menu Utama Permainan

#### Mulai Permainan

Mulai permainanmerupakan opsi untuk memulai permainan. Untuk mengakses opsi ini pemain perlu menekan angka satu dan menekan enter pada *keyboard*. Setelah itu, pemain akan diminta untuk memasuki nama permain yang dimana tidak boleh lebih dari lima karakter. Sesudah itu, permainan akan menampilkan tiga level pemain komputer dan pemain dapat memilih salah satu dari tiga pilihan tersebut dengan memasukkan nomor satu sampai tiga.

#### Skor Tertinggi

Skor Tertinggi merupakan opsi untuk menampikan riwayat skor paling tinggi. Untuk mengakses opsi ini pemain perlu menekan angka dua dan menekan enter pada *keyboard.* Pada opsi ini akan ditampilkan satu skor paling tinggi yang diraih oleh pemain dari seluruh permainan yang pernah dijalankan beserta waktu yang berupa tanggal, bula, tahun, jam, menit, dan detik kapan permainan dimulai. Pada tampilan Skor Tertinggi, skor berupa sisa jumlah uang dan harga keseluruhan harta pada akhir permainan.

#### Riwayat Skor

Riwayat Skor merupakan opsi untuk menampikan seluruh riwayat skor. Untuk mengakses opsi ini pemain perlu menekan angka tiga dan menekan enter pada *keyboard*. Riwayat skor akan ditampilkan berdasarkan tanggal paling awal ke tanggal paling terbaru disertai waktu yang berupa tanggal, bula, tahun, jam, menit, dan detik kapan permainan dimulai.

#### Bantuan

Bantuan merupakan opsi untuk menampilkan tujuan permainan dan cara bermain dari permainan tersebut. Untuk mengakses opsi ini pemain perlu menekan angka empat dan menekan enter pada *keyboard.* Opsi ini berfungsi untuk membantu pemain dalam memahami aturan dan skenario versi game kami.

#### Kredit

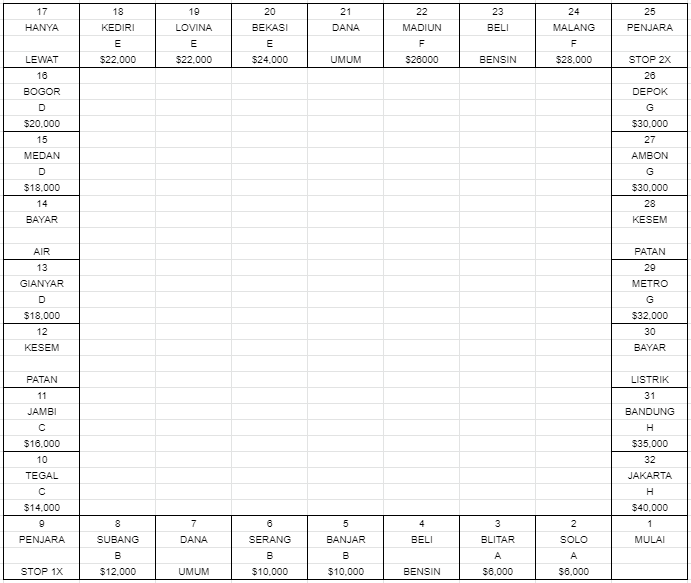
Kredit merupakan opsi untuk menampilkan data pembuat. Untuk mengakses opsi ini pemain perlu menekan angka lima dan menekan enter pada *keyboard*. Tampilan kredit berisi nama permainan dan data pembuat permainan yaiut nama, NIM, jurusan, produ, perguruan tinggi, dan tahun permainan dibuat.

#### Keluar

Keluar merupakan opsi untuk berhenti jalannya program. Untuk mengakses opsi ini pemain perlu menekan angka enam dan menekan enter pada *keyboard.*

### Permainan Monopoli

#### Papan

Monopoli versi kami memiliki 32 petak yang terdiri dari 1 petak mulai, 20 petak properti, 4 petak tagihan uang, 4 petak *fortune,* dan 2 petak hukuman. Sketsa papan monopoli kami dapat dilihat pada Gambar I.2.

Gambar I.2. Gambar sketsa papan monopoli

#### Pemain

Untuk tiap permainan, monopoli versi kami hanya dimainkan oleh dua pemain. Adapun, tipe pemain yaitu satu pemain merupakan pemain bertipe *human player* dan satu pemain bertipe *computer player*. Pada tipe pemain komputer terdapat 3 tingkat kecerdasan, yaitu:

1. First Timer (tingkat kecerdasan mudah)
2. Entrepreneur (tingkat kecerdasan sedang)
3. Tycoon (tingkat kecerdasan sulit)

#### Awal Permainan

Setiap awal permaian, tiap pemain bermain dengan uang sebanyak $150,000. Permainan dimulai dari pemain *human player* dan semua bidak pemain berada pada petak “Mulai” di papan.

#### Uang Pemain

Pada tiap permainan, pemain dapat melihat jumlah uang yang dimiliki semua pemain pada layar.

#### Perpindahan Bidak

Bidak pemain akan bergerak sebanyak angka hasil kocokan dadu pemain tersebut. Tiap pemain dapat memindahkan bidak pada papan seara bergiliran.

### Aksi Pemain

#### Membeli Properti

Ketika bidak pemain berhenti di petak properti yang belum dimiliki pemain lainnya, maka pemain pemain akan diberi pilihan untuk membeli tanah sekaligus membangun rumah seharga yang tertera di kartu hak milik atau menolak untuk membeli properti tersebut.

#### Membayar Uang Sewa

Ketika bidak pemain berhenti di petak properti yang sudah dimiliki, maka pemilik properti akan menerima uang sewa seharga yang tertera pada kartu hak milik.

#### Kesempatan dan Dana Umum

Ketika bidak berhenti di petak kesempatan atau dana umum, pemain akan mendapatkan kartu secara acak dan mendapatkan tindakan sesuai apa yang tertulis pada kartu.

#### Membayar Uang Tagihan

Ketika bidak berhenti di petak tagihan uang, pemain harus membayar sejumlah uang sesuai dengan apa yang tertulis pada kartu.

#### Masuk Penjara

Ketika bidak berhenti di petak hukuman, maka pemain akan distop giliran selanjutnya sebanyak yang tertera di papan. Pada versi monopoli kami terdapat 2 aksi petak hukuman yaitu distop di-*stop­* satu kali dan di­-*stop* dua kali.

#### Hanya Lewat

Ketika bidak berhenti di petak hanya lewat, maka pemain hanya akan berhenti pada petak tersebut dan tidak melakukan tindakan apa-apa sampai gilirannya yang selanjutnya.

#### Menginjak Petak Mulai

Ketika bidak berhenti di petak “Mulau” setelah mengelilingi papan permainan, maka pemain akan mendapatkan uang sejumlah $20,000.

#### Membangun Rumah

Ketika bidak berhenti di petak properti miliknya sendiri, pemain dapat membangun rumah maksimal empat rumah. Pemain dapat membangun rumah berkali-kali hingga empat rumah tiap mendapat kesempatan membangun rumah.

#### Kekurangan Uang

Ketika jumlah uang pemain tidak cukup untuk membayar tagihan uang atau membayar uang sewa maka pemain akan diberi pilihan untuk menjual propertinya. Pada permainan monopoli versi kami, menjual properti akan diberi keuntungan sebanyak setengah harga tanah. Jika, properti yang akan dijual sudah dibangun rumah, pemain tidak dapat menjual rumah dan tanahnya secara bertahap, tetapi pemain hanya menjual tanah sekaligus rumah yang dibangun dalam sekali menjual.

## Aturan Permainan

### Awal Permainan

Setiap awal bermain, pemain akan diberlakukan hal-hal berikut, yaitu:

1. Tiap pemain mulai dengan uang sebanyak $150,000
2. Tiap pemain belum memiliki properti apapun
3. Tiap pemain memulai permainan di petak “Mulai”

## Inisialisasi Papan

### Petak Properti

Petak Properti memiliki harga jual, harga sewa tanpa rumah, harga sewa sesuai dengan banyak rumah yang telah dibangun di properti tersebut, dan harga membeli satu rumah untuk dibangun.

## Ruang Lingkup

Ruang lingkup pekerjaan yang diberikan penulis, sebagai *back end developer*, pada pengerjaan proyek *Jitsi* *Bandwidth Reduction* di PT. Padepokan Tujuh Sembilan adalah sebagai berikut.

1. Analisis metode untuk mengurangi penggunaan *bandwidth* pada konfe-rensi video dengan menggunakan aplikasi Jitsi.
2. Implementasi modifikasi dan penambahan fungsi.
3. Pengujian dan *bug-fixing* hasil impementasi modifikasi dan penambahan fungsi.

## Tujuan

Tujuan dari pelaksanaan Praktik Kerja Lapangan (PKL) adalah sebagai berikut.

1. Mendapatkan pengalaman bekerja di dunia industri untuk mempersiapkan karir di masa mendatang.
2. Mempelajari budaya kerja, seluk-beluk dalam pengerjaan proyek perusa-haan, dan berbagai hal lainnya yang menambah wawasan dalam bekerja di suatu perusahaan di bidang TI.
3. Memperoleh *life lesson* dan meningkatkan *soft skill* untuk menunjang di dunia kerja, seperti cara berkomunikasi dengan rekan tim dan atasan, mempelajari situasi untuk membuat suatu keputusan, cara untuk menyata-kan dan memaparkan pendapat maupun kritik, dan berbagai pembelajaran lainnya.
4. Mempelajari dan menambah wawasan teknologi yang digunakan dalam penugasan selama PKL, termasuk metode pendekatan untuk mengurangi penggunaan *bandwidth* pada konferensi daring, berbagai perangkat untuk melakukan *refactoring*, perangkat untuk otomatisasi *refactoring*, aplikasi untuk *project management*, *version control*, dan berbagai teknologi dan perangkat pendukung lainnya.

## Tahapan

Tahapan pengerjaan proyek *Jitsi Bandwidth Reduction* yang dilakukan selama pelaksanaan PKL adalah sebagai berikut.

1. Mempelajari hal-hal seputar Jitsi, termasuk komponen yang menyusun arsitektur Jitsi dan metode komunikasi berbasis XMPP untuk komunikasi antar komponen pada arsitektur Jitsi.
2. Persiapan *environment* dan instalasi perangkat yang dibutuhkan dalam pengerjaan proyek.
3. Analisis metode untuk mengurangi penggunaan *bandwidth* pada konfe-rensi video dengan menggunakan aplikasi Jitsi. Analisis dilakukan dengan menelusuri kode program (komponen) yang menyusun Jitsi, melakukan pencarian di internet, dan diselingi dengan simulasi konferensi video
4. Implementasi modifikasi dan penambahan fungsi sesuai hasila analisis yang telah dilakukan sebelumnya.
5. Pengujian hasil implementasi modifikasi dan penambahan fungsi untuk memastikan hasil implementasi bekerja sesuai dengan harapan. Pengujian dilakukan dengan melakukan simulasi konferensi video menggunakan komponen Jitsi yang telah diubah seusai kebutuhan.
6. Jika terdapat eror, atau modifikasi/fungsi yang ditambahkan masih belum sesuai dengan kebutuhan, dilakukan *bug-fixing* untuk memperbaiki hasil implementasi.

Proyek ini dikembangkan dengan menggunakan metode Agile sehingga tahap pengerjaan ketiga – keenam dilakukan beberapa kali sepanjang pengerjaan proyek sampai didapatkan sebauh metode untuk mengurangi penggunaan *bandwidth* pada konferensi video dengan menggunakan aplikasi Jitsi.

## Lokasi dan Waktu

PKL dilaksanakan di PT. Padepokan Tujuh Sembilan secara *remote* (*work from home*) karena pandemi global COVID-19. Namun, ini tidak menutup kemungkinan bagi peserta PKL untuk bekerja di kantor (*work form office*), terutama jika terdapat kendala yang tidak dapat diselesaikan secara *remote*.

### Lokasi

Pada pelaksanaan *work from home*, komunikasi di dalam tim dilakukan melalui media komunikasi WhatsApp. Selain itu, media yang digunakan adalah aplikasi konferensi video Google Meet dan Google Hangout untuk komunikasi secara *real-time*. Adapun, *work from office* dilaksanakan di kantor PT. Padepokan Tujuh Sembilan yang berlokasi di Kompleks Terasana No. 6A, Jalan Cihempelas (Bawah), Bandung 40171. *Work from office* dilakukan sekali sepanjang PKL, yaitu pada tanggal 10 September 2020.

### Waktu

PKL berlangsung selama 17 minggu, dari tanggal 10 Agustus 2020 sampai dengan 6 Desember 2020. Hari kerja dari Senin sampai dengan Jumat, dari pukul 08:00 – 17:00 WIB.

## Sistematika Penulisan

Sistematika penulisan laporan PKL program studi Diploma IV Teknik Informatika, Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung tahun 2020 adalah sebagai berikut:

|  |  |
| --- | --- |
| **BAB I** | **Pendahuluan**, berisi gambaran pelaksanaan Praktik Kerja Lapangan (PKL). Bab ini tersusun dari tujuh buah subbab, yaitu: latar belakang, perumusan masalah, ruang lingkup, tujuan, tahapan, lokasi dan waktu, dan sistematika penulisan. |
| **BAB II** | **Profil Perusahaan**, berisi tinjauan umum perusahaan yang menjadi tempat pelaksanaan Praktik Kerja Lapangan (PKL). Bab ini tersusun dari tiga subbab, yaitu: tinjauan umum perusahaan, struktur organisasi, dan deskripsi kerja. |
| **BAB III** | **Pengetahuan Dan Perangkat Pendukung**, berisi penjelasan konsep, bahasa pemrograman, teknologi, dan perangkat pendukung yang digu-nakan dalam pengerjaan proyek. |
| **BAB IV** | **Analisis dan Evaluasi**, berisi penjabaran proses dan hasil analisis yang dilakukan selama pengerjaan proyek. Di samping itu, dijelaskan pula evaluasi hasil analisis. |
| **BAB V** | **Perancangan**, berisi penjabaran proses dan hasil perancangan selama pengerjaan proyek. Tidak terdapat perancangan formal pada proyek ini sehingga bab ini menjelaskan kegiatan yang dilakukan dalam proses perancangan informal. |
| **BAB VI** | **Implementasi**, berisi penjabaran hasil implementasi selama pengerjaan proyek. |
| **BAB VII** | **Penutup**, berisi uraian pembelajaran dan kesimpulan pelaksanaan PKL, serta saran untuk PT. Padepokan Tujuh Sembilan dan Jurusan Teknik Informatika Polban terkait pelaksanaan PKL. |

# PROFIL PERUSAHAAN

Bab ini berisi tinjauan umum perusahaan yang menjadi tempat pelaksanaan Praktik Kerja Lapangan (PKL), yaitu PT. Padepokan Tujuh Sembilan. Informasi mengenai perusahaan didapatkan dari *website* resmi perusahaan ([tujuhsembilan.com](http://www.tujuhsembilan.com)), laman LinkedIn resmi perusahaan, dan beberapa karyawan perusahaan.

## Tinjauan Umum Perusahaan

PT. Padepokan Tujuh Sembilan adalah perusahaan konsultan Teknologi Informasi (TI) yang menyediakan berbagai layanan di bidang TI. Sejak didirikan pada tahun 2010, perusahaan ini telah berkembang menjadi perusahaan yang dipercaya untuk menangani banyak proyek dari klien nasional dan multinasional. Hingga laporan ini ditulis, PT. Padepokan Tujuh Sembilan telah mengerjakan 22 proyek untuk 30 klien, dengan 104 sumber daya manusia (talenta) yang terampil untuk mendukung proyek TI klien. Selain itu, PT. Padepokan Tujuh Sembilan memiliki sebuah visi untuk menjadi perusahaan penyedia talenta di bidang pengembangan TI terbesar di Indonesia pada tahun 2025.



Gambar II.1. Logo PT. Padepokan Tujuh Sembilan

Sejarah singkat tentang PT. Padepokan Tujuh Sembilan, perusahaan ini sebelumnya merupakan perusahaan persekutuan komanditer dengan nama CV. Tujuh Sembilan. Perusahaan beralih bentuk badan usaha menjadi perseroan terbatas (PT) pada awal tahun 2020 untuk memperluas jangkauan perusahaan dalam melakukan proyek dan berinvestasi di bidang pengembangan TI.

Sebagai perusahaan yang menyediakan layanan, PT. Padepokan Tujuh Sembilan memiliki komitmen untuk memberikan layanan berkualitas tinggi, andal, efisien, dan mudah diaplikasikan/digunakan oleh klien. Untuk mencapai tujuan tersebut, perusahaan berfokus pada peningkatan kemampuan talenta – dalam segi *hard-skill* dan *soft-skill* – agar didapatkan talenta-talenta yang cakap dan unggul di bidang pengembangan TI. Hal ini tercermin pada slogan perusahaan, *“Always improving you”*, yang berarti perusahaan ini selalu meningkatkan (kemampuan) talenta; dan filosofi di balik nama perusahaan. Adapun nama perusahaan ini – Padepokan Tujuh Sembilan – secara harfiah memiliki arti “tempat belajar untuk mengembangkan kemampuan talenta dari apa adanya menjadi berdaya guna dan kompeten”, dari kata Padepokan yang berarti tempat belajar, Tujuh yang melambangkan apa adanya, dan Sembilan berarti berdaya guna dan kompeten. Selain itu, perusahaan menjunjung kejujuran dalam bekerja dengan menolak segala bentuk korupsi, dan berkompromi untuk mencapai titik tengah yang disetujui oleh semua pihak terkait dalam suatu masalah.

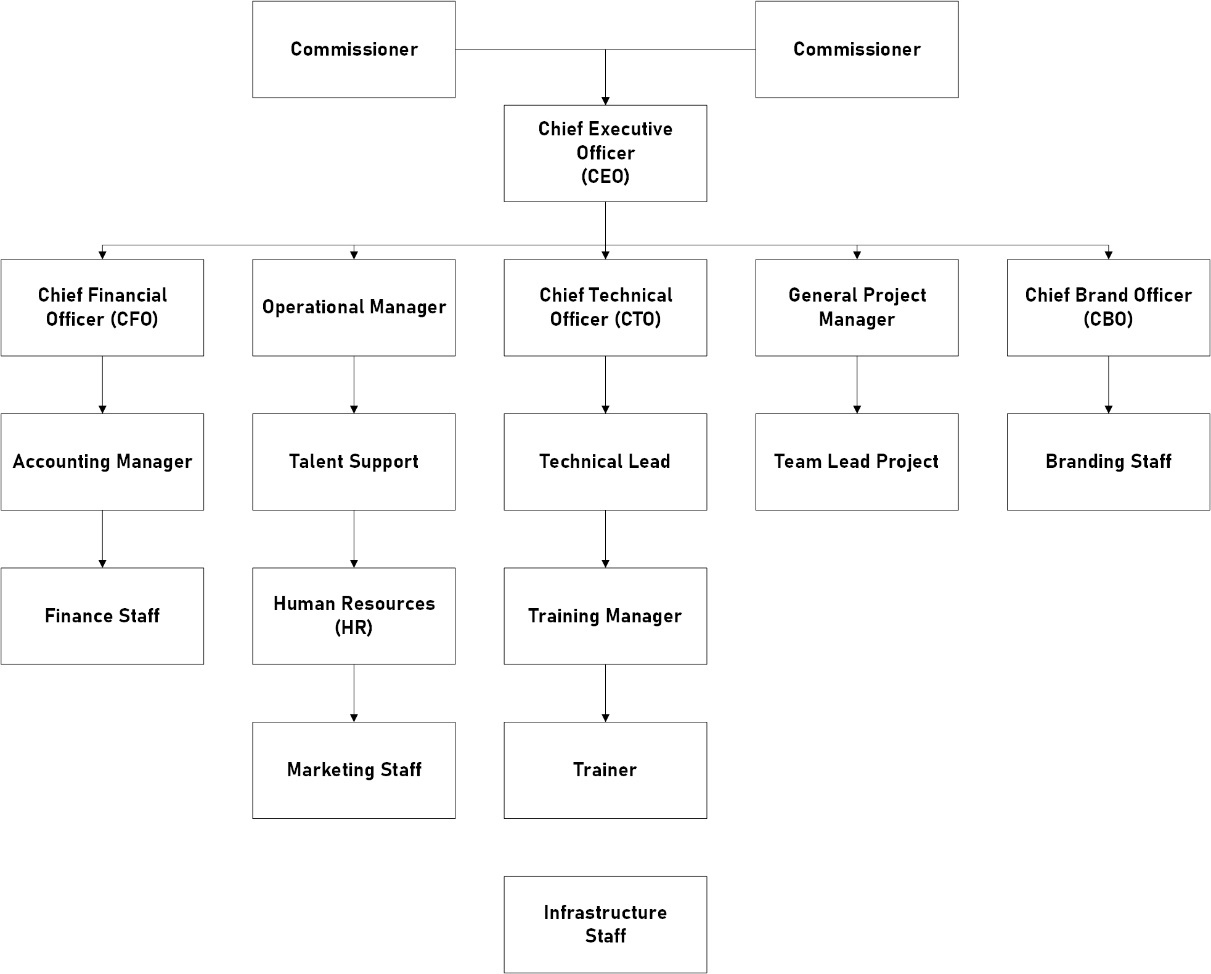
PT. Padepokan Tujuh Sembilan menyediakan beberapa layanan pengembangan teknologi informasi, yaitu:

1. *onsite placement*, penyediaan talenta berkualitas yang memenuhi syarat dan kompatibel untuk mendukung proyek TI klien;
2. *software project*, layanan proyek TI yang sesuai kebutuhan dan permintaan klien;
3. *hiring*, layanan perekrutan talenta TI yang kompatibel dengan klien;
4. *bootcamp*, program pelatihan kemampuan *hard* skill dan *hard skill* talenta selama tiga bulan untuk bersaing di industri TI;
5. *training*, program pelatihan TI untuk meningkatkan keterampilan spesifik talenta dengan durasi waktu yang lebih sedikit dibanding *bootcamp*.

Per Desember 2020, PT. Padepokan Tujuh Sembilan memiliki sebuah kantor yang terletak di Kompleks Terasana No. 6A, Jl. Cihampelas (Bawah), Bandung 40171. Perusahaan dapat dihubungi dengan nomor telepon (022) 20505455, dan *platform* media sosial LinkedIn, Facebook, dan Instagram dengan nama Padepokan Tujuh Sembilan.

## Struktur Organisasi

PT. Padepokan Tujuh Sembilan dibangun oleh komponen-komponen yang saling bahu membahu untuk membangun perusahaan yang kohesif. Adapun, organisasi perusahaan secara struktural ditunjukkan pada Gambar II.2.



Gambar II.2. Struktur Organisasi PT. Padepokan Tujuh Sembilan

Adapun, tugas dan tanggung jawab dari setiap komponen organisasi PT. Padepokan Tujuh Sembilan adalah sebagai berikut.

1. *Commissioner* (komisaris) mengawasi kinerja perusahaan secara berkala dan mengevaluasi hasil perusahaan. Selain itu, komisaris berhak menunjuk direktur perusahaan, mempertimbangkan rencana yang akan diajukan oleh pimpinan perusahaan, dan memberikan masukan bagi perusahaan.
2. *Chief Executive Officer* (CEO) memimpin, mengawasi, memantau, dan mengendalikan divisi-divisi yang berada di bawahnya.
3. *Chief Financial Officer* (CFO) merancang penggalangan dana, mengelola arus kas, merumuskan dokumen keuangan, dan ikut serta dalam proses pengambilan keputusan dalam perekrutan dan pengembangan SDM.
4. *Operational Manager* mengembangkan kebijakan dan prosedur operasional perusahaan, memantau kepatuhan karyawan terhadap hukum dan regulasi, mempromosikan budaya perusahaan, membangun aliansi dan kemitraan dengan organisasi/perusahaan lain, mengawasi penganggaran, pelaporan, perencanaan, dan audit; dan membantu departemen HR dalam perekrutan karyawan (jika diperlukan).
5. *Chief Technical Officer* (CTO) memantau proses pengembangan produk, menentukan dan implementasi teknologi yang cocok dengan produk yang dikembangkan, dan mengelola pengembangan produk.
6. *General Project Manager* berkoordinasi dengan manajer proyek dan *team leader*, memfasilitasi tim untuk mendapatkan solusi untuk isu yang dialami selama pengerjaan proyek, dan estimasi nilai proyek berdasarkan *scope* pekerjaan.
7. *Chief Brand Officer* (CBO) mengawasi, memantau, mengendalikan, dan koordinasi dalam pengelolaan *brand* perusahaan.
8. *Accounting Manager* mencatat jurnal harian, rekonsiliasi kas, menyusun laporan keuangan, mengawasi anggaran keuangan, menindaklanjuti status faktur dan tagihan klien, memperbarui *bank statement*, estimasi *cash flow*, dan proyeksi pendapatan dan keuntungan perusahaan.
9. *Finance Staff* melakukan pembayaran yang berhubungan dengan kas kecil, membuat dan mengirimkan faktur, dan mengurus pembayaran uang saku *bootcamp*.
10. *Technical Lead* berperan sebagai perantara antara tim pengembang tim produk, menganalisis spesifikasi dari tim produk, menjelaskan teknis tugas kepada tim pengembang (*frontend*, *backend*, dan *quality assurance*) untuk direalisasikan pada proyek.
11. *Training Manager* menggali potensi talenta, meningkatkan kemampuan talenta, dan menyusun materi pelatihan sesuai kebutuhan perusahaan.
12. *Trainer* menyiapkan bahan ajar dan memberikan pelatihan sesuai silabus, memberikan penilaian kepada peserta pelatihan, dan memberikan laporan evaluasi pelatihan kepada *training manager*.
13. *Talent Support* memfasilitasi talenta yang melakukan wawancara dengan klien, melakukan kunjungan untuk memantau kondisi dan kebutuhan talenta yang bekerja *onsite*, memantau perkembangan talenta, dan evaluasi kinerja talenta.
14. *Human Resources* (HR) mendata absensi, memantau dan evaluasi kinerja karyawan, memberi kompensasi dan proteksi, memberi layanan konsultasi dan konseling, mengelola pinjaman karyawan, memasok organisasi klien dengan tenaga kerja yang komptibel, dan memeriksa kelengkapan *talent bootcamp*.
15. *Marketing Staff* mencari klien baru, menerima kebutuhan layanan/produk dari klien, komunikasi dengan klien terkait kebutuhan/spesifikasi talenta yang dibutuhkan, dan meneruskan kebutuhan layanan/produk ke divisi yang terkait.
16. *Team Lead Project* menetapkan tujuan tim, memantau kinerja tim, membuat laporan harian, mendelegasikan tugas pada anggota tim dan menetapkan tenggat waktu pengerjaan tugas tersebut.
17. *Branding Staff* membuat konten harian dan materi pemasaran, memantau aktivitas media sosial milik perusahaan, dan memenuhi keperluan aktivitas *branding* perusahaan.
18. *Infrastructure Staff* menginisiasi pengerjaan proyek, instalasi *environment* pengembangan dan produksi aplikasi, *deploy* aplikasi, menangani kendala listrik dan jaringan/internet, memelihara jaringan/internet di perusahaan, dan instalasi sistem operasi.

## Deskripsi Kerja

Pada pelaksanaan PKL di PT. Padepokan Tujuh Sembilan, penulis dilibatkan dalam pengerjaan proyek *Jitsi Bandwidth Reduction* sebagai *back end developer*. Adapun, cakupan tugas penulis, yaitu: analisis metode untuk mengurangi penggunaan *band-width* pada konferensi video dengan menggunakan aplikasi Jitsi, implementasi modifikasi dan penambahan fungsi, dan menguji serta *bug-fixing* hasil impementasi modifikasi dan penambahan fungsi. Pada pengerjaan proyek ini, penulis berada di bawah bimbingan Arif Syaripudin, selaku pembimbing industri dari PT. Padepokan Tujuh Sembilan, dan Ikhwan Fadhillah, selaku manajer proyek *Jitsi Bandwidth Reduction*.

# PENGETAHUAN DAN PERANGKAT PENDUKUNG

Bab ini berisi penjelasan mengenai konsep, bahasa pemrograman, teknologi, dan beberapa perangkat pendukung yang digunakan penulis dalam pengerjaan proyek *Jitsi Bandwidth Reduction*.

## Konsep Dasar

Subbab ini berisi penjelasan beberapa konsep dasar yang dibutuhkan untuk menun-jang pengerjaan proyek *Jitsi Bandwidth Reduction*.

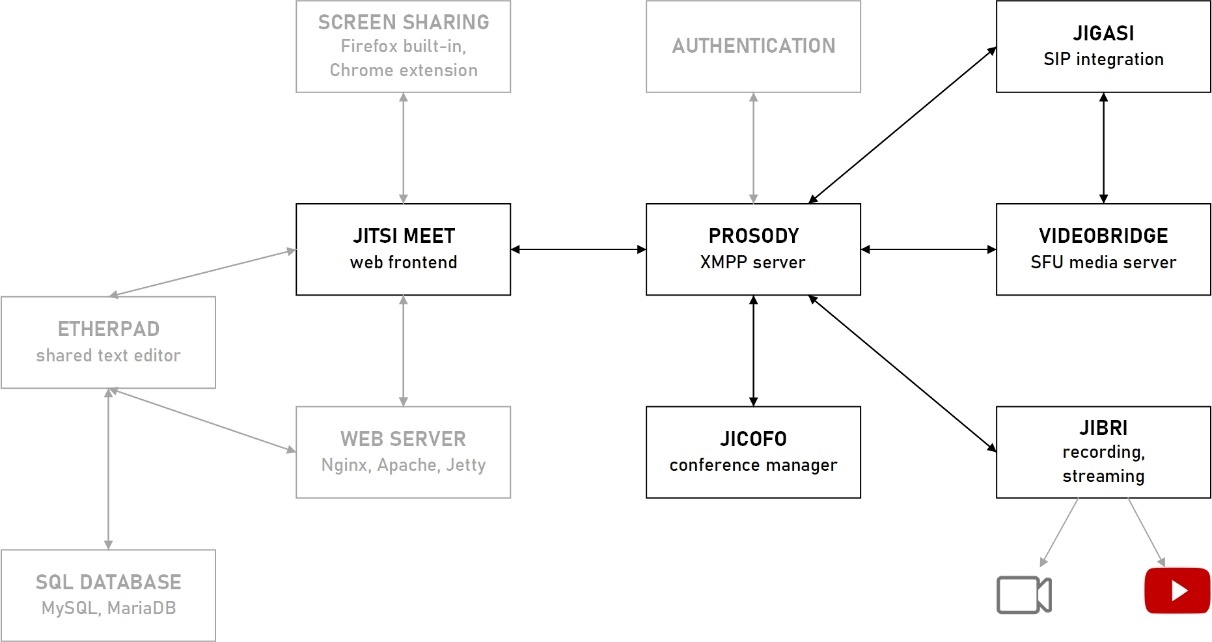
### Jitsi

Jitsi adalah sebuah aplikasi konferensi video yang dirintis oleh Emil Ivov pada tahun 2003. Jitsi dibangun oleh sekumpulan program (komponen) *open-source* yang dapat dipergunakan secara bebas untuk membangun aplikasi konferensi video dengan modifikasi (*What is Jitsi? - open source video conferencing API’s, SDKs, and installers*, n.d.). Jitsi menyediakan berbagai fitur, di antaranya: tampilan video dan audio pengguna konferensi, *dial-in*, perekaman, dan *simulcasting*.

Komponen yang membangun layanan konferensi video Jitsi adalah sebagai berikut (*Introduction · Jitsi Meet Handbook*, n.d.).

1. Jitsi Meet adalah aplikasi JavaScript yang kompatibel dengan WebRTC berfungsi sebagai *web frontend* Jitsi untuk menyajikan *user interface* (UI) konferensi video.
2. Jitsi VideoBridge (JVB) adalah server yang kompatibel dengan WebRTC, dirancang untuk mangatur aliran paket audio dan video semua partisipan di sebuah konferensi video.
3. Jitsi Conference Focus (Jicofo) adalah komponen *focus* di sisi server Jitsi untuk mengelola sesi media antara setiap peserta di sebuah konferensi video dengan JVB.
4. Jitsi Gateway to SIP (Jigasi) adalah aplikasi di sisi server yang berperan sebagai gerbang bagi klien SIP reguler/biasa untuk bergabung ke dalam konferensi video di Jitsi.
5. Jibri adalah rangkaian perangkat pendukung pada Jitsi untuk merekam dan melakukan *streaming* konferensi video.
6. Prosody adalah program eksternal di luar Jitsi, sebuah server komunikasi XMPP untuk komunikasi antar komponen di dalam Jitsi.

Bagan arsitektur layanan Jitsi ditunjukkan pada Gambar III.1.



Gambar III.1 Arsitektur Jitsi

### Bandwidth

*Bandwidth* adalah kapasitas maksimal jalur komunikasi untuk melakukan proses pengiriman dan penerimaan data dalam hitungan detik (*Pengertian Bandwidth dan Fungsinya (Lengkap) - Niagahsoter*, n.d.). *Bandwidth* jalur komunikasi (jaringan) menentukan kecepatan pengiriman dan penerimaan data. Apabila bandwidth tidak memadai, lalu lintas data akan tersendat dan menghambat aktivitas pengguna di jaringan. Dalam konteks digital, *bandwidth* diukur dalam satuan bit per sekon (bps) atau bytes per sekon.

### Last N

*Last N* adalah skema yang dikembangkan untuk meningkatkan performa konferensi video pada aplikasi Jitsi dengan membatasi jumlah data video partisipan *remote* yang diterima partisipan lokal (sebanyak N buah). Untuk mencapai tujuan tersebut, dibuat modifikasi pada aliran data agar data mengalir secara dinamis sesuai dengan aktivitas audio partisipan. Dengan kata lain, partisipan lokal hanya dapat menerima data video dari partisipan *remote* dengan audio aktif (partisipan *speaker*). Namun, partisipan lokal tetap dapat memilih untuk menerima data video dari partisipan *remote* dengan aktivitas audio nihil. Skema *Last N* hanya dapat digunakan untuk membatasi aliran video dan bukan untuk audio (Grozev et al., 2015).

Skema *Last N* termasuk ke dalam metode pendekatan yang dipertimbangkan tim proyek untuk mengurangi penggunaan *bandwidth* pada konferensi video, namun skema ini tidak sesuai dengan kebutuhan klien sehingga tidak digunakan.

### Pagination

*Pagination* adalah suatu metode pembagian konten situs *website* menjadi beberapa halaman terpisah. Pada proyek ini, *pagination* diterapkan untuk membatasi jumlah data audio dan video partisipan *remote* yang diterima partisipan lokal dalam kurun waktu tertentu. Implementasi *pagination* tidak hanya dikakukan pada *web frontend* Jitsi Meet – dengan menyembunyikan tampilan video partisipan *remote*, namun juga pada *backend* aplikasi Jitsi dengan menghentikan pengiriman data sehingga penggunaan *bandwidth* akan lebih kecil dibandingkan dengan sebelum penerapan *pagination*.

### WebRTC

WebRTC adalah teknologi yang dikembangkan oleh tim Google WebRTC untuk menambahkan fitur komunikasi *real-time* pada aplikasi. Teknologi ini mendukung pengiriman data video, suara, dan generik antar pengguna; digunakan pada aplikasi konferensi video. Teknologi ini diimplementasikan sebagai *open web standard* dan tersedia sebagai API JavaScript di berbagai browser. Untuk aplikasi *native*, misal aplikasi Android atau iOS, tersedia *library* untuk fungsionalitas yang sama. Proyek WebRTC bersifat *open-source* dan didukung oleh Apple, Google, Microsoft, dan Mozilla (*WebRTC*, n.d.).

### RESTful API

*REpresentational State Transfer* (REST) adalah model arsitektur untuk sistem hypermedia terdistribusi yang dicetuskan oleh Roy Fielding di dalam disertasinya pada tahun 2000 (*What is REST – REST API Tutorial,* n.d.).

*Application Programming Interface* (API) adalah kumpulan protokol yang menjadi kontrak dalam komunikasi antara penyedia informasi dan pengguna (Framework & Tips, 2010). Suatu API dapat disebut RESTful API apabila memenuhi enam kriteria berikut

1. *Client – server*, klien dan penyimpanan data (server) diimplementasi secara terpisah, meningkatkan portabilitas klien di berbagai *platform* dan skalabilitas dengan menyederhanakan komponen server.
2. *Stateless*, *request* dari klien harus mengandung semua informasi yang penting untuk dapat dipahami server dan tidak dapat menggunakan konteks yang disimpan di server.
3. *Cacheable*, data respon terhadap sebuah *request* diberi label *cacheable* atau *non-cacheable*. Jika *cacheable*, data respons dapat digunakan kembali untuk *request* yang serupa.
4. *Uniform interface,* *interface* seragam sehingga informasi yang dikirimkan dalam bentuk/format yang standar.
5. *Layered system,* bentuk sistem berlapis yang tersusun secara hierarkis dengan setiap lapisan tidak mengetahui lapisan lain, kecuali lapisan yang berinteraksi secara langsung satu sama lain, untuk enkapsulasi informasi hanya di satu lapisan tertentu.
6. *Code on demand* (opsional), meningkatkan fungsionalitas klien dengan mengunduh dan menjalankan kode dalam bentuk applet atau skrip.

Pada arsitektur REST, informasi di server (disebut *resource*) diidentifikasi oleh *Universal Resource Identifier* (URI). *Resource* dikirimkan kepada klien dalam bentuk representasi *JavaScript Object Notation* (JSON), HTML, XML, XLT, atau teks melalui *Hypertext Transfer Protocol* (HTTP).

## Bahasa Pemrograman dan Teknologi

Subbab ini berisi penjelasan ringkas mengenai bahasa pemrograman dan beberapa teknologi yang digunakan untuk menunjuang pengerjaan proyek *Jitsi Bandwidth Reduction*.

### XMPP

*Extensible Messaging and Presence Protocol* (XMPP) adalah protokol jaringan berbasis XML yang digunakan untuk melakukan komunikasi secara *real-time* (*XMPP – Prosody IM*, 2018)*.* XMPP mendukung berbagai aplikasi termasuk di antaranya: perpesanan instan, obrolan multi-pihak, panggilan audio dan video, *middleware* ringan, sindikasi konten, dan penentuan rute data XML.

### Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems dan dirilis pada tahun 1995. Java tidak terikat pada perangkat keras atau sistem operasi tertentu (*platform-neutral*), dan didukung di berbagai sistem operasi, di antaranya: Windows, Mac OS, dan UNIX. Java dimodelkan dari bahasa pemrograman C dan C++ dengan modifikasi sedemikian rupa sehingga bahasa ini lebih canggih daripada pendahulunya. Java dirancang sebagai bahasa yang andal, portabel, sederhana, ringkas, dan interaktif (Devosa & Csallner, 2010).

Pada aplikasi Jitsi, bahaha pemrograman Java digunakan untuk membangun Jitsi Videobridge dan Jitsi Conference Focus. Di samping itu, kedua komponen ini pun menggunakan bahasa pemrograman Kotlin, namun pada lingkup proyek ini, tidak dibuat modifikasi untuk bagian program yang menggunakan bahasa Kotlin.

### JavaScript

JavaScript adalah bahasa skrip atau bahasa pemrograman yang digunakan untuk menerapkan fitur kompleks pada halaman web. JavaScript adalah salah satu dari tiga teknologi web standar (dua teknologi lainnya yaitu HTML dan CSS) (*What is JavaScript? - Learn web development | MDN*, 2019).

Pada aplikasi Jitsi, bahaha pemrograman JavaScript digunakan untuk membangun Jitsi Meet, sebagai *web frontend* aplikasi Jitsi.

### Lua

Lua adalah bahasa skrip yang mendukung berbagai jenis pemrograman, termasuk prosedural, berorientasi objek, fungsional, berbasis data, dan deskripsi data. Lua dikembangkan dan dikelola oleh sebuah tim dari Pontifical Catholic University of Rio de Janeiro di Brazil (*Lua: About*, 2018).

Pada aplikasi Jitsi, bahaha pemrograman Lua digunakan untuk membangun modul Prosody yang terdapat di Jitsi Meet.

### React

React adalah *library* JavaScript yang deklaratif, efisien, dan fleksibel untuk membangun *user interface* (UI) yang kompleks dari potongan-potongan kode yang disebut “komponen” (Facebook, 2020).

Pada aplikasi Jitsi, *library* React digunakan dalam pembangunan Jitsi Meet untuk mengelola fungsionalitas dan UI fitur pada komponen ini.

### AJAX

*Asynchronous JavaScript and XML* (AJAX) adalah metode penggunaan objek XMLHttpRequest untuk berkomunikasi dengan server yang dicetuskan oleh Jesse J. Garrett pada tahun 2005. AJAX digunakan untuk mengirim atau menerima informasi dalam format JSON, XML, HTML, atau teks. Selain itu, AJAX bersifat *asynchronous* sehingga komunikasi dengan server tetap dilakukan tanpa *refresh* situs/laman *website* (Luis et al., 2019).

Pada pengerjaan proyek ini, metode AJAX digunakan dalam pembuatan fungsi untuk pemanggilan fungsi RESTful API yang diimpelementasi di Jitsi Videobridge – *backend* – dari Jitsi Meet– *web frontend*.

### Axios

Axios adalah *library* JavaScript untuk membuat *request* HTTP dari node.js atau XMLHttpRequest dari browser. Axios mendukung API Promise untuk permintaan *asynchronous*, mengubah data *request* dan respons ke dalam format JSON secara otomatis, dan melindungi klien dari *client-site request forgery* (CSRF) (Axios Community, 2020).

Pada pengerjaan proyek ini, prmbuatan fungsi AJAX di Jitsi Meet menggunakan *library* Axios.

### Secure Shell

Secure Shell (SSH) adalah metode yang digunakan untuk mengakses perangkat lain, beserta data pada perangkat tersebut, secara *remote.* SSH menyediakan beberapa metode *authentication* yang kuat dan menjaga keamanan dan integritas komunikasi dengan enkripsi (ssh.com, 2020).

Pada pengerjaan proyek ini, metode SSH digunakan untuk mengakses server kantor secara *remote*. Kebutuhan untuk mengakses server kantor dalam rangka melakukan simulasi konferensi video, memeriksa *log*, dll.

## Perangkat Pendukung

Subbab ini berisi penjelasan ringkas mengenai bahasa pemrograman dan beberapa teknologi yang digunakan untuk menunjuang pengerjaan proyek *Jitsi Bandwidth Reduction*.

### Node.js

Node.js adalah platform pengembangan aplikasi di sisi server dan jaringan yang dikembangkan oleh Ryan Dahl dan dirilis pada tahun 2009. Node.js dibangun di atas JavaScript Engine (V8 Engine) Google Chrome. Aplikasi Node.js digunakan untuk mengeksekusi aplikasi yang dibangun menggunakan bahasa pemrograman JavaScript, dan didukung di berbagai sistem operasi, di antaranya: Mac OS X, Windows, dan Linux (Point, 2019).

Pada pengerjaan proyek ini, Node.js digunakan untuk mengeksekusi program Jitsi Meet di sisi server.

### IntelliJ IDEA

IntelliJ IDEA adalah *integrated development environment* (IDE) dari JetBrains yang dirilis pada tahun 2011 untuk pengembangan aplikasi dengan menggunakan bahasa pemrograman berbasis Java *bytecode* (*Berkenalan Dengan IntelliJ IDEA - Blog | Dicoding Indonesia | Blog | Dicoding Indonesia*, n.d.; D’Arcy, 2009)*.* Tidak hanya itu, pengguna dapat menggunakan bahasa pemrograman lain menggunakan plugin. IntelliJ IDEA didukung di sistem operasi Windows, Mac OS, dan Linux. IntelliJ IDEA menyediakan berbagai fitur, di antaranya: editor *source code*, otomatisasi *build*, *decompiler, debugger*, *code completion*, *automated refactoring tool*, dan terminal untuk menuliskan perintah secara manual.

Pada pengerjaan proyek ini, IntelliJ IDEA digunakan untuk menyunting komponen Jitsi Videobridge dan Jitsi Conference Focus.

### Visual Studio Code

Visual Studio Code adalah aplikasi editor *source code* yang dikembangkan oleh Microsoft untuk menyunting kode dalam berbagai bahasa pemrograman. Aplikasi memiliki perangkat pendukung *built-in* untuk bahasa pemrograman JavaScript, TypeScript, dan Node.js, serta ekstensi untuk berbagai bahasa lain, di antaranya: C ++, C #, Java, Python, dan PHP. Visual Studio Code menyediakan berbagai fitur, di antaranya: editor *source code*, *debugger*, dan terminal. Selain itu, aplikasi ini didukung di sistem operasi Windows, Mac OS, dan Linux (Visual Studio Code, 2015).

Pada pengerjaan proyek ini, Visual Studio Code digunakan untuk menyunting komponen Jitsi Meet.

### Windows Subsystem Linux

Windows Sub System Linux (WSL) adalah *platform* di sistem operasi Windows untuk menjalankan *environment* GNU/Linux. Saat ini, terdapat dua versi WSL, yaitu WSL 1 dan WSL 2. WSL 2 dikembangkan untuk menjalankan binary ELF64 Linux (Microsoft, 2018).

Pada pengerjaan proyek ini, WSL digunakan untuk menjalankan program Jitsi Meet secara lokal dengan menggunakan Node.js. Adapun Jitsi Meet dijalankan dengan menggunakan sistem operasi Linux dengan mengikuti dokumen *Developer Guide* (*Web)* yang dirilis pada *website* Jitsi Meet Handbook (*Developer Guide (Web) · Jitsi Meet Handbook*, n.d.).

### Ubuntu

Ubuntu adalah sistem operasi berbasis Linux yang dirancang untuk *platform* komputer, ponsel pintar, dan server. Ubuntu dikembangkan oleh perusahaan Canonical Ltd. dengan landasan prinsip pengembangan perangkat lunak *open source* (Point, n.d.).

### GitLab

GitLab adalah repositori Git berbasis *web* dan *platform* DevOps lengkap yang dikembangkan oleh Dmitriy Zaporozhets dan Valery Sizov untuk membantu tim mengembangkan perangkat lunak. Adapun, GitLab membantu tim dengan mening-katkan produktivitas dan efisiensi kolaborasi tim. GitLab menyediakan layanan *Continuous Integration/Delivery* (CI/CD) otomatis, *built-in* secara gratis untuk memudahkan kolaborasi tim. Selain itu, GitLab menyediakan fitur pelacak isu dan *bug*. GitLab dikembangkan pada tahun 2011 dengan menggunakan bahasa pemro-graman Ruby dan Go (*DevOps Platform Delivered as a Single Application | GitLab*, n.d.; Peham, 2020).

Pada pengerjaan proyek ini, GitLab digunakan untuk mengelola komponen yang digunakan dan dimodifikasi untuk kebutuhan proyek.

### AnyDesk

AnyDesk adalah aplikasi yang menyediakan layanan akses *remote* ke perangkat lain. Beberapa fitur yang disediakan AnyDesk, di antaranya akses dan transfer data antar perangkat, dan VPN. Anydesk didukung di sistem operasi Windows, Mac OS, iOS, Android, Linux, FreeBSD, Raspberry Pi, dan Chrome OS (*AnyDesk | What is AnyDesk - Javatpoint*, n.d.).

Pada pengerjaan proyek ini, Anydesk digunakan untuk mengakses server kantor secara *remote*. Kebutuhan untuk mengakses server kantor dalam rangka melakukan simulasi konferensi video, memeriksa *log*, dll. Selain itu, Anydesk digunakan untuk mengakses komputer rekan tim secara *remote*, misalkan untuk membantu rekan mengatasi eror, dll.

### GlobalProtect

GlobalProtect adalah program yang disediakan oleh Palo Alto Networks untuk mengamankan lalu lintas jaringan pengguna dengan menerapkan kemampuan Next-Generation Firewall untuk memeriksa lalu lintas jaringan pengguna, koneksi IPsec/SSL VPN untuk akses data di jaringan tanpa risiko, dan mengeliminasi titik buta di lalu lintas jaringan pengguna dengan visibilitas penuh ke semua lalu lintas jaringan, aplikasi, *port*, dan protokol (*GlobalProtect - Palo Alto Networks*, n.d.).

Pada pengerjaan proyek ini, perusahaan klien menggunakan GlobalProtect sebagai VPN yang digunakan tim dari PT. Padepokan Tujuh Sembilan untuk mengakses server klien.

### Google Chrome

Google Chrome adalah browser multi-*platform* yang dikembangkan oleh Google. Google Chrome menyediakan berbagai fitur, di antaranya: *address* *bar* multifungsi, penyimpanan kata sandi, dan *browser* yang terhubung dengan akun Google (*Helpful Features For Your Browser - Google Chrome*, n.d.).

Pada pengerjaan proyek ini, Google Chrome digunakan untuk melakukan simulasi konferensi video, melakukan pencarian untuk kebutuhan analisis, dll.

### Google Docs

Google Docs adalah layanan yang dikembangkan oleh Google untuk membuat dokumen dari berbagai platform. Google Docs terintegrasi penuh dengan layanan Google Drive sehingga pengguna dapat mengakses dan menyunting dokumen lebih mudah. Google Docs menyediakan berbagai fitur, di antaranya: berbagai *template* dokumen, akses bersama pengguna lain secara daring, penyimpanan hasil sunting dokumen secara otomatis, dan mendukung dokumen dengan format .docx (Google, 2019).

Pada pengerjaan proyek ini, Google Docs digunakan untuk membuat dokumentasi yang akan diserahkan kepada klien terkait progres proyek.

### WhatsApp

WhatsApp adalah aplikasi perpesanan multi-*platform* yang dirintis oleh Jan Koum dan Brian Acton pada tahun 2009 dan diakuisisi oleh Facebook pada tahun 2014 (WhatsApp Inc., 2018). WhatsApp menyediakan berbagai fitur, di antaranya: pesan dalam format teks, suara, gambar, dan video; enkripsi pesan teks, obrolan multi-pihak, dan panggilan suara dan video.

WhatsApp adalah media komunikasi utama yang digunakan dalam pengerjaan proyek ini. Pelaporan progres dan kendala, diskusi di dalam tim dilakukan dengan menggunakan aplikasi ini.

### Google Meet

Google Meet adalah aplikasi konferensi video daring yang disediakan oleh Google. Pengguna dapat membuat konferensi video di Google Meet secara gratis dengan syarat pengguna menggunakan akun Google untuk membuat konferensi tersebut. Aplikasi menyediakan berbagai fitur, di antaranya: tampilan audio dan video dan *screen-sharing* (*How to Use Google Meet Video Conferencing | Google Meet*, n.d.).

Google Meet adalah media komunikasi alternatif yang digunakan dalam pengerjaan proyek ini. Aplikasi ini digunakan apabila tim membutuhkan media komunikasi *real-time*, misal pada simulasi konferensi video di mana tim harus berkoordinasi dengan cepat dan efisien, dll.

### AirTabel

AirTabel adalah platform *hybrid* antara *spreadsheet* dan database untuk membuat *workflow.* AirTabel dapat digunakan untuk membuat *workflow* manajemen proyek, perencanaan acara besar, pengelolaan kalender editorial, dll. AirTabel dirilis oleh Howie Liu, Andrew Ofstad, dan Emmett Nicholas pada tahun 2013 (*AirTabel - Crunchbase Company Profile & Funding*, n.d.).

Pada pengerjaan proyek ini, AirTabel adalah media yang digunakan untuk membuat *workflow* pengerjaan proyek.

# ANALISIS DAN EVALUASI

Proyek *Jitsi Bandwidth Reduction* adalah proyek yang bertujuan untuk mengurangi penggunaan *bandwidth* pada konferensi video dengan menggunakan aplikasi Jitsi. Hal ini dilakukan untuk menjaga performa konferensi video sehingga komunikasi antar pengguna tidak terhambat. Untuk mencapai tujuan tersebut, klien berupaya untuk mengurangi penggunaan *bandwidth* dengan membatasi jumlah data (audio dan video) yang dapat diterima pengguna dalam kurun waktu tertentu.

Bab ini berisi penjabaran proses, hasil, dan evaluasi hasil analisis yang dilakukan selama pengerjaan proyek *Jitsi Bandwidth Reduction* untuk membatasi jumlah data yang dapat diterima pengguna dalam kurun waktu tertentu pada konferensi video dengan menggunakan aplikasi Jitsi.

## Analisis

Pada pengerjaan proyek ini, dilakukan analisis terdahap beberapa pendekatan untuk membatasi jumlah data yang dapat diterima pengguna dalam kurun waktu, yaitu *Last N* dan *pagination*. Skema *Last N* sudah diteliti sebelumnya oleh Boris Grozev et al. dan dipaparkan dalam sebuah makalah pada tahun 2015 (Grozev et al., 2015), sedangkan metode *pagination* adalah pendeketan yang diusulkan oleh klien dalam sebuah pertemuan via daring antara pihak klien dengan tim PT. Padepokan Tujuh Sembilan. Sehubungan dengan hal ini, analisis dilakukan terhadap tiga komponen penyusun Jitsi, yaitu: Jitsi Meet, Jitsi Videobridge (JVB) dan Jitsi Conference Focus (Jicofo).

Selanjutnya, subbab ini berisi penjabaran *timeline* proses, teknik yang digunakan dalam analisis, dan kendala/masalah yang dialami selama melakukan analisis pada pengerjaan proyek *Jitsi Bandwidth Reduction*.

### Timeline Analisis

Pada pengerjaan proyek *Jitsi Bandiwdth Reduction*, proses analisis yang dilakukan dibagi menjadi beberapa tahap sebelum didapatkan sebuah fungsi yang mumpuni untuk membatasi jumlah data yang diterima pengguna dalam kurun waktu tertentu. *Timeline* proses analisis dipaparkan pada Tabel IV.1.

Tabel IV.1. *Timeline* Proses Analisis

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Fokus Analisis | Durasi (hari) | Keterangan |
| 1 | Elemen *log* Prosody | 2 | Analisis ini dilakukan untuk mempelajari unsur elemen atribut stanza pada *log* Prosody dan alur proses konferensi video dengan menggunakan aplikasi Jitsi. |
| 2 | *Last N* | 2 | Penulis tidak banyak terlibat dalam analisis ini karena proses analisis dihentikan empat hari setelah penulis bergabung ke dalam tim proyek *Jitsi Bandwidth Reduction*. Selain itu, penulis pun ditugaskan untuk fokus mempelajari mate-ri lain. |
| 3 | Alur pengiriman paket di Jitsi Videobridge (JVB) | 3 | Analisis bertujuan untuk menghentikan pengi-riman paket (audio dan video) di JVB. Analisis dilakukan dengan bimbingan intensif dari manajer proyek dan komisaris PT. Padepokan Tujuh Sembilan, sekaligus pembimbing JTK Polban, Bapak Joe Lian Min. |
| 4 | Metode komunikasi pada Jitsi dengan menggunakan media berbasis XMPP (IQ stanza) dengan kostumisasi. |  | Analisis ini bertujuan untuk mengakses fungsi *paging* di JVB dari *web frontend* Jitsi Meet dengan menggunakan media komunikasi basis XMPP (IQ stanza) dengan penambahan atribut kostum berisi data yang akan dikirimkan untuk proses *pagination* di JVB. |
| 4.1 | Analisis Jitsi Meet | 7 | Analisis modul Prosody pada komponen Jitsi Meet untuk metode pembuatan dan pengiriman IQ stanza kostum. |
| 4.2 | Analisis *library* API Jitsi Meet (lib-jitsi-meet) | 5 | Analisis metode pembuatan dan pengiriman IQ stanza kostum *library* API Jitsi Meet (lib-jitsi-meet).  *Library* ini diperuntukkan bagi pengembang Jitsi untuk membangun *web frontend* Jitsi Meet dengan modifikasi GUI dalam skala besar. |
| 5 | Metode pemanggilan API dari Jitsi Meet |  | Analisis metode pemanggilan API untuk meng akses fungsi *paging* di JVB dari *web frontend* Jitsi Meet.  Analisis dibagi menjadi dua sesi, yaitu analisis IFrame API dan analisis metode pengambilan data untuk kebutuhan fungsi AJAX di Jitsi Meet. |
| 5.1 | Analisis IFrame API | 5 | Analisis ini bertujuan untuk mencari metode pemanggilan API di IFrame API.  *Library* ini diperuntukkan bagi pengembang Jitsi untuk membangun *web frontend* Jitsi Meet dengan modifikasi GUI dalam skala kecil. |
| 5.2 | Analisis metode pengambilan data untuk kebutuhan fungsi AJAX di Jitsi Meet | 6 | Analisis Jitsi Meet untuk mengambil data yang dibutuhkan dalam pembuatan fungsi AJAX (di Jitsi Meet). Proses analisis ini berjalan secara beriringan dengan impelementasi fungsi AJAX di Jitsi Meet. |
| 6 | Alur pengiriman paket di Jitsi Conference Focus (Jicofo) |  | Analisis dilakukan karena proses *pagination* di JVB tidak mendukung untuk jenis partisipan octo. Dengan analisis ini, diharapkan metode *pagination* dapat dipindahkan ke Jicofo. |
| 6.1 | Analisis metode pengambilan data untuk kebutuhan fungsi RESTful API di Jicofo | 2 | Analisis Jicofo untuk mengambil data yang dibutuhkan dalam pembuatan dua buah fungsi RESTful API, yaitu:   1. fungsi untuk mendapatkan semua konfe-rensi video yang sedang berlangsung di Jitsi; 2. fungsi untuk mendapatkan semua partisi-pan non-octo dan octo di ruang konferensi video.   Analisis untuk hal ini dilakukan secara berem-buk oleh seluruh anggota tim. |
| 6.2 | Analisis untuk menghentikan pengiriman paket di Jicofo | 5 | Analisis bertujuan untuk menghentikan pengi-riman paket (audio dan video) di Jicofo agar proses *pagination* dapat dipindahkan ke Jicofo. |
| 6.3 | Analisis partisipan octo | ~7 | Analisis ini bertujuan untuk mendapatkan ID partisipan octo. Analisis ini dilakukan di antara proses analisis lain dan berlanjut sampai pada analisis Jitsi Meet untuk penerapan *pagination* di *web frontend* berdasarkan urutan waktu *join* partisipan (6.2). |
| 6.4 | Analisis *source* *media* | 3 | Analisis ini bertujuan untuk mencari metode untuk memodifikasi isi konten paket (*source media*) pada Jicofo. Analisis dilakukan dengan arahan dari manajer proyek. |
| 7 | Penerapan *pagination* di *web frontend* berdasarkan urutan partisipan berdasarkan waktu *join* konferensi video. |  | Analisis bertujuan untuk meyakinkan urutan partisipan yang di-*mute* di JVB sesuai dengan urutan tampilan video pada *web frontend* Jitsi Meet.  Analisis dibagi menjadi dua sesi, yaitu analisis di JVB dan analisis di Jitsi Meet untuk menda-patkan urutan partisipan waktu *join* konferensi. |
| 7.1 | Analisis di JVB | 4 | Analisis bertujuan untuk mendapatkan metode pengambilan data waktu *join* partisipan di JVB. Analisis ini diselingi beberapa kegiatan untuk analisis partisipan octo sehingga waktu analisis yang dicantumkan lebih lama daripada waktu eksekusi sebenarnya. |
| 7.2 | Analisis di Jitsi Meet | 2 | Analisis bertujuan untuk mendapatkan metode pengambilan data waktu *join* partisipan di Jitsi Meet. |

### Teknik Analisis

Beberapa teknik yang digunakan dalam proses analisis pada pengerjaan proyek *Jitsi Bandiwdth Reduction* adalah sebagai berikut.

1. Melakukan penelusuran kode pada komponen yang menyusun Jitsi untuk mencari fungsi atau memahami alur suatu metode yang dibutuhkan dalam pengerjaan proyek. Untuk memperkecil ruang lingkup analisis pada suatu komponen, dilakukan pencarian kata kunci yang berkaitan dengan fokus analisis. Misalkan, pada analisis metode komunikasi pada Jitsi dengan menggunakan media berbasis XMPP (IQ stanza) yang dikostumisasi, dilakukan pencarian beberapa kata kunci, di antaranya: IQ, stanza, XMPP, dan Prosody untuk mencari unsur yang berkaitan dengan IQ stanza pada komponen Jitsi Meet.
2. Mencari sumber referensi dari internet untuk menunjang proses analisis dalam pengerjaan proyek. Referensi yang dipelajari adalah sumber yang kredibel, misalkan dokumen dari *website* resmi suatu teknologi/organisasi dan pendapat/gagasan dari *platform* tanya-jawab umum yang menyerta-kan bukti pendukung atau dicetuskan oleh narasumber dengan kredibilitas tinggi, misalnya: Damian Minkov (Damencho), salah satu pengembang aplikasi Jitsi.
3. Melakukan simulasi konferensi video di server kantor atau server klien (sesuai kebutuhan simulasi). Simulasi konferensi video dapat dilakukan untuk berbagai tujuan, di antaranya: memeriksa *log*, dilakukan pada analisis metode komunikasi pada Jitsi dengan menggunakan media berbasis XMPP (IQ stanza) yang dikostumisasi untuk mengecek kemunculan IQ stanza dari modul Prosody pada Jitsi Meet, atau pada analisis partisipan octo untuk mencari ID partisipan octo.
   1. mengecek tampilan *user interface*, dilakukan pada proses analisis penerapan *pagination* di *web frontend* berdasarkan urutan partisi-pan berdasarkan waktu *join* konferensi video untuk mendapatkan urutan tampilan video partisipan pada perangkat partisipan lokal dan menentukan hal-hal yang memengaruhi urutan tampilan video partisipan.
4. Diskusi dengan rekan tim untuk mengonfirmasi hasil suatu analisis atau meminta *insight* atau pendapat rekan tim ketika menghadapi kebingungan dalam proses analisis.
5. Meminta bantuan manajer proyek ketika menghadapi kendala/masalah dalam proses analisis yang tidak dapat diselesaikan sendiri.

### Kendala Analisis

Pada proses analisis yang dilakukan selama pengerjaan proyek, dihadapi beberapa kendala/masalah yang – utamanya – disebabkan oleh ruang lingkup analisis yang luas. Adapun, hal ini karena aplikasi Jitsi dibangun oleh beberapa komponen yang – masing-masing – relatif cukup besar dan kompleks. Beberapa kendala/masalah yang ditimbulkan oleh perkara lingkup analisis yang luas, di antaranya:

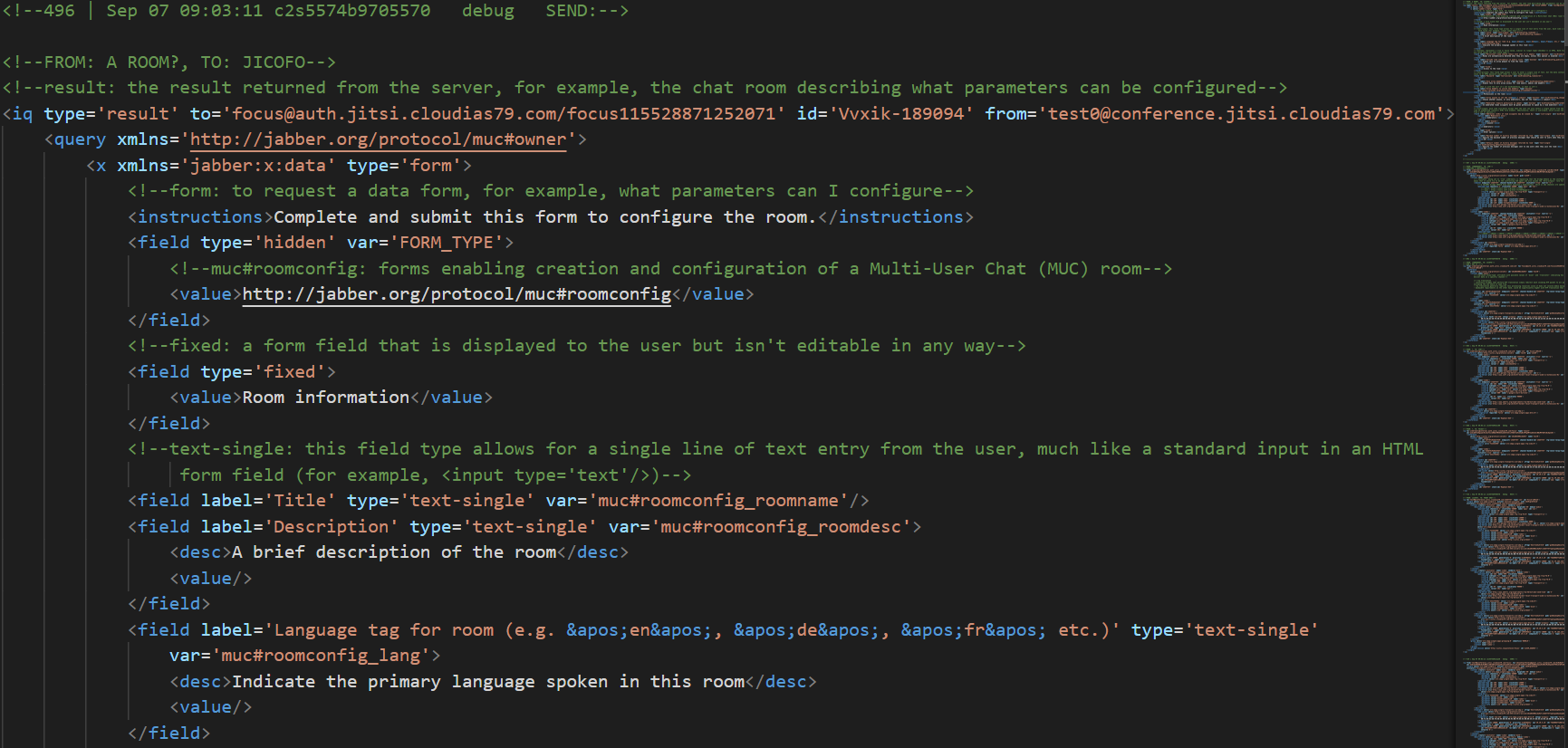
1. analisis untuk setiap topik/fokus masalah memakan waktu lama walaupun sudah diusahakan untuk memperkecil ruang lingkup analisis.
2. keraguan dalam mengakhiri dan menyimpulkan analisis suatu topik/fokus karena tidak yakin apakah analisis sudah dilakukan terhadap semua unsur yang berkaitan dengan fokus analisis.

## Hasil Analisis

Subbab ini berisi penjabaran hasil analisis yang telah dilakukan pada pengerjaan proyek *Jitsi Bandwidth Reduction*. Adapun, beberapa hasil analisis yang dijabarkan pada bagian ini adalah hasil analisis yang dilakukan rekan tim yang berdampak pada keputusan/kesimpulan analisis.

### Log Prosody

Analisis *log* Prosody dilakukan pada awal pengerjaan proyek untuk mempelajari alur proses konferensi video dengan menggunakan aplikasi Jitsi. Proses analisis ini dimulai dengan mempelajari penggunaan protokol XMPP dalam komunikasi pada aplikasi *Multi-User Chat* (MUC). Kemudian, dilakukan analisis atribut dua ekstensi protokol XMPP yang digunakan pada konferensi video di Jitsi, yaitu *COnferences with LIghtweight BRIdging* (COLIBRI) dan Jingle. Pada Gambar IV.1 ditampilkan potongan informasi *log* Prosody yang didapatkan dari sebuah sesi konferensi video dengan menggunakan aplikasi Jitsi.



Gambar IV.1. Potongan *Log* Prosody

### Last N

Pada subbab sebelumnya, disebutkan bahwa penulis tidak banyak terlibat dalam analisis ini karena proses analisis dihentikan empat hari setelah penulis bergabung ke dalam tim proyek *Jitsi Bandwidth Reduction*. Dan, dalam selang waktu tersebut, penulis ditugaskan untuk fokus mempelajari materi lain yang berhubungan dengan proyek. Adapun, analisis *Last N* dihentikan karena metode pendekatan ini tidak sesuai dengan kebutuhan klien karena metode pengiriman paket video yang berdasarkan pada aktivitas audio partisipan.

Berdasarkan hasil penelitian yang telah dilakukan Boris Grozev et al. (2015), aliran data partisipan diubah sedemikian rupa sehingga mengalir secara dinamis sesuai dengan aktivitas audio partisipan. Dengan kata lain, partisipan lokal hanya dapat menerima data video dari partisipan *remote* dengan audio aktif (partisipan *speaker*). Namun, dari hasil penelitian pun disebutkan bahwa partisipan lokal dapat memilih untuk tetap menerima data video dari partisipan *remote* dengan aktivitas audio nihil (Grozev et al., 2015). Sehubungan dengan hal ini, penulis tidak mengetahui analisis yang telah dilakukan dalam upaya untuk memodifikasi alur skema *Last N* agar sesuai dengan kebutuhan proyek ini.

### Alur Pengiriman Paket di Jitsi Videobridge (JVB)

Pada analisis alur pengiriman paket di JVB, ruang lingkup analisis difokuskan pada dua buah *method*, yaitu:

1. sendOut(), prosedur untuk mendistribusikan paket ke setiap partisipan  
   yang menginginkan paket tersebut,
2. wants(), fungsi untuk mengecek kesediaan/keinginan partisipan dalam menerima paket.

Pada *method* sendOut(), fungsi wants() dipanggil untuk mengecek kesediaan/ke-inginan partisipan dalam menerima paket. Apabila partisipan bersedia untuk mene-rima (menginginkan) suatu paket, JVB akan mengirimkan paket tersebut kepada partisipan. Begitu pula sebaliknya, jika partisipan tidak menginginkan (menolak) paket, JVB tidak akan mengirimkan paket tersebut.

Tabel IV.2. *Method* sendOut()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/Conference.java |
| *Source code* | private void sendOut(PacketInfo packetInfo)  {  String sourceEndpointId = packetInfo.  getEndpointId();    PotentialPacketHandler prevHandler = null;  for (Endpoint endpoint : endpointsCache)  {  if (endpoint.getID().equals(sourceEndpointId))  {  continue;  }  **// pemanggilan fungsi wants() untuk pengiriman**  **// kepada partisipan non-octo**  if (endpoint.wants(packetInfo))  {  if (prevHandler != null)  {  prevHandler.send(packetInfo.clone());  }  prevHandler = endpoint;  }  }  // pemanggilan fungsi wants() untuk pengiriman  // kepada partisipan octo  if (tentacle != null && tentacle.wants(packetInfo))  {  if (prevHandler != null)  {  prevHandler.send(packetInfo.clone());  }  prevHandler = tentacle;  }  if (prevHandler != null)  {  prevHandler.send(packetInfo);  }  else  }  ByteBufferPool.returnBuffer(packetInfo.  getPacket().getBuffer());  }  } |

Dalam hal ini, yang dimaksud dengan partisipan “menginginkan” paket bukan partisipan dapat memilih untuk menerima/menolak pengiriman paket, melainkan partisipan dapat menerima jenis paket yang didistribusikan (paket RTP atau RTCP) dan JVB dapat terhubung dengan jalur transportasi paket milik partisipan tersebut. Singkat kata, fungsi wants() bukan bertujuan untuk digunakan oleh partisipan untuk memilih menerima/menolak paket dari JVB. Pada JVB, tidak terdapat fungsi untuk tujuan tersebut. Bilapun ada, fungsi tersebut tidak ditemukan pada proses analisis ini.

Tabel IV.3. *Method* wants()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/Endpoint.java |
| *Source code* | public boolean wants(PacketInfo packetInfo)  {  **// mengecek apakah JVB dapat terhubung dengan jalur**  **// transportasi paket partisipan**  if (!isTransportConnected())  {  return false;  }  Packet packet = Objects.requireNonNull(packetInfo.  getPacket(), "packet");  **// mengecek apakah paket yang didistribusi adalah**  **// paket RTP**  if (packet instanceof RtpPacket)  {  if (packet instanceof VideoRtpPacket)  {  return acceptVideo && bitrateController.  accept(packetInfo);  }  if (packet instanceof AudioRtpPacket)  {  return acceptAudio;  }  }  **// mengecek apakah paket yang didistribusi adalah**  **// paket RTCP**  else if (packet instanceof RtcpPacket)  {  if (packet instanceof RtcpSrPacket)  {  return bitrateController.accept(  (RtcpSrPacket) packet);  }  else if (packet instanceof RtcpFbPliPacket ||  packet instanceof RtcpFbFirPacket)  {  return true;  }  else  {  logger.warn("Ignoring a rtcp packet of  type" + packet.getClass().getSimpleName());  return false;  }  }  logger.warn("Ignoring an unknown packet type:" +  packet.getClass().getSimpleName());    return false;  } |

### Media Komunikasi Berbasis XMPP dengan Kostumisasi

Analisis metode komunikasi pada Jitsi dengan menggunakan media berbasis XMPP (IQ stanza) yang dikostumisasi dibagi menjadi dua sesi, yaitu analisis Jitsi Meet dan analisis *library* API Jitsi Meet (lib-jitsi-meet).

Pada komponen Jitsi Meet, dilakukan analisis terhadap modul Prosody, yaitu media komunikasi untuk Jitsi Meet yang dibuat dalam bahasa pemrograman Lua. Dari hasil penelusuran yang dilakukan, ditemukan sebuah fungsi yang bertujuan untuk pembuatan IQ stanza pada modul Prosody. Fungsi tersebut diuraikan pada Tabel IV.4.

Tabel IV.4. Fungsi sendIq()

|  |  |
| --- | --- |
| *Path* | resources/prosody-plugins/mod\_jibri\_queue\_component.lua |
| *Source code* | local function sendIq(participant,action,requestId,time,position,token)  local iqId = uuid\_gen();  local from = module:get\_host();  local outStanza = st.iq({type = 'set', from = from,  to = participant, id = iqId}):tag("jibri-queue",  { xmlns = 'http://jitsi.org/protocol/jibri-  queue', requestId = requestId, action = action});  if token then  outStanza:tag("token"):text(token):up()  end  if time then  outStanza:tag("time"):text(tostring(time)):  up()  end  if position then  outStanza:tag("position"):text(tostring(  position)):up()  end  module:send(outStanza);  end |

Fungsi pembuatan IQ stanza pada awalnya akan dibuat dengan mencontoh pada fungsi sendIq(), namun, analisis lebih lanjut modul Prosody menunjukkan bahwa modul Prosody pada Jitsi Meet tidak sesuai dengan kebutuhan tim untuk mengakses fungsi *paging* di JVB dari *web frontend* Jitsi Meet.

Ditemukan petunjuk yang mengarah pada *library* API Jitsi Meet (lib-jitsi-meet) sehingga proses analisis dialihkan ke *library* tersebut.

Hasil penelusuran di lib-jitsi-meet mengarahkan pada beberapa fungsi pembuatan dan pengiriman IQ stanza. Pembuatan IQ stanza di *library* ini dengan menggunakan Strophe.js, *library* XMPP untuk JavaScript. Adapun, konsep fungsi pembuatan IQ stanza yang sebelumnya akan dibuat mencontoh pada fungsi yang diuraikan pada Tabel IV.5.

Tabel IV.5. Fungsi createNonAnonymousRoom()

|  |  |
| --- | --- |
| *Path* | modules/xmpp/ChatRoom.js |
| *Source code* | createNonAnonymousRoom() {  if (this.options.disableDiscoInfo) {  return;  }  const getForm = $iq({ type: 'get',to:this.roomjid})  .c('query', { xmlns: 'http://jabber.org/protocol/  muc#owner' })  .c('x', { xmlns: 'jabber:x:data',type: 'submit' });  this.connection.sendIQ(getForm, form => {  if (!$(form).find(  '>query>x[xmlns="jabber:x:data"]' +  '>field[var="muc#roomconfig\_whois"]').  length) {  const errmsg = 'non-anonymous rooms not  supported';  GlobalOnErrorHandler.callErrorHandler(  new Error(errmsg));  logger.error(errmsg);  return;  }  const formSubmit = $iq({ to: this.roomjid,  type: 'set' })  .c('query',{xmlns: 'http://jabber.org/protocol/  muc#owner' });  formSubmit.c('x', {xmlns: 'jabber:x:data',type:  'submit' });  formSubmit.c('field', { 'var': 'FORM\_TYPE' })  .c('value')  .t('http://jabber.org/protocol/  muc#roomconfig').up().up();  formSubmit.c('field', { 'var': 'muc#roomconfig\_  whois' })  .c('value').t('anyone').up().up();  this.connection.sendIQ(formSubmit);  }, error => {  GlobalOnErrorHandler.callErrorHandler(error);  logger.error('Error getting room configuration  form: ', error);  });  } |

Analisis ini dihentikan setelah ditemukan sebuah referensi yang menyatakan bahwa kostumisasi IQ stanza tidak dapat dilakukan karena teknologi XMPP yang tidak mendukung kebutuhan tersebut (*Add ability to invite people by JID · Issue #6235 · jitsi/jitsi-meet · GitHub*, n.d.).

### Metode Pemanggilan API dari Jitsi Meet

Analisis metode pemanggilan API adalah pendekatan alternatif untuk mengakses fungsi *paging* di JVB dari *web frontend* Jitsi Meet setelah didapatkan informasi bahwa komunikasi dengan menggunakan media komunikasi berbasis XMPP yang dikostumisasi tidak didukung oleh teknologi terkait. Analisis dibagi menjadi dua sesi, yaitu analisis IFrame API dan analisis metode pengambilan data untuk kebu-tuhan fungsi AJAX di Jitsi Meet.

Analisis IFrame API tidak membuahkan hasil yang diharapkan. Adapun, analisis ini cukup berlarut-larut karena petunjuk yang sedikit dan rancu sehingga memakan waktu yang cukup lama. Keputusan akhir analisis IFrame API yaitu *library* ini tidak merepresentasikan kebutuhan tim untuk mengakes fungsi *paging* di JVB.

Analisis untuk kebutuhan fungsi AJAX di Jitsi Meet sebelumnya diawali dengan pembuatan *button* untuk kebutuhan interaksi pengguna. Analisis dilakukan dari sisi React dan HTML. Adapun, analisis di sisi React dilakukan pada fitur Toolbox Jitsi Meet. Fitur ini dipilih karena pada fitur ini, ditangani pembuatan banyak *button* di Jitsi Meet. Namun, analisis pada fitur ini tidak mencapai titik terang. Solusi untuk pembuatan *button* yaitu dengan menambahkan elemen *button* di fitur Filmstrip pada fungsi render() React.

Pembuatan fungsi AJAX pada komponen Jitsi Meet dengan menggunakan *library* Axios. Hal ini diputuskan setelah dilakukan perbandingan antara *library* Axios dan Fetch untuk membuat fungsi pemanggilan API pada aplikasi React. Beberapa pertimbangan dalam memilih Axios, yaitu:

1. Axios mengonversi data menjadi JSON secara otomatis,
2. Axios didukung di lebih banyak browser,
3. Axios bersifat *asynchronous*.

Secara singkat, Axios lebih unggul dibandingkan Fetch di berbagai aspek sehingga *library* ini dipilih.

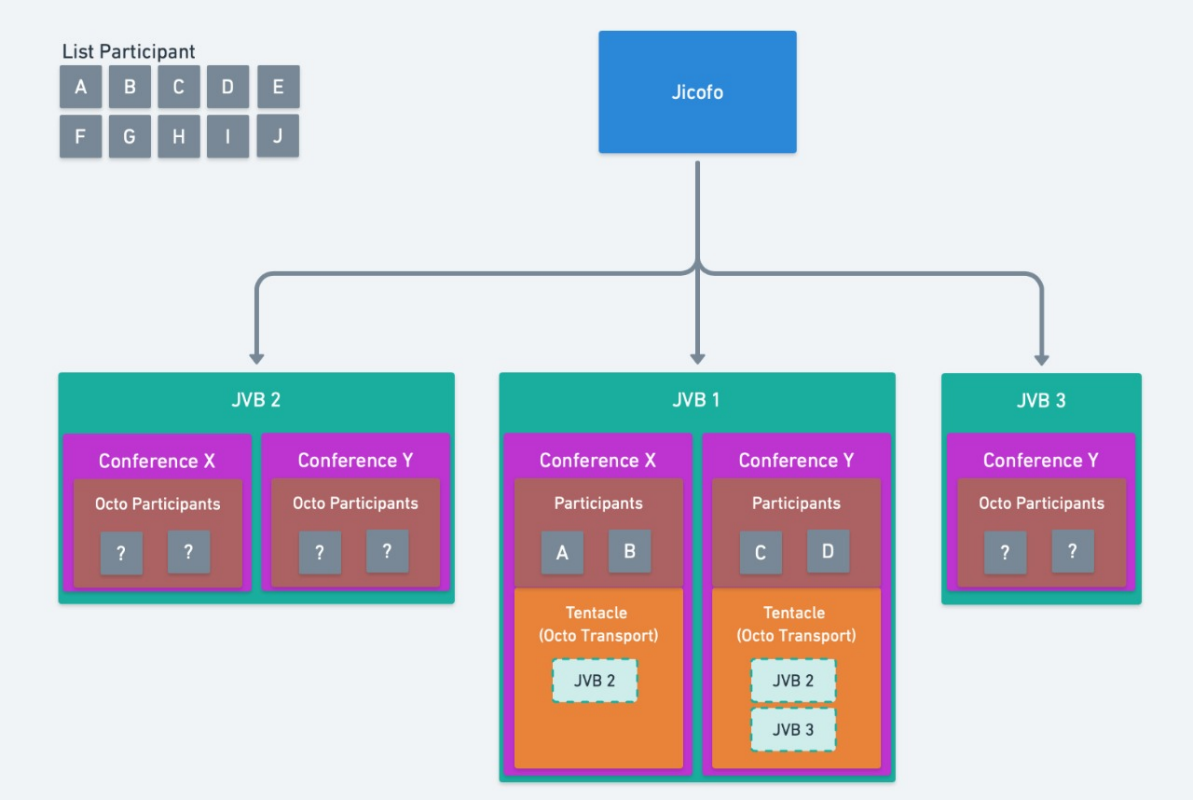
Metode pengambilan data untuk kebutuhan fungsi AJAX di Jitsi Meet yaitu dengan mengambil data yang disimpan pada *redux state* fitur terkait. Beberapa data yang diperlukan pada pembuatan fungsi AJAX adalah ID partisipan lokal, jumlah total partisipan, dan ID konferensi di Jitsi Meet. Pengambilan data ID partisipan lokal dan jumlah total partisipan dapat dilakukan tanpa mengalami kendala apapun. Akan tetapi, tidak begitu untuk pengambilan ID konferensi.

Sebauh referensi di internet mengatakan bahwa ID konferensi di Jitsi Meet dan ID konferensi yang digunakan di JVB berbeda. Adapun, ID konferensi yang digunakan pada JVB sebetulnya adalah ID COLIBRI (ID universal sebuah ruang konferensi pada server Jitsi) dan tidak dapat dicocokkan dengan ID konferensi pada Jitsi Meet. Bilapun ada, metode untuk mencocokkan kedua ID sangat rumit (*How to match the conference ID between Jitsi-meet and Rest Colibri? - Developers - Jitsi Community Forum - developers & users*, n.d.). Dengan kata lain, data ID konferensi pada Jitsi Meet tidak sesuai dengan kebutuhan data untuk proses *pagination* di JVB. Untuk mengatasi kendala tersebut, data ID konferensi diambil langsung dari JVB, sedang-kan fungsi AJAX di Jitsi Meet akan mengirimkan kebutuhan data lainnya.

### Partisipan Octo

Analisis lebih lanjut pada *method* sendOut() di JVB mengarahkan pada penemuan dua jenis partisipan pada konferesi daring dengan menggunakan aplikasi Jitsi, yaitu partisipan octo dan partisipan non-octo. Pemanggilan *method* wants() pada *method* sendOut() tidak hanya untuk mengecek keinginan partisipan non-octo (*Endpoint*), namun juga keinginan partisipan octo (*ConfOctoTransport*). *ConfOctoTransport* berfungsi untuk mengatur pengiriman paket ke partisipan octo di JVB.

Adapun, Partisipan octo adalah partisipan di suatu ruang konferensi daring yang berasal dari JVB yang berbeda dengan JVB yeng menyediakan layanan untuk ruang konferensi, sedangkan partisipan non-octo adalah partisipan yang berasal dari JVB peyedia layanan untuk ruang konferensi tersebut. Pada Gambar IV.1 adalah ilustrasi parti-sipan octo dan non-octo pada layanan Jitsi.



Gambar IV.2. Ilustrasi Partisipan Octo dan Non-Octo pada Jitsi

Pada ilustrasi Gambar IV.2, diilustrasikan JVB 1 adalah penyedia layanan untuk konferensi X. Pada konferensi X, partisipan A dan B adalah partisipan non-octo, sedangkan partisipan yang berasal dari JVB 2 adalah partisipan octo. Tentacle (objek *Octo Transport*) pada ruang konferensi X di JVB 1 berperan sebagai jalur transportasi paket ke partisipan octo.

Partisipan OCTO akan terbentuk di ruang konferensi apabila JVB penyedia layanan sudah mencapai batas maksimal partisipan yang dapat ditangani dalam kurun waktu tertentu. Sumber referensi mengatakan bahwa batas maksimal partisipan yang dapat dilayani sebuah JVB saat ini adalah 75 partisipan (*Jisti Conference with Up To 100 Participant - Install & Config - Jitsi Community Forum - developers & users*, n.d.).

Pada analisis alur pengiriman paket di JVB, perbedaan antara partisipan octo dan non-octo tidak didalami karena diarahkan untuk fokus pada alur pengiriman paket ke partisipan non-octo. Analisis partisipan octo dialihkan ke Jicofo karena analisis yang telah dilakukan sebelumnya untuk menghentikan pengiriman paket dari JVB ke partisipan octo tidak membuahkan hasil yang diharapkan.

### Alur Pengiriman Paket di Jitsi Conference Focus (Jicofo)

Analisis alur pengiriman paket di Jicofo dilakukan karena proses *pagination* di JVB tidak mendukung untuk penanganan partisipan octo. Dengan dilakukan analisis ini, diharapkan metode *pagination* dapat dipindahkan ke Jicofo.

Analisis di Jicofo dibagi menjadi empat sesi, yaitu:

1. analisis metode pengambilan data untuk kebutuhan fungsi RESTful API,
2. analisis untuk menghentikan pengiriman paket di Jicofo,
3. analisis partisipan octo, dan
4. analisis source media.

Analisis Jicofo untuk pengambilan data dibutuhkan dalam pembuatan dua fungsi RESTful API, yaitu:

1. fungsi untuk mendapatkan semua konferensi video yang tersambung dengan Jicofo,
2. fungsi untuk mendapatkan semua partisipan non-octo dan octo di ruang konferensi video.

Adapun, pembuatan kedua fungsi ini bertujuan untuk mencari ID partisipan octo di ruang konferensi. Pada fungsi kedua, *identifier* partisipan octo di Jicofo adalah ID relay. Asumsi awal penggunaan ID relay yaitu ID akan mengidentifikasi partisipan octo sebagaimana ID partisipan dapat membedakan satu partisipan dengan partisi-pan lainnya. Akan tetapi, kenyataannya ID relay partisipan octo pada Jicofo adalah IP *address* JVB yang menyediakan layanan untuk partisipan tersebut. Hal ini karena pada JVB, pengiriman paket ke partisipan octo dilakukan secara bersamaan untuk semua partisipan octo di JVB yang sama. Pencarian lebih lanjut di Jicofo untuk mendapatkan ID partisipan octo tidak membuahkan hasil yang diharapkan tim.

Analisis untuk menghentikan pengiriman paket dilakukan dengan menelusuri dua *method* di Jicofo, yaitu *method* createOffer() dan muteParticipant().

*Method* createOffer() berfungsi untuk membuat Jingle *offer* berisi paket yang akan dikirimkan ke semua partisipan di ruang konferensi. Dari hasil penelusuran *method* ini, diketahui *method* hanya dieksekusi sekali untuk pengiriman satu jenis paket di Jicofo. Dengan kata lain, pengiriman paket untuk semua partisipan dilakukan secara bersamaan. Paket yang dikirimkan pada *method* ini tidak dapat dihapus karena akan berpengaruh pengiriman paket untuk semua partisipan.

Tabel IV.6. *Method* createOffer()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/ParticipantChannelAllocator.java |
| *Source code* | protected List<ContentPacketExtension> createOffer()  throws UnsupportedFeatureConfigurationException  {  EntityFullJid address = participant.getMucJid();  List<String> features = DiscoveryUtil.  discoverParticipantFeatures(meetConference.  getXmppProvider(), address);    participant.setSupportedFeatures(features);  JitsiMeetConfig config = meetConference.  getConfig();  OfferOptions offerOptions = new OfferOptions();  OfferOptionsKt.applyConstraints(offerOptions,  config);  OfferOptionsKt.applyConstraints(offerOptions,  participant);  JingleOfferFactory jingleOfferFactory =  FocusBundleActivator.getJingleOfferFactory();    return jingleOfferFactory.  createOffer(offerOptions);  } |

Analisis dengan sudut pandang yang berkebalikan dengan *method* createOffer(), yaitu mencari metode agar dapat menentukan partisipan yang akan menerima paket pada Jingle *offer*, pun tidak menemukan titik terang.

*Method* mutePartisipan() berfungsi untuk *mute* paket audio partisipan *remote* dari Jicofo. Dari hasil penelusuran dan eksplorasi *method* ini, diketahui bahwa *method* ini tidak dapat digunakan untuk menghentikan pengiriman paket video. *Method* ini menggunakan media komunikasi berbasis XMPP untuk mengirimkan pesan bahwa partisipan melakukan *mute*.

Tabel IV.7. *Method* muteParticipant()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/impl/protocol/xmpp/colibri/ColibriConferenceImpl.java |
| *Source code* | public boolean muteParticipant(ColibriConferenceIQ channelsInfo, boolean mute)  {  if (checkIfDisposed("muteParticipant"))  {  return false;  }  ColibriConferenceIQ request =  new ColibriConferenceIQ();  request.setID(conferenceState.getID());  request.setName(conferenceState.getName());  **// jenis paket yang di-mute adalah audio**  ColibriConferenceIQ.Content audioContent =  channelsInfo.getContent("audio");  if (audioContent == null || isBlank(request.  getID()))  {  logger.error("Failed to mute - no audio content ." + " Conf ID: " + request.getID());    return false;  }  ColibriConferenceIQ.Content requestContent  = new ColibriConferenceIQ.Content(audioContent. getName());  **// pengiriman IQ stanza**  for (ColibriConferenceIQ.Channel channel :  audioContent.getChannels())  {  ColibriConferenceIQ.Channel requestChannel  = new ColibriConferenceIQ.Channel();  requestChannel.setID(channel.getID());  **// pengaturan jalur pengiriman paket**  requestChannel.setDirection(mute ?  MediaDirection.SENDONLY.toString() :  MediaDirection.SENDRECV.toString());  requestContent.addChannel(requestChannel);  }  if (requestContent.getChannelCount() == 0)  {  logger.error("Failed to mute - no channels to  modify." + " ConfID:" + request.getID());  return false;  }  request.setType(IQ.Type.set);  request.setTo(jitsiVideobridge);  request.addContent(requestContent);  connection.sendStanza(request);  return true;  } |

Dari hasil penelusruan *method* muteParticipant(), didapatkan sebuah metode untuk pengaturan jalur pengiriman paket, yaitu dengan mengubah *direction* (arah/jalur) media yang akan dikirimkan. Jenis MediaDirection dipaparkan pada Tabel IV.8.

Tabel IV.8. Jenis MediaDirection

|  |  |
| --- | --- |
| Jenis MediaDirection | Keterangan |
| INACTIVE | partisipan tidak dapat menerima dan mengirimkan paket dari/ke partisipan lain di konferensi |
| SENDONLY | partisipan tidak dapat menerima paket dari partisipan lain pada konferensi, namun dapat mengirimkan paket pada partisipan lain di konferensi |
| RECVONLY | partisipan dapat menerima paket dari partisipan lain pada konferensi, namun tidak dapat mengirimkan paket pada partisipan lain di konferensi |
| SENDRECV | partisipan dapat menerima dan mengirimkan paket dari/ke partisipan lain di konferensi |

MediaDirection tidak diatur pada komponen Jicofo, melainkan di JVB. Dari hasil pencarian di JVB, pengaturan jalur media partisipan hanya ditemukan pada fungsi updateMediaDirection(). Akan tetapi, berdasarkan keterangan pada fungsi tersebut, fungsi tidak digunakan di JVB, namun digunakan pada plugin.

Tabel IV.9. *Method* updateMediaDirection()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/Endpoint.java |
| *Source code* | **// Komentar dari salah satu pengembang Jitsi**  **@SuppressWarnings("unused") // Used by plugins (Yuri)**  public void updateMediaDirection(MediaType type, Stringdirection)  {  switch (direction)  {  case "sendrecv":  case "sendonly":  case "recvonly":  case "inactive":  {  for(ChannelShim channelShim : channelShims)  {  if (channelShim.getMediaType() == type)  {  channelShim.  setDirection(direction);  }  }  break;  }  default:  throw new IllegalArgumentException("Media  direction unknown: " + direction);  }  } |

Penelusuran lebih lanjut pada *method* muteParticipant() mengarahkan pada *method* handleMuteIq() untuk menangani pengiriman IQ yang dibutuhkan untuk proses *mute* audio partisi-pan di Jicofo. Fungsi ini diuraikan pada Tabel IV.8.

Tabel IV.10. Potongan *Method* handleMuteIQ()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/MeetExtensionsHandler.java |
| *Source code* | private IQ handleMuteIq(MuteIq muteIq)  {  Boolean doMute = muteIq.getMute();  Jid jid = muteIq.getJid();  . . .  } |

Untuk dapat dieksekusi, *method* handle handleMuteIQ() membutuhkan parameter sebuah objek MuteIq. Dan, ditunjukkan pada Tabel IV.11, objek MuteIq memiliki atribut variabel NAMESPACE. Dari beberapa sumber referensi, atribut *namespace* (XML namespace – xmlns) tidak dapat dimodifikasi seusai kebutuhan klien (*Mute video - Developers - Jitsi Community Forum - developers & users*, n.d.; *Mute Video For Remote User - Install & Config - Jitsi Community Forum - developers & users*, n.d.). Kasus serupa terjadi pada pembuatan IQ stanza kostum untuk pemanggilan fungsi *paging* di JVB dari Jitsi Meet.

Tabel IV.11. Potongan Kode *Class* MuteIq

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/MeetExtensionsHandler.java |
| *Source code* | public class MuteIq extends IQ  {  **// XML namespace (xmlns) untuk ekstensi mute audio**  public static final String NAMESPACE =  "http://jitsi.org/jitmeet/audio";  . . .  } |

Analisis terkahir pada Jicofo yaitu analisis *source media* untuk memodifikasi konten paket (*source media*) yang dikirimkan keapda partisipan. Analisis dilakukan dengan menelusuri program dan melakukan simulasi konferensi video untuk mengecek atribut setiap paket yang diterima partisipan. Adapun, tujuan dilakukan pengecekan atribut adalah untuk mendapatkan informasi pemilik paket. Tetapi, dari hasil penelusuran, didapatkan bahwa Jicofo menetapkan *room* (ruang) konferensi sebagai pemilik paket yang didistribusikan, bukan partisipan. Penetapan pemilik paket pada *method* claimSources() yang diuraikan pada Tabel IV.12.

Tabel IV.12. *Method* claimSources()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/Participant.java |
| *Source code* | public void claimSources(MediaSourceMap sourceMap)  {  **// menetapkan pemilik paket**  Jid roomJid = roomMember.getOccupantJid();  sourceMap  .getMediaTypes()  .forEach(  mediaType -> sourceMap  .getSourcesForMedia(mediaType)  .forEach(source -> SSRCSignaling.  setSSRCOwner(source, roomJid)));  } |

### Penerapan *Pagination* di Jitsi Meet

Analisis bertujuan untuk meyakinkan urutan partisipan yang di-*mute* di JVB sesuai dengan urutan tampilan video pada *web frontend* Jitsi Meet. Analisis dilakukan di JVB dan di Jitsi Meet untuk mendapatkan urutan partisipan berdasarkan waktu *join* konferensi. Akan tetapi, analisis di JVB tidak membuahkan hasil yang memenuhi kebutuhan tim. Hal ini karena waktu *join* partisipan di Jitsi Meet tidak disimpan di JVB.

Dari hasil penelusuran di Jitsi Meet, metode untuk mendapatkan urutan partisipan berdasarkan waktu *join* (bergabung) ke ruang konferensi dilakukan setelah proses pembuatan *layout* video partisipan *remote* pada monitor perangkat partisipan lokal.

Tabel IV.13. Fungsi addRemoteParticipantContainer()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/MeetExtensionsHandler.java |
| *Source code* | addRemoteParticipantContainer(participant) {  if (!participant || participant.local) {  return;  } else if (participant.isFakeParticipant) {  const sharedVideoThumb = new SharedVideoThumb(  participant,  SHARED\_VIDEO\_CONTAINER\_TYPE,  VideoLayout);  this.addRemoteVideoContainer(participant.id,  sharedVideoThumb);  return;  }  const id = participant.id;  const jitsiParticipant = APP.conference.  getParticipantById(id);  **// pembuatan layout video partisipan remote**  **// untuk menentukan waktu *join* partisipan**  const remoteVideo =  new RemoteVideo(jitsiParticipant, VideoLayout);  this.\_setRemoteControlProperties(jitsiParticipant, remoteVideo);  this.addRemoteVideoContainer(id, remoteVideo);  this.updateMutedForNoTracks(id, 'audio');  this.updateMutedForNoTracks(id, 'video');  } |

## Evaluasi Analisis

Subbab ini berisir penjabaran evaluasi dari hasil analisis yang telah dilakukan pada pengerjaan proyek *Jitsi Biandwidth Reduction*. Adapun, beberapa poin penting dari hasil analisis adalah sebagai berikut.

1. Hasil analisis menunjukkan bahwa modifikasi pada alur pengiriman paket tidak dapat dilakukan di Jitsi Meet atau Jitsi Conference Focus (Jicofo). Pengaturan untuk pengiriman/penerimaan paket dapat dilakukan di Jitsi Videobridge (JVB). Bilapun pada komponen Jitsi Meet dan Jicofo terdapat fungsi untuk tujuan tersebut, fungsi tersebut tidak dapat ditemukan pada analisis ini.
2. Teknologi XMPP tidak mendukung untuk memodifikasi media komuni-kasi berbasis XMPP. Oleh sebab itu, komunikasi antar komponen yang dibuat pada proyek ini dilakukan dengan metode pemanggilan API melalui HTTP.
3. Tidak terdapat metode untuk menghentikan pengiriman paket kepada partisipan octo baik di JVB maupun Jicofo. Adapun, metode pengiriman paket kepada partisipan octo yang ditemukan pada sebuah *library* di JVB tidak dimodifikasi karena kekhawatiran modifikasi akan berpengaruh pada fungionalitas unsur lain di JVB.
4. Metode untuk mendapatkan urutan partisipan berdasarkan waktu *join* tidak dapat diimplementasi di JVB karena tidak terdapat fungsi untuk memenuhi tujuan tersebut di JVB, sehingga analisis dan penerapan metode dialihkan ke komponen Jitsi Meet.

# PERANCANGAN

Bab ini berisi penjabaran proses dan hasil perancangan yang dilakukan selama pengerjaan proyek *Jitsi Bandwidth Reduction*. Perancangan pada proyek ini tidak dilakukan secara formal; perancangan dilakukan secara beriringan dengan proses analisis, dan modifikasi/penyesuaian setiap rancangan dibuat berdasarkan wawasan baru yang didapatkan dari waktu ke waktu pada proses analisis. Hasil rancangan yang dipaparkan pada bab ini adalah rancangan paling akhir yang diimplementasi pada aplikasi Jitsi.

## Perancangan pada Jitsi Videobridge (JVB)

Perancangan pada komponen JVB bertujuan untuk menerapkan metode *pagination* dari sisi *backend* aplikasi Jitsi. Adapun, klien telah mengimplementasi metode ini pada *web front end* Jitsi Meet dan tim PT. Padepokan Tujuh Sembilan melanjutkan pencapaian tersebut dengan membuat modifikasi di sisi *backend*.

Perancangan yang dilakukan pada komponen JVB untuk memenuhi kebutuhan tim adalah sebagai berikut.

### Pemberhentian Pengiriman Paket ke Partisipan (Non-Octo)

Proses perancangan metode pemberhentian pengiriman paket ke partisipan (non-octo) dibagi ke dalam dua tahap. Berikut adalah uraian untuk kedua tahapan proses perancangan tersebut.

1. Modifikasi pada Objek Endpoint di JVB

Dari hasil analisis alur pengiriman paket di JVB, tidak ditemukan *method* yang bertujuan agar partisipan dapat memilih untuk menerima/menolak paket dari JVB. Modikasi yang dirancang pada objek Endpoint bertujuan untuk hal ini.

Modifikasi yang dibuat pada objek Endpoint yaitu penambahan dua buah variabel *list* untuk menampung ID partisipan *remote* yang paketnya (audio atau video) ingin diterima partisipan lokal. Adapun, variabel yang dibuat, yaitu:

* 1. wantToReceiveAudioFromEndPoints, dan
  2. wantToReceiveVideoFromEndPoints.

Modifikasi lainnya yaitu penambahan beberapa *method* untuk menangani pengiriman paket dari partisipan *remote*. *Method* yang dibuat, yaitu:

* 1. wantsPacketFromEndpoint() berfungsi untuk mengecek keinginan partisipan lokal untuk menerima paket dari partisipan *remote*. Rancangan fungsi ini diuraikan pada Tabel V.1.
  2. addEndpointToAcceptedList() dipanggil saat partisipan lokal ingin menerima paket dari partisipan *remote* tertentu. ID partisipan *remote* tersebut akan dimasukkan ke dalam *list* wantToReceive-AudioFromEndPoints atau wantToReceiveVideoFromEnd-Points sesuai jenis paket yang diinginkan partisipan. Rancangan fungsi ini diuraikan pada Tabel V.2.
  3. removeEndpointToAcceptedList() dipanggil saat partisipan lokal ingin berhenti menerima paket dari partisipan *remote* tertentu. ID partisipan *remote* tersebut akan dihapus dari *list* wantToReceiveAudioFromEndPoints atau wantToReceive-VideoFromEndPoints sesuai jenis paket yang ditolak partisipan. Rancangan fungsi ini diuraikan pada Tabel V.3.
  4. clearWantsToRecieveFromEndPoints() berfungsi untuk mengosongkan *list* wantToReceiveFromEndPoints dari semua ID partisi-pan *remote*. Rancangan fungsi ini diuraikan pada Tabel V.4.

Tabel V.1. Rancangan *Method* wantsPacketFromEndpoint()

|  |  |  |
| --- | --- | --- |
| Nama | wantsPacketFromEndpoint | |
| Tujuan | Fungsi untuk mengecek keinginan partisipan lokal untuk menerima paket dari partisipan *remote*. | |
| Input | * endpointID : * packetType : | ID partisipan *remote*  jenis paket yang diinginkan |
| Output | Keluaran boolean yang menyatakan keinginkan partisipan lokal untuk meneri-ma/menolak data dari partisipan remote *tersebut*. | |
| Proses | 1. Mengecek nilai ID partisipan *remote* apakah seduah terdapat di dalam *list* wantTo-ReceiveFromEndPoints. 2. Jika ID partisipan *remote* ditemukan di dalam *list*, *output* bernilai true, dan sebalik-nya, tidak ditemukan, *output* bernilai false. | |

Tabel V.2. Rancangan *Method* addEndpointToAcceptedList()

|  |  |  |
| --- | --- | --- |
| Nama | addEndpointToAcceptedList | |
| Tujuan | Fungsi dipanggil saat partisipan lokal ingin menerima paket dari partisipan *remote* tertentu. | |
| Input | * acceptedEndPoint : * packetType : | ID partisipan *remote* yang paketnya akan ditolak  jenis paket yang akan dierima |
| Output | ID partisipan *remote* dihapus dari daftar ID partisipan *remote* yang paketnya akan diterima. | |
| Proses | 1. Memasukkan ID partisipan *remote* ke dalam *list* wantToReceiveFromEnd-Points. | |

Tabel V.3. Rancangan *Method* removeEndpointFromAcceptedList()

|  |  |  |
| --- | --- | --- |
| Nama | removeEndpointFromAcceptedList | |
| Tujuan | Fungsi dipanggil saat partisipan lokal ingin menerima paket dari partisipan *remote* tertentu. | |
| Input | * removeEndPoint : * packetType : | ID partisipan *remote* yang paketnya akan ditolak  jenis paket yang akan ditolak |
| Output | ID partisipan *remote* dihapus dari daftar ID partisipan *remote* yang paketnya akan diterima. | |
| Proses | 1. Menghapus ID partisipan *remote* dari *list* wantToReceiveFromEndPoints. | |

Tabel V.4. Rancangan *Method* clearWantsToRecieveFromEndPoints()

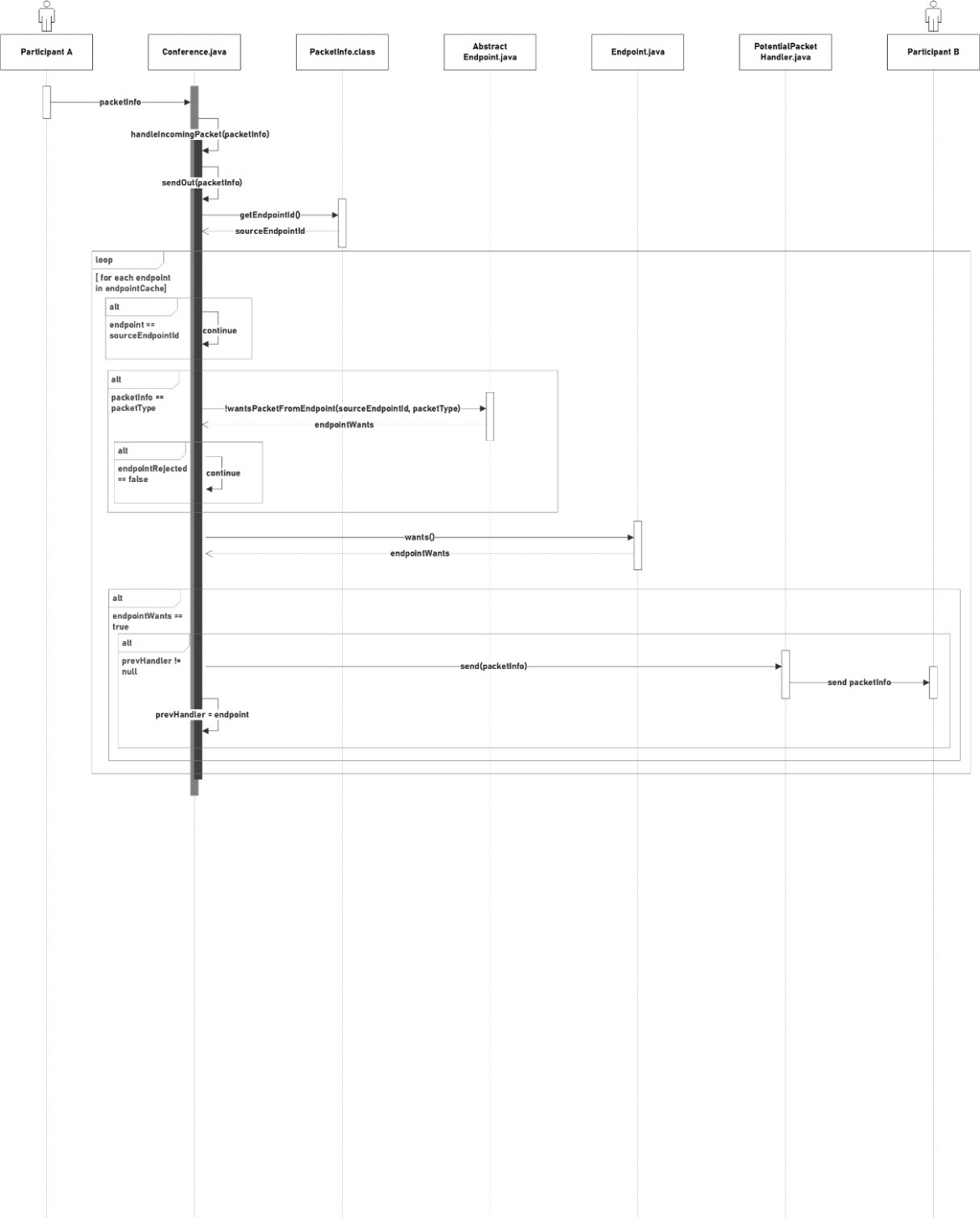
|  |  |  |
| --- | --- | --- |
| Nama | clearWantsToRecieveFromEndPoints | |
| Tujuan | Fungsi untuk mengosongkan *list* wantToReceiveFromEndPoints dari semua ID partisipan *remote* | |
| Input | * packetType : | jenis paket yang akan ditolak |
| Output | Semua ID partisipan remote terhapus dari *list* wantToReceiveFromEndPoints | |
| Proses | 1. Menghapus semua ID partisipan *remote* dari *list* wantToReceiveFromEnd-Points. | |

1. Modifikasi pada Alur Pengiriman Paket

Modifikasi yang dibuat pada alur pengiriman paket di JVB, tepatnya pada *method* sendOut(). Modifikasi yang dibuat yaitu pemanggilan *method* wantsPacketFromEndpoint() sebelum pemanaggilan *method* wants(). Hal ini bertujuan untuk beberapa hal, antara lain:

* 1. mengubah alur pengiriman paket sehingga kondisi awal konferensi video, yaitu partisipan lokal tidak menerima paket audio dan paket video apapun dari partisipan *remote*,
  2. mencegah pengiriman paket dari partisipan *remote* sebelum parti-sipan lokal memilih untuk menerima/menolak paket dari partisi-pan *remote* tersebut.

*Sequence diagram* (alur) pengiriman paket di JVB setelah dimodifikasi ditunjukkan pada Gambar V.1.



Gambar V.1. Alur Pengiriman Paket di JVB Setelah Dimodifikasi

### Fungsi RESTful API

Setelah dibuat modifikasi pada alur pengiriman paket, selanjutnya dibuat sejumlah fungsi RESTful API untuk proses *pagination* di JVB. Fungsi RESTful API yang dirancang, antara lain:

1. removeWantedRemoteAudioStream() berfungsi untuk menghentikan pengiriman paket audio dari partisipan remote. Rancangan fungsi ini diuraikan pada Tabel V.5.
2. removeWantedRemoteVideoStream() berfungsi untuk memberhentikan penerimaan paket video dari partisipan remote. Rancangan fungsi ini diuraikan pada Tabel V.6.
3. unmuteAllRemoteAudioStream() befungsi untuk *unmute* semua partisipan *remote*. Fungsi ini dibuat setelah pelaksanaan PKL selesai. Adapun, fungsi dibuat atas permintaan klien untuk mengkaktifkan semua audio partisipan tanpa melalui fungsi *paging*. Rancangan fungsi ini diuraikan pada Tabel V.7.

Tabel V.5. Rancangan Fungsi removeWantedRemoteAudioStream()

|  |  |  |
| --- | --- | --- |
| Nama | removeWantedRemoteAudioStream | |
| Tujuan | Fungsi untuk menghentikan pengiriman paket audio dari partisipan *remote*. | |
| Input | * localEndPointID : * remoteEndPointID : | ID partisipan lokal  ID partisipan *remote* yang paketnya akan ditolak |
| Output | ID partisipan *remote* dihapus dari daftar ID partisipan *remote* yang paketnya akan diterima. | |
| Proses | 1. Mengambil data ID konferensi berdasarkan *input* ID partisipan lokal de-ngan menggunakan *method* getConferenceId(). 2. Mengambil data partisipan lokal berdasarkan *input* ID partisipan lokal. 3. Mengambil data partisipan *remote* berdasarkan *input* ID partisipan *remote.* 4. Menghapus ID partisipan *remote* dari *list* wantToReceiveFromEndPoints dengan memanggil *method* removeEndpointFromAcceptedList dengan *packetType* bertipe audio. | |

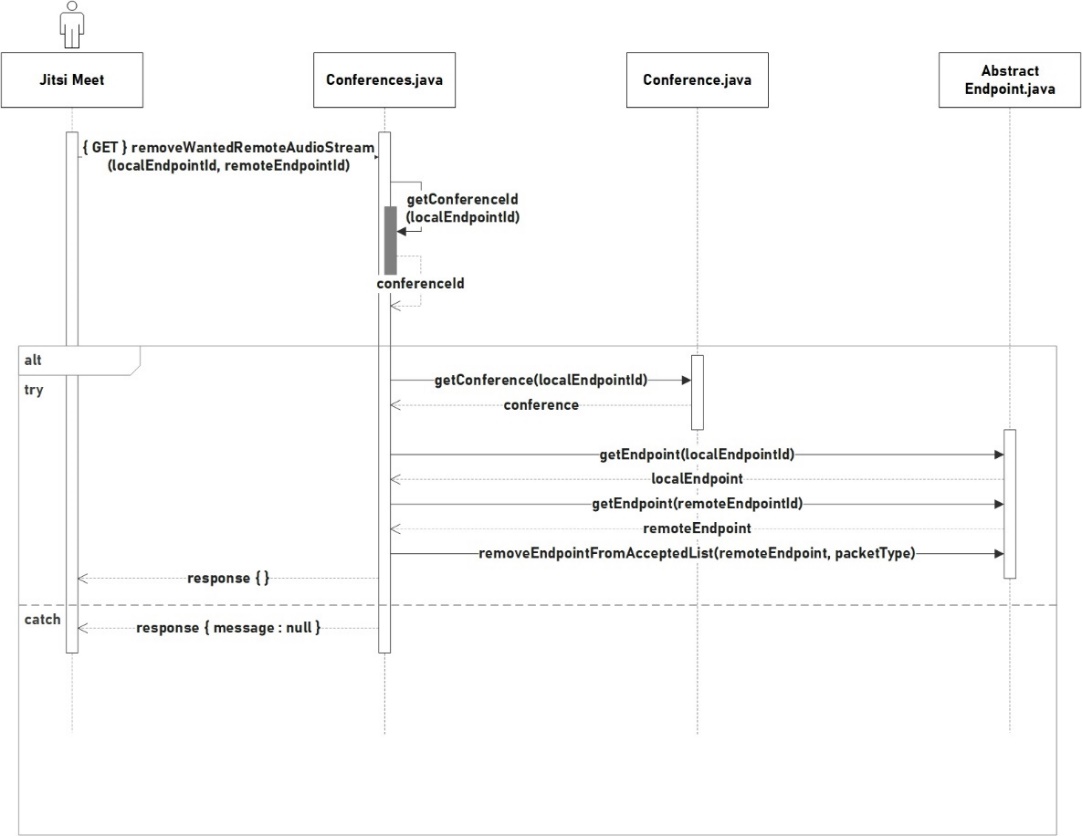
Tabel V.6. Rancangan Fungsi removeWantedRemoteVideoStream()

|  |  |  |
| --- | --- | --- |
| Nama | removeWantedRemoteVideoStream | |
| Tujuan | Fungsi untuk memberhentikan penerimaan paket video dari partisipan *remote*. | |
| Input | * localEndPointID : * remoteEndPointID : | ID partisipan lokal  ID partisipan *remote* yang paketnya akan ditolak |
| Output | ID partisipan *remote* dihapus dari daftar ID partisipan *remote* yang paketnya akan diterima. | |
| Proses | 1. Mengambil data ID konferensi berdasarkan *input* ID partisipan lokal de-ngan menggunakan *method* getConferenceId(). 2. Mengambil data partisipan lokal berdasarkan *input* ID partisipan lokal. 3. Mengambil data partisipan *remote* berdasarkan *input* ID partisipan *remote.*   Menghapus ID partisipan *remote* dari *list* wantToReceiveFromEndPoints dengan memanggil *method* removeEndpointFromAcceptedList dengan *packetType* bertipe audio. | |

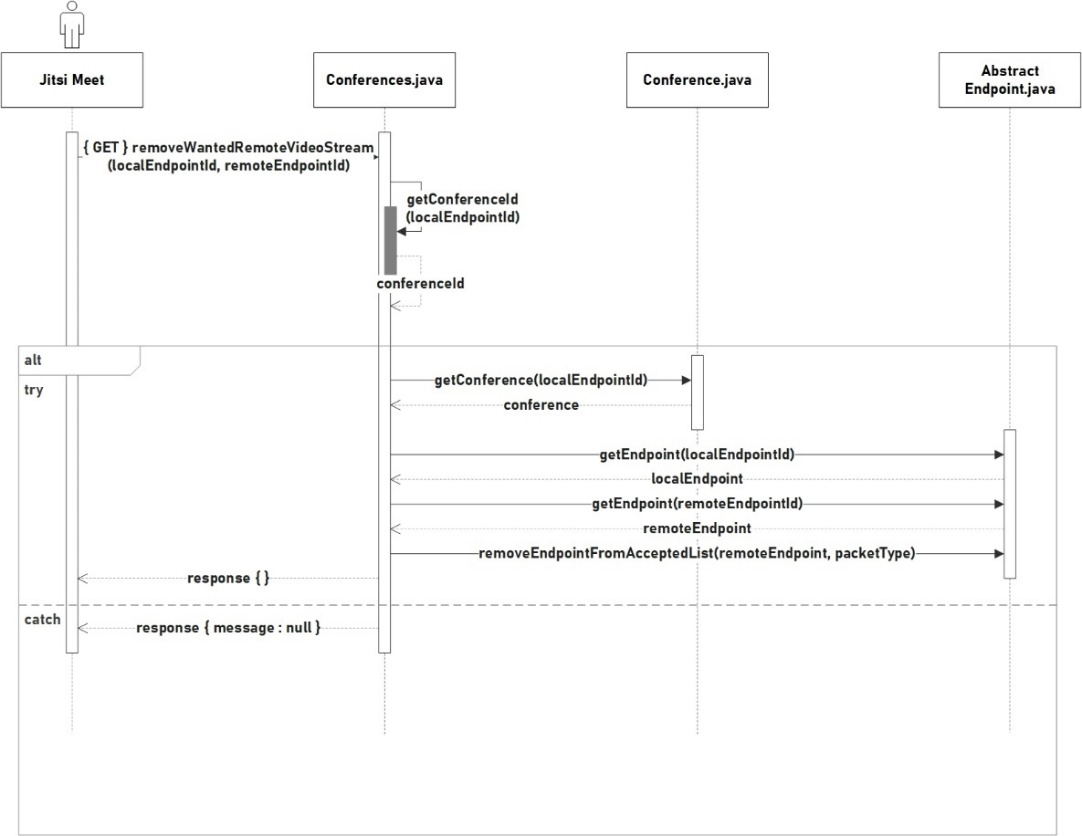
Tabel V.7. Rancangan Fungsi unmuteAllRemoteAudioStream()

|  |  |  |
| --- | --- | --- |
| Nama | unmuteAllRemoteAudioStream | |
| Tujuan | Fungsi untuk *unmute* semua partisipan *remote*. | |
| Input | localEndPointID : | ID partisipan lokal |
| Output | Partisipan lokal akan menerima pengiriman paket audio dari semua partisipan *remote* di ruang konferensi daring. | |
| Proses | 1. Mengambil data ID konferensi berdasarkan ID partisipan lokal dengan menggunakan *method* getConferenceId(). 2. Menghapus semua ID partisipan *remote* dari *list* wantToReceiveFromEnd-Points dengan memanggil *method* r clearWantsToRecieveFromEndPoints pada objek Endpoint dengan *packetType* bertipe audio. | |

Selanjutnya, pada Gambar V.2. dan V.3 ditunjukkan *sqeuence sequence* untuk alur fungsi removeWantedRemoteAudioStream() dan removeWantedRemoteVideo-Stream().



Gambar V.2. Alur fungsi removeWantedRemoteAudioStream()



Gambar V.3. Alur fungsi removeWantedRemoteVideoStream()

### Pengambilan Data ID Konferensi

Dari hasil analisis metode pemanggilan API dengan menggunakan AJAX, ditarik kesimpulan bahwa ID universal sebuah ruang konferensi (ID COLIBRI) tidak dapat diambil dari Jitsi Meet. Hal ini karena ID konferensi pada Jitsi Meet berbeda dengan ID konferensi yang digunakan komunikasi antar komponen penyusun aplikasi Jitsi. Sehubungan dengan ini, dibuat sebuah fungsi pada komponen JVB dengan tujuan tersebut bernama getConferenceID(). Rancangan fungsi ini diuraikan pada Tabel V.10.

Tabel V.8. Fungsi getConferenceID()

|  |  |  |
| --- | --- | --- |
| Nama | unmuteAllRemoteAudioStream | |
| Tujuan | Fungsi untuk mendapatkan ID konferensi di JVB (ID COLIBRI). | |
| Input | * localEndPointID : | ID partisipan lokal |
| Output | ID konferensi universal (ID COLIBRI) | |
| Proses | 1. Mengambil data semua konferensi yang sedang berlangsung di server JVB. 2. Melakukan pengecekan ID konferensi yang memiliki partisipan lokal sama dengan ID partisipan lokal yang di-*input*-kan. 3. Jika didapatkan konferensi dengan ID partisipan yang sesuai, ID konferensi diambil. | |

## Perancangan pada Jitsi Conference Focus (Jicofo)

Pada permulaan proses analisis komponen Jicofo, tim diminta untuk membuat dua buah fungsi RESTful API untuk menunjang proses analisis. Adapun, dua fungsi yang ditambahkan adalah sebagai berikut.

1. getConferences() berfungsi untuk mendapatkan semua konferensi video yang tersambung dengan Jicofo. Rancangan fungsi ini diuraikan pada Tabel V.9.
2. getBridgesParticipants() berfungsi untuk mendapatkan semua partisipan non-octo dan octo di suatu ruang konferensi video. Rancangan fungsi ini diuraikan pada Tabel V.10.

Tabel V.9. Rancangan Fungsi getConferences()

|  |  |  |
| --- | --- | --- |
| Nama | getConferences | |
| Tujuan | Fungsi untuk mendapatkan semua konferensi video yang tersambung dengan Jicofo. | |
| Input | - |  |
| Output | JSON berisi ID konferensi yang tersambung dengan Jicofo. | |
| Proses | 1. Mengambil data semua konferensi yang tersambung ke Jicofo. 2. Mengambil data *bridge* dari semua konferensi. 3. Mengambil ID semua konferensi. | |

Tabel V.10. Rancangan Fungsi getBridgesParticipants()

|  |  |  |
| --- | --- | --- |
| Nama | getBridgesParticipants | |
| Tujuan | Fungsi untuk mendapatkan semua partisipan non-octo dan octo di ruang konferensi video. | |
| Input | * conferenceId : | ID konferensi |
| Output | ID konferensi universal (ID COLIBRI) | |
| Proses | 1. Mengambil data semua konferensi yang tersambung ke Jicofo. 2. Mengambil data *bridge* dari semua konferensi. 3. Jika ID konferensi pada *bridgeSession* sama dengan ID konferensi yang di-input-kan, mengambil ID semua konferensi. | |

## Perancangan pada Jitsi Meet

Modifikasi pada komponen Jitsi bertujuan untuk mensimulasikan proses *paging* yang diterapkan pada komponen JVB. Adapun, pada awalnya fungsi pemanggilan API yang dibuat pada komponen Jitsi Meet dimaksudkan untuk memanggil fungsi getRemoteEndpointPage() – yang merupakan fungsi untuk menerapkan *pagination* di sisi *backend* aplikasi Jitsi. Pada pengerjaan proyek ini, penulis tidak terlibat pada perancangan fungsi tersebut.

Tabel V.11. Rancangan Fungsi getRemoteEndpointPage()

|  |  |  |
| --- | --- | --- |
| Nama | getRemoteEndpointPerPage | |
| Tujuan | Fungsi ini berfungsi untuk menerapkan *pagination* di sisi *backend* aplikasi Jitsi. | |
| Input | * conferenceId : * pageNumber : * pageData : | ID konferensi  jumlah halaman di ruang konferensi  jumlah video yang akan tampil pada satu laman di perangkat partisipan lokal |
| Ouput | *Paging* berhasil dilakukan dimana paket dari *remote participant* sebanyak *page data* berhasil diterima. | |

Tetapi, setelah dilakukan analisis untuk meyakinkan bahwa urutan partisipan yang di-*mute* di JVB sesuai dengan urutan tampilan video pada *web frontend* Jitsi Meet (analisis untuk mendapatkan urutan partisipan yang di-*mute* berdasarkan waktu join konferensi), proses *pagination* pada Jitsi mengalami perubahan. Perubahan yang terjadi yaitu penerapan *pagination* utamanya dilakukan di sisi *web front end* Jitsi Meet, dan JVB hanya menunjang proses tersebut dengan menyediakan beberapa fungsi, antara lain:

1. addWantedRemoteAudioStream(),
2. addWantedRemoteVideoStream(),
3. removeWantedRemoteAudioStream(), dan
4. removeWantedRemoteVideoStream().

Hasil rancangan yang akan dijabarkan adalah hasil rancangan setelah analisis untuk mendapatkan urutan partisipan yang di-*mute* berdasarkan waktu join konferensi.

Modifikasi yang dibuat pada komponen Jitsi Meet yaitu penambahan variabel *list* untuk menampung ID partisipan *remote* yang paketnya ingin diterima partisipan lokal, berdasarkan urutan waktu *join* partisipan tersebut ke dalam ruang konferensi. Adapun, variabel ini bernama idList. Variabel lain yang ditambahkan ada variabel bertipe integer untuk mengatur laman (*page*) yang aktif, yaitu variabel bernama pageNumber.

Kemudian, dibuat modifikasi penambahan beberapa fungsi yang bertujuan untuk pemanggilan API dan beberapa fungsi untuk penanganan *unser interface* (UI) *web frontend* Jitsi Meet pada proses *pagination*. Adapun, beberapa fungsi yang dibuat, yaitu:

1. Fungsi pemanggilan API
   1. unmuteParticipant() berfungsi untuk melakukan *unmute* sejumlah parti-sipan remote sesuai jumlah nilai data yang ditentukan pengguna. Ranca-ngan fungsi ini diuraikan pada Tabel V.12.
   2. muteParticipant() berfungsi untuk melakukan *mute* sejumlah partisipan remote sesuai jumlah nilai data yang ditentukan pengguna. Rancangan fungsi ini diuraikan pada Tabel V.13.
   3. muteAllParticipant() berfungsi untuk melakukan *mute* semua partisipan remote sesuai jumlah nilai data yang ditentukan pengguna. Rancangan fungsi ini diuraikan pada Tabel V.14.
2. Fungsi untuk penanganan UI
   1. addParticipantId() berfungsi untuk memasukkan ID partisipan ke dalam idList secara terurut sesuai waktu *join* ke ruang konferensi daring. Rancangan fungsi ini diuraikan pada Tabel V.15.
   2. removeParticipantId() berfungsi untuk menghapus ID partisipan dari idList ketika partisipan *left* dari ruang konferensi daring. Rancangan fungsi ini diuraikan pada Tabel V.16.

Tabel V.12. Rancangan Fungsi unmuteParticipant()

|  |  |  |
| --- | --- | --- |
| Nama | unmuteParticipant | |
| Tujuan | Fungsi untuk melakukan *unmute* sejumlah parti-sipan remote sesuai jumlah nilai data yang ditentukan pengguna | |
| Input | * localParticipant : * totalParticipant : * pageDataNumber : * pageData : * baseUrlVideo : * baseUrlAudio : | ID partisipan lokal  jumlah total partisipan di ruang konferensi daring  jumlah halaman di ruang konferensi  jumlah video yang akan tampil pada satu halaman di perangkat partisipan lokal  alamat untuk mengakses API untuk *unmute* video partisipan *remote*  alamat untuk mengakses API untuk *unmute* video partisipan *remote* |
| Output | Partisipan lokal akan menerima tampilan video partisipan sejumlah nilai page-Data yang ditetapkan. | |
| Proses | 1. Mengambil ID partisipan lokal. 2. Mengambil jumlah semua partisipan pada sebuah ruang konferensi. 3. Memberhentikan pengiriman video dari partisipan *remote* pada laman (*page*) sebelumnya. 4. Memanggil fungsi addWantedRemoteVideoStream agar partisipan lokal dapat menerima pengiriman video dari partisipan remote pada laman yang sedang terbuka. | |

Tabel V.13. Rancangan Fungsi muteParticipant()

|  |  |  |
| --- | --- | --- |
| Nama | muteParticipant | |
| Tujuan | Fungsi untuk melakukan *mute* sejumlah partisipan remote sesuai jumlah nilai data yang ditentukan pengguna. | |
| Input | * localParticipant : * totalParticipant : * pageDataNumber : * pageData :      * baseUrlVideo : * baseUrlAudio : | ID partisipan lokal  jumlah total partisipan di ruang konferensi daring  jumlah halaman di ruang konferensi  jumlah video yang akan tampil pada satu halaman di perangkat partisipan lokal  alamat untuk mengakses API untuk *unmute* video partisipan *remote*  alamat untuk mengakses API untuk *unmute* video partisipan *remote* |
| Output | Menghentikan pengriman paket video partisipan *remote* sejumlah nilai page-Data yang ditetapkan. | |
| Proses | 1. Mengambil ID partisipan lokal. 2. Memanggil fungsi removeWantedRemoteVideoStream untuk menghenti-kan pengiriman video dari partisipan *remote* di laman yang sedang terbuka. | |

Tabel V.14. Rancangan Fungsi muteAllParticipant()

|  |  |  |
| --- | --- | --- |
| Nama | muteAllParticipant | |
| Tujuan | Fungsi untuk melakukan *mute* semua partisipan remote sesuai jumlah nilai data yang ditentukan pengguna. | |
| Input | - |  |
| Output | Fungsi *pagination* yang diimplementasi di JVB akan dipanggil di Jitsi Meet dan meampilkan video partisipan remote di perangkat partisipan lokal sesuai jumlah pageDataData yang ditentukan partisipan lokal. | |
| Proses | 1. Mengambil ID partisipan lokal. 2. Memanggil fungsi removeWantedRemoteVideoStream untuk menghenti-kan pengiriman video dari semua partisipan *remote.* | |

Tabel V.15. Rancangan Fungsi addParticipantId()

|  |  |  |
| --- | --- | --- |
| Nama | addParticipantId | |
| Tujuan | Fungsi untuk memasukkan ID partisipan ke dalam idList secara terurut sesuai waktu *join* ke ruang konferensi daring. | |
| Input | * id : | ID partisipan *remote* |
| Output | ID partisipan masuk ke dalam *list* idList | |

Tabel V.16. Rancangan Fungsi removeParticipantId()

|  |  |  |
| --- | --- | --- |
| Nama | removeParticipantId | |
| Tujuan | Fungsi untuk menghapus ID partisipan dari idList ketika partisipan *left* dari ruang konferensi daring. | |
| Input | * id : | ID partisipan *remote* |
| Output | ID partisipan dihapus dari dalam *list* idList | |

# IMPLEMENTASI

Bab ini berisi penjabaran hasil implementasi yang dilakukan – untuk penambahan fungsi dan modifikasi pada program – selama pengerjaan proyek *Jitsi Bandwidth Reduction*. Adapun hasil impelementasi yang dipaparkan pada bab ini adalah segala bentuk modifikasi pada komponen Jitsi, termasuk pembuatan fungsi dan penam-bahan log pada program untuk kebutuhan proyek.

## Implementasi di Jitsi Videobridge (JVB)

Implementasi pada komponen JVB dialkuakn berdasarkan rancangan yang telah dibuat pada komponen ini untuk penerapan metode *pagination* dari sisi *backend* aplikasi Jitsi.

Implementasi yang dilakukan pada komponen JVB untuk memenuhi kebutuhan tim berdasarkan rancangan untuk pemberhentian pengiriman paket ke partisipan non-octo dan implementasi fungsi RESTful API.

### Pemberhentian Pengiriman Paket ke Partisipan (Non-Octo)

Implementasi metode pemberhentian pengiriman paket ke partisipan (non-octo) dibagi ke dalam dua tahap beritkut.

1. Modifikasi pada Objek *Endpoint* di JVB

Implementasi modifikasi pada objek *Endpoint* pertama-tama adalah pe-nambahan dua buah variabel bertipe *map*, wantToReceiveAudioFromEnd-Points dan wantToReceiveVideoFromEndPoints untuk menampung ID partisipan *remote* yang paketnya ingin diterima partisipan lokal.

Kemudian, diimpplementasi fungsi RESTful API berdasarkan rancangan, yaitu:

* 1. wantsPacketFromEndpoint() diuraikan pada Tabel VI.1.
  2. addEndpointToAcceptedList() diuraikan pada Tabel VI.2.
  3. removeEndpointToAcceptedList() diuraikan pada Tabel VI.3.
  4. clearWantsToRecieveFromEndPoints() diuraikan pada Tabel VI.4.

Tabel VI.1. Fungsi wantsPacketFromEndpoint()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/AbstractEndpoint.java |
| *Source code* | public boolean wantsPacketFromEndpoint(String  endpointID, String packetType)  {  if (packetType.equals("Video") && this.  wantToReceiveVideoFromEndPoints.containsKey(  endpointID))  {  return true;  }  if (packetType.equals("Audio") && this.  wantToReceiveAudioFromEndPoints.containsKey(  endpointID))  {  return true;  }  return false;  } |

Tabel VI.2.Fungsi addEndpointToAcceptedList()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/AbstractEndpoint.java |
| *Source code* | public void addEndpointToAcceptedList(AbstractEndpoint rejectedEndPoint, String packetType)  {  if (packetType.equals("Video"))  {  this.wantToReceiveVideoFromEndPoints.put(  rejectedEndPoint.getID(),rejectedEndPoint  ;  }  if (packetType.equals("Audio"))  {  this.wantToReceiveAudioFromEndPoints.put(  rejectedEndPoint.getID(), rejectedEndPoint)  ;  }  }  } |

Tabel VI.3. Fungsi removeEndpointToAcceptedList()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/AbstractEndpoint.java |
| *Source code* | public void removeEndpointFromAcceptedList(  AbstractEndpoint removeEndPoint, String packetType)  {  if (packetType.equals("Video"))  {  this.wantToReceiveVideoFromEndPoints.remove(  removeEndPoint.getID());  }  if (packetType.equals("Audio"))  {  this.wantToReceiveAudioFromEndPoints.remove(  removeEndPoint.getID());  }  } |

Tabel VI.4. Fungsi clearWantsToRecieveFromEndPoints()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/AbstractEndpoint.java |
| *Source code* | public void clearWantsToRecieveFromEndPoints(String packetType)  {  if (packetType.equals("Video"))  {  this.wantToReceiveVideoFromEndPoints.clear();  }  if (packetType.equals("Audio"))  {  this.wantToReceiveAudioFromEndPoints.clear();  }  } |

1. Modifikasi Alur Pengiriman Paket

Implemetnasi modifikasi yang dibuat pada alur pengiriman paket di JVB yaitu penambahan baris untuk pemanggilan *method* wantsPacketFrom-Endpoint() sebelum pemanaggilan *method* wants(). Fungsi ini dirauikan pada Tabel VI.5.

Tabel VI.5. Fungsi sendOut() dengan Modifikasi

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/Conference.java |
| *Source code* | private void sendOut(PacketInfo packetInfo)  {  String sourceEndpointId = packetInfo.  getEndpointId();    PotentialPacketHandler prevHandler = null;  for (Endpoint endpoint : endpointsCache)  {  if (endpoint.getID().equals(sourceEndpointId))  {  continue;  }  **// pemanggilan fungsi wantsPacketFromEndpoint()**  **//** **untuk mengecek keinginan partisipan untuk**  **// menerima paket**  if (endpoint.isPacket(packetInfo, "Video"))  {  if (!endpoint.wantsPacketFromEndpoint(  sourceEndpointId, "Video"))  {  continue;  }  }  if (endpoint.isPacket(packetInfo, "Audio"))  {  if (!endpoint.wantsPacketFromEndpoint(  sourceEndpointId, "Audio"))  {  continue;  }  }  if (endpoint.wants(packetInfo))  {  if (prevHandler != null)  {  prevHandler.send(packetInfo.clone());  }  prevHandler = endpoint;  }  }  if (tentacle != null && tentacle.wants(packetInfo))  {  if (prevHandler != null)  {  prevHandler.send(packetInfo.clone());  }  prevHandler = tentacle;  }  if (prevHandler != null)  {  prevHandler.send(packetInfo);  }  else  }  ByteBufferPool.returnBuffer(packetInfo.  getPacket().getBuffer());  }  } |

### Fungsi RESTful API

Setelah implementasi modifikasi pada alur pengiriman paket, selanjutnya adalah implementasi fungsi RESTful API untuk proses *pagination* di JVB. Pada penger-jaan proyek ini, fungsi RESTful API yang diimplementasi, antara lain:

1. removeWantedRemoteAudioStream() diuraikan pada Tabel VI.6.
2. removeWantedRemoteVideoStream() diuraikan pada Tabel VI.7.

Adapun fungsi removeWantedRemoteAudioStream() dan removeWantedRemote-VideoStream() memiliki alur proses yang serupa, namun dengan *input* *packetType* yang berbeda, tergantung kebutuhan pengguna untuk menerima pengiriman paket audio atau video.

Tabel VI.6. Fungsi removeWantedRemoteAudioStream()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/rest/root/colibri/conferences/  Conferences.java |
| *Source code* | @GET  @Path("/removeWantedRemoteAudioStream/{localEndPointID}/{remoteEndPointID}")  @Produces(MediaType.APPLICATION\_JSON)  public String removeWantedRemoteVideoStream(  @PathParam("localEndPointID") String  localEndPointID,  @PathParam("remoteEndPointID") String  remoteEndPointID)  {  JSONObject response = new JSONObject();  String conferenceId = getConferenceId(  localEndPointID);  try  {  Conference conference =  videobridgeProvider.get().  getConference(conferenceId);  AbstractEndpoint localEndpoint =  conference.getEndpoint(localEndPointID);  AbstractEndpoint remoteEndpoint =  conference.getEndpoint(remoteEndPointID);    localEndpoint.removeEndpointFromAcceptedList  (remoteEndpoint, "Audio");  } catch (Exception ex) {  response.put("Message", ex.getLocalizedMessage  ());  ex.printStackTrace();  }  return response.toJSONString();  } |

Tabel VI.7. Fungsi removeWantedRemoteVideoStream()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/rest/root/colibri/conferences/  Conferences.java |
| *Source code* | @GET  @Path("/removeWantedRemoteVideoStream/{localEndPointID}/{remoteEndPointID}")  @Produces(MediaType.APPLICATION\_JSON)  public String removeWantedRemoteVideoStream(  @PathParam("localEndPointID") String  localEndPointID,  @PathParam("remoteEndPointID") String  remoteEndPointID)  {  JSONObject response = new JSONObject();  String conferenceId = getConferenceId(  localEndPointID);  try  {  Conference conference =  videobridgeProvider.get().  getConference(conferenceId);  AbstractEndpoint localEndpoint =  conference.getEndpoint(localEndPointID);  AbstractEndpoint remoteEndpoint =  conference.getEndpoint(remoteEndPointID);    localEndpoint.removeEndpointFromAcceptedList  (remoteEndpoint, "Video");  } catch (Exception ex) {  response.put("Message", ex.getLocalizedMessage  ());  ex.printStackTrace();  }  return response.toJSONString();  } |

Untuk proses *pagination* di sisi *backend* aplikasi Jitsi, adapun fungsi utama yang dibuat adalah fungsi getRemoteEndpointPerPage() yang diuraikan pada Tabel VI.8.

Tabel VI.8. Fungsi getRemoteEndpointPerPage()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/rest/root/colibri/conferences/  Conferences.java |
| *Source code* | @ GET  @Path("/{localEndpointId}/{pageNumber}/{pageData}")  @Produces(MediaType.APPLICATION\_JSON)  public String getRemoteEndpointPerPage(  @PathParam("localEndpointId") String  localEndpointId,  @PathParam("pageNumber") int pageNumber,  @PathParam("pageData") int pageData)  {  JSONObject endpointsPerPageJSON = new JSONObject();  try {  String conferenceId = getConferenceId(  localEndpointId);  Conference conference = videobridgeProvider.  get().getConference(conferenceId);    if ((pageNumber > 0) && (pageData > 0))  {  List<AbstractEndpoint> endpointsPerPage =  conference.getEndpoints().  stream().  filter(endpoint -> !endpoint.getID().  equals(localEndpointId)).  skip(pageData\*(pageNumber-1)).  limit(pageData).  collect(Collectors.toList());  AbstractEndpoint localEndpoint = conference  .getEndpoint(localEndpointId);    localEndpoint.clearWantsToRecieveFromEnd  Points("Video");  localEndpoint.clearWantsToRecieveFromEnd  Points("Audio");  for(AbstractEndpoint endpoint :  endpointsPerPage)  {  localEndpoint.addEndpointToAccepted  List(endpoint, "Video");  localEndpoint.addEndpointToAccepted  List(endpoint, "Audio");  endpointsPerPageJSON.put(endpoint.  getID(),endpoint.  getDisplayName());  }  } else {  endpointsPerPageJSON.put("Message",  "The value of page number and page data  must be greater than 0");  }  } catch (Exception e) {  e.printStackTrace();  }  return endpointsPerPageJSON.toJSONString();  } |

### Pengambilan Data ID Konferensi

Berrdasarkan rancangan fungsi getConferenceId() yang diuraikan pada Tabel V.10, fungsi yang diimplementasi di JVB untuk mengambil ID universal sebuah ruang konferensi (ID COLIBRI) diuraikan pada Tabel VI.9.

Tabel VI.9. Fungsi getConferenceId()

|  |  |
| --- | --- |
| *Path* | jvb/src/main/java/org/jitsi/videobridge/rest/root/colibri/conferences/  Conferences.java |
| *Source code* | public String getConferenceId(String localParticipantId()  {  String conferences = getConferences();  conferences = conferences.replace("[", "");  conferences = conferences.replace("]", "");  conferences = conferences.replace("/"", "");  conferences = conferences.replace("{id:", "");  conferences = conferences.replace("}", "");  String[] conferenceIDs = conferences.split(",");  for(int i = 0; i < conferenceIDs.length; i++)  {  String conferenceID = conferenceIDs[i];  Conference conference =  videobridgeProvider.get().getConference(  conferenceID);  for (Endpoint endpoint : conference.  getLocalEndpoints())  {  if(endpoint.getID().equals(  localParticipantId))  {  return conferenceID;  }  }  }  return null;  } |

## Implementasi di Jitsi Conference Focus (Jicofo)

Untuk menunjang proses analisis komponen Jicofo, dua buah fungsi RESTful API yang diimplementasi, yaitu:

1. getConferences()diuraikan pada Tabel V.10.
2. getBridgesParticipants() diuraikan pada Tabel V.11. Adapun pada fungsi ini, *identifier* yang digunakan untuk partisipan octo adalah ID relay untuk setiap partisipan, berupa IP *address*. Hal ini belum sesuai dengan kebutuhan tim, namun tidak ditemukan variabel yang merupakan *identifier* untuk setiap partisipan octo.

Tabel VI.10. Fungsi getConferences()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/rest/Conferences.java |
| *Source code* | @GET  @Produces(MediaType.APPLICATION\_JSON)  public String getConferences()  {  JSONObject conferencesJSON = new JSONObject();  try {  List<JitsiMeetConference> conferences =  focusManagerProvider.get().getConferences();  for (JitsiMeetConference conf : conferences)  {  List<BridgeSession> bridges = conf.  getBridgesList();    for(BridgeSession bridge : bridges)  {  conferencesJSON.put(bridge.  colibriConference.getConferenceId(),  null);  }  }  } catch (Exception e) {  conferencesJSON.put("Message", e.getMessage());  e.printStackTrace();  }  return conferencesJSON.toJSONString();  } |

Tabel VI.11. Fungsi getBridgesParticipants()

|  |  |
| --- | --- |
| *Path* | src/main/java/org/jitsi/jicofo/rest/Conferences.java |
| *Source code* | @GET  @Path("/participants/{conferenceId}")  @Produces(MediaType.APPLICATION\_JSON)  public String getBridgesParticipants(  @PathParam("conferenceId") String conferenceId)  {  JSONObject participantsJSON = new JSONObject();  try {  List<JitsiMeetConference> conferences =  focusManagerProvider.get().getConferences();  List<Participant> participants =  new ArrayList<>();  List<String> participantsID =  new ArrayList<>();  OctoParticipant octoParticipant = null;  String bridgeConferenceId;  for (JitsiMeetConference conf : conferences)  {  List<BridgeSession> bridges =  conf.getBridgesList();  for (BridgeSession bridgeSession : bridges)  {  bridgeConferenceId = bridgeSession.  colibriConference.getConferenceId();    if (bridgeConferenceId.equals(  conferenceId))  {  participants =  bridgeSession.participants;    octoParticipant =  bridgeSession.octoParticipant;    }  }  }  if (participants != null)  {  for(Participant participant : participants)  {  participantsID.add(participant.  getEndpointId());  }  participantsJSON.put("Participants",  participantsID);  }  if (octoParticipant != null)  {  participantsJSON.put("Octo Participants",  octoParticipant.getRelays());  }  } catch (Exception e) {  participantsJSON.put("Message", e.getMessage()  );  e.printStackTrace();  }  return participantsJSON.toJSONString();  } |

## Implementasi di Jitsi Meet

Implementasi modifikasi yang pertama dibuat adalah penambahkan objek *button* pada *web front end* Jitsi Meet sehingga pengguna dapat mengakses fungsi *paging* di JVB dengan mudah. Penambahan elemen *button* pada fungsi render() pada fitur Filmstrip diuraikan pada Tabel VI.12.

Tabel VI.12. Modifikasi penambahan *button* di Jitsi Meet

|  |  |
| --- | --- |
| *Path* | react/features/filmstrip/components/web/Filmstrip.js |
| *Source code* | <button id='next' onClick={() => unmuteParticipant(1)}>  NEXT  </button>  <button id='prev' onClick={() => unmuteParticipant(-1)}>  PREV  </button>  <button id='unmute' onClick={unmuteAllParticipant}>  UNMUTE  </button>  <button id='mute' onClick={muteAllParticipant}>  MUTE  </button> |

Modifikasi yang selanjutnya dibuat yaitu penambahan variabel bertipe *list* pada komponen Jitsi Meet untuk menampung ID partisipan *remote* yang paketnya ingin diterima partisipan lokal, berdasarkan urutan waktu *join* partisipan tersebut ke dalam ruang konferensi dan variabel bertipe integer untuk mengatur laman (*page*) yang aktif. Kedua variabel ini ditambahkan di react/features/filmstrip/functions. web.js dengan nama idList dan pageNumber.

Kemudian, dilakukan implementasi penambahan fungsi sesuai rancangan yang telah dibuat untuk beberapa fungsi pemanggilan API dan fungsi penanganan *unser interface* (UI) *web frontend* Jitsi Meet pada proses *pagination*.

1. Fungsi pemanggilan API
   1. unmuteParticipant() diuraikan Tabel VI.13.
   2. muteParticipant() diuraikan pada Tabel V.14.
   3. muteAllParticipant() diuraikan pada Tabel V.15.
2. Fungsi untuk UI
   1. addParticipantId() diuraikan pada Tabel V.16.
   2. removeParticipantId() diuraikan pada Tabel V.17.

Fungsi addParticipantId() dipanggil di fungsi addRemoteParticipant-Container(), yaitu fungsi untuk membuat *layout* video untuk partisipan *remote*, sedangkan, fungsi removeParticipantId() dipanggil di fungsi remove-ParticipantContainer() apabila partisipan ingin berhenti menerima pengiriman data. Hasil implementasi modifikasi diuraikan pada Tabel VI.18 dan VI.19.

Adapun setelah dilakukan implementasi, monitor video partisipant remote tidak hilang – pergerakan pada video berhenti (*freezed*), namun tampilan video tidak hilang dari layar partisipan lokal – setelah eksekusi fungsi muteParticipant() dan muteAllParticipant() sehingga dibuat dua buah fungsi untuk memenuhi kebutuhan ini, yaitu: hideMutedRemoteVideo() dan showRemoteVideo() yang diuraikan pada Tabel VI.20 dan VI.21.

Tabel VI.13. Fungsi unmuteParticipant()

|  |  |
| --- | --- |
| *Path* | react/features/filmstrip/functions.web.js |
| *Source code* | export function unmuteParticipant(page) {  const localParticipant = getLocalParticipant(APP.  store.getState()).id;  const totalParticipant =  getParticipantCountWithFake(APP.store.  getState());  const pageData = 1;  if (((pageNumber + page) \* pageData <  totalParticipant) && (pageNumber + page > 0))  {  muteParticipant(pageNumber);  pageNumber += page;  const baseUrlVideo = "https://jitsi.cloudias79.  com:8081/colibri/conferences/addWantedRemo  teVideoStream/";  const baseUrlAudio = "https://jitsi.cloudias79.  com:8081/colibri/conferences/addWantedRemo  teAudioStream/";      for (let i = ((pageNumber - 1) \* pageData); i <  (pageData \* pageNumber); i++) {  const url = localParticipant + "/" +  idList[i];  const video = baseUrlVideo + url;  const audio = baseUrlAudio + url;    axios  .get(video)  .then(function(response) {  console.log(response.data);  console.log(video);  })  .catch(function(error) {  console.log(error);  });    axios  .get(audio)  .then(function(response) {  console.log(response.data);  console.log(audio);  })  .catch(function(error) {  console.log(error);  });    VideoLayout.showRemoteVideo(idList[i]);  }  }  } |

Tabel VI.14. Fungsi muteParticipant()

|  |  |
| --- | --- |
| *Path* | react/features/filmstrip/functions.web.js |
| *Source code* | export function unmuteParticipant(page) {  const localParticipant = getLocalParticipant(APP.  store.getState()).id;  const totalParticipant =  getParticipantCountWithFake(APP.store.  getState());  const pageData = 1;  if (((pageNumber + page) \* pageData <  totalParticipant) && (pageNumber + page > 0))  {  muteParticipant(pageNumber);  pageNumber += page;  const baseUrlVideo = "https://jitsi.cloudias79.  com:8081/colibri/conferences/addWantedRemo  teVideoStream/";  const baseUrlAudio = "https://jitsi.cloudias79.  com:8081/colibri/conferences/addWantedRemo  teAudioStream/";    for (let i = ((pageNumber - 1) \* pageData); i <  (pageData \* pageNumber); i++) {  const url = localParticipant + "/" +  idList[i];  const video = baseUrlVideo + url;  const audio = baseUrlAudio + url;    axios  .get(video)  .then(function(response) {  console.log(response.data);  console.log(video);  })  .catch(function(error) {  console.log(error);  });    axios  .get(audio)  .then(function(response) {  console.log(response.data);  console.log(audio);  })  .catch(function(error) {  console.log(error);  });    VideoLayout.showRemoteVideo(idList[i]);  }  }  } |

Tabel VI.15. Fungsi muteAllParticipant()

|  |  |
| --- | --- |
| *Path* | react/features/filmstrip/functions.web.js |
| *Source code* | export function muteAllParticipant() {  const localParticipant = getLocalParticipant(APP.  store.getState()).id;  const baseUrlVideo = "https://jitsi.cloudias79.com:  8081/colibri/conferences/  removeWantedRemoteVideoStream/";  const baseUrlAudio = "https://jitsi.cloudias79.com:  8081/colibri/conferences/  removeWantedRemoteAudioStream/";    for (let i = 0; i < getParticipantCountWithFake(  APP.store.getState()); i++)  {  const url = localParticipant + "/" + idList[i];  const video = baseUrlVideo + url;  const audio = baseUrlAudio + url;  axios  .get(video)  .then(function(response) {  console.log(response.data);  console.log(video);  })  .catch(function(error) {  console.log(error);  });  axios  .get(audio)  .then(function(response) {  console.log(response.data);  console.log(audio);  })  .catch(function(error) {  console.log(error);  });  VideoLayout.hideMutedRemoteVideo(idList[i]);  }  pageNumber = 0;  } |

Tabel VI.16. Fungsi addParticipantId()

|  |  |
| --- | --- |
| *Path* | react/features/filmstrip/functions.web.js |
| *Source code* | export function addParticipantId(id: number) {  idList.push(id);  } |

Tabel VI.17. Fungsi removeParticipantId()

|  |  |
| --- | --- |
| *Path* | react/features/filmstrip/functions.web.js |
| *Source code* | export function removeParticipantId(removedId: number) {  idList = idList.filter(id => id !== removedId)  } |

Tabel VI.18. Fungsi addRemoteParticipantContainer() dengan Modifikasi

|  |  |
| --- | --- |
| *Path* | modules/UI/videolayout/VideoLayout.js |
| *Source code* | addRemoteParticipantContainer(participant) {  if (!participant || participant.local) {  return;  } else if (participant.isFakeParticipant) {  const sharedVideoThumb = new SharedVideoThumb(  participant, SHARED\_VIDEO\_CONTAINER\_TYPE,  VideoLayout);  this.addRemoteVideoContainer(participant.id,  sharedVideoThumb);  return;  }  const id = participant.id;  const jitsiParticipant = APP.conference.  getParticipantById(id);  const remoteVideo = new RemoteVideo(  jitsiParticipant, VideoLayout);  **// pemanggilan fungsi addParticipantId() untuk**  **// memmasukkan ID partisipan remote ke idList**  addParticipantId(id);  this.\_setRemoteControlProperties(jitsiParticipant, remoteVideo);  this.addRemoteVideoContainer(id, remoteVideo);  this.updateMutedForNoTracks(id, 'audio');  this.updateMutedForNoTracks(id, 'video');  } |

Tabel VI.19. Fungsi addRemoteParticipantContainer() dengan Modifikasi

|  |  |
| --- | --- |
| *Path* | modules/UI/videolayout/VideoLayout.js |
| *Source code* | removeParticipantContainer(id) {  if (this.getPinnedId() === id) {  logger.info('Focused video owner has left the  conference');  APP.store.dispatch(pinParticipant(null));  }  const remoteVideo = remoteVideos[id];  if (remoteVideo) {  logger.info(`Removing remote video: ${id}`);  delete remoteVideos[id];  remoteVideo.remove();  } else {  logger.warn(`No remote video for ${id}`);  }  **// pemanggilan fungsi removeParticipantId() untuk**  **// menghapus ID partisipan remote dari idList**  removeParticipantId(id);  muteParticipant();  } |

Tabel VI.20. Fungsi hideMutedRemoteVideo()

|  |  |
| --- | --- |
| *Path* | modules/UI/videolayout/VideoLayout.js |
| *Source code* | hideMutedRemoteVideo(id) {  APP.UI.setVideoMuted(id, true);  APP.UI.setAudioMuted(id, true);  } |

Tabel VI.21. Fungsi showRemoteVideo()

|  |  |
| --- | --- |
| *Path* | modules/UI/videolayout/VideoLayout.js |
| *Source code* | showRemoteVideo(id) {  APP.UI.setAudioMuted(id, false);  APP.UI.setVideoMuted(id, false);  } |

# PENUTUP

## Kesimpulan

Pelaksanaan Praktik Kerja Lapangan (PKL) dilaksanakan di PT. Padepokan Tujuh Sembilan selama 17 minggu. Dalam selang waktu tersebut beberapa pekerjaan yang dilakukan, antara lain:

1. *refactoring* pada proyek *Field Service Management* (FSM),
2. mencari perangkat alternatif dan plugin SonarQube, serta eksplorasi pe-rangkat otomatisasi *refactoring* untuk kebutuhan perusahaan, dan
3. pengerjaan proyek *Jitsi Bandwidth Reduction*.

Proyek *Jitsi Bandwidth Reduction* bertujuan untuk mengurangi penggunaan *band-width* pada konferensi video yang menggunakan aplikasi Jitsi dengan membatasi jumlah data yang dapat diterima pengguna dalam kurun waktu tertentu. Hal ini dapat direalisasikan dengan menerapkan metode *pagination* pada alur pengiriman data konferensi video sehingga penggunaan *bandwdith* hanya terbatas pada pengi-riman data dari partisipan yang aktif pada laman (*page*) tertentu. Adapun, beberapa pencapaian pada pengerjaan proyek ini, di antaranya:

1. menginisiasi dan membentuk kerangka fungsi pemanggilan API dengan menggunakan AJAX pada komponen Jitsi Meet dan fungsi untuk mena-ngani tampilan UI setelah partisipan mengeksekusi fungsi pemanggilan API tersebut;
2. menginisaisi dan menyusun metode untuk mendapatkan urutan partisipan berdasarakan urutan *join* (bergabung) ke dalam ruang konferensi. Adapun, hal ini tidak dapat direalisasikan pada komponen Jitsi Videobridge (JVB) seperti harapan manajer proyek, namun dapat teratasi di Jitsi Meet;
3. membuat beberapa fungsi untuk kebutuhan proses *paging* di komponen JVB dan turut serta membantu rekan tim melakukan simulasi untuk men-dukung proses modifikasi fungsi selanjutnya;
4. turut serta melakukan analisis komponen Jitsi Conference Focus (Jicofo), mencetuskan beberapa ide yang mendukung kelangsungan analisis, dan menemukan beberapa hal penting yang menjadi *key-point* peralihan proses analisis ke fokus lain;
5. adapun, karena pengerjaan di proyek ini sebagian besar dihabiskan untuk melakukan analisis, mendapatkan penemuan-penemuan baru yang menen-tukan langkah yang akan diambil pada proyek ini selanjutnya.

## Lesson Learned

Dari pelaksanaan PKL di PT. Padepokan Tujuh Sembilan, penulis memetik bebe-rapa pembelajaran sebagai berikut.

1. Jangan membuang-baung waktu dengan memaksakan analisis walaupun petunjuk yang dimiliki rancu dan membingungkan. Hal ini dialami dalam proses analisis IFrame API. Waktu yang terbuang sia-sia karena bersi-keras untuk memeriksa semua unsur walaupun petunjuk yang dimiliki rancu, dan hal ini pun menghambat progres proyek. Seandainya saat itu penulis dapat menyikapi dengan langsung menyampaikan situasi kepada manajer proyek dan memaparkan asumsi bahwa *library* tidak bisa diguna-kan, atau menyatakan kebingungan dan meminta bantuan, waktu untuk proses analisis IFrame API akan lebih singkat, atau setidaknya akan dida-patkan petunjuk/keputusan yang lebih jelas dengan bantuan dari rekan tim lain. Dalam pengerjaan proyek, penulis tidak dapat melakukan eksplorasi sepuas hati karena hal itu hanya akan menghambat proyek – setiap jam yang terbuang memukul mundur penyelesaian proyek.
2. Ketika mengalami suatu kendala atau menemukan jalan buntu, solusi akan lebih cepat ditemukan apabila dilakukan diskusi dan bertukar pendapat dengan rekan tim. Pembuatan fungsi RESTful API di Jicofo dilakukan secara berembuk. Tim dapat membuat fungsi dengan lancar dan dengan kendala minim karena tim saling bertukar pikiran, dan bahu-membahu dalam melakukan analisis untuk membuat kedua fungsi tersebut. Walau-pun hal ini memang kurang efektif, namun terbukti bahwa kerja sama lebih unggul daripada pengerjaan solo. Pengerjaan secara solo untuk mencari solusi membuang-buang waktu.
3. Analisis secara mendetail dan telusuri setiap unsur yang berhubungan dengan topik secara mendalam. Pada proyek ini, pekerjaan utama yang dilakukan adalah melakukan analisis. Dari pengalaman yang didapatkan, analisis secara mendetail dan pemikiran *out-of-the-box* sangat membantu utnuk menemukan petunjuk-petunjuk baru bagi kelangsung analisis. Na-mun, perlu dicatat bahwa analisis yang mendetail harus menyesuaikan dengan waktu untuk melakukan analisis dan, ketika sudah mencapai titik buntu, minta masukan dan pandangan dari rekan lain. Dengan demikian, proses menjadi lebih efisein dan hasil analisis pun lebih maksimal.
4. Di tengah pengerjaan proyek ini, dua talenta perusahaan bergabung untuk membantu di pengerjaan proyek Jitsi Bandwidth Reduction. Pada saat itu, penulis diminta untuk membantu rekan tim baru mempelajari dan mem-biasakan diri dengan penugasan pada proyek ini. Dengan kata lain, penulis diminta untuk menjadi pembimbing dan “guru” bagi rekan tim baru. Pada awalnya, penulis tidak merasa percaya diri karena tidak yakin dengan kemampuan diri sendiri, selain itu penulis hanya mahasiswa yang magang, sedangkan rekan tim adalah talenta yang lebih berpengalaman. Namun, sedikit demi sedikit, penulis dapat menemukan kepercayaan diri dalam kemampuan penulis dan membantu rekan tim dengan membimbing dan “mengajari” mereka. Dari situ, penulis memetik beberapa pembelajaran, di antaranya:
   1. cara dan sikap untuk menyampaikan pendapat dan memberikan masukan kepada rekan tim secara halus dan sopan sehingga tidak menyinggung perasaan pihak terkait,
   2. kesabaran dan pemikiran terbuka dalam membimbing seseorang karena kemampuan setiap orang berbeda dalam menangkap dan mamahami suatu konsep,
   3. kapan harus melakukan komunikasi formal dan informal sehingga atmosfer percakapan tidak kaku.
5. Kejelasan dalam menyampaikan informasi sangat penting untuk mencegah misinformasi/miskominikasi. Dalam kasus di mana tim proyek dan klien tidak sepaham dalam bercakap tentang progres proyek, pengubahan alur proyek yang dilakukan dan pengerjaan *test case* dua minggu setelahnya dilakukan sia-sia.

## Saran

Subbab ini berisi saran dari penulis kepada PT. Padepokan Tujuh Sembilan dan Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, berkenaan pelaksanaan PKL yang telah dilakukan.

### PT. Padepokan Tujuh Sembilan

Saran dari penulis: tingkatkan komunikasi dengan peserta PKL. Dalam konteks ini, penulis merujuk pada komunikasi antara penulis dengan karyawan PT. Padepokan Tujuh Sembilan (di luar lingkup proyek) dan kendala komunikasi yang dialami selama pengerjaan proyek *Jitsi Bandwidth Reduction*, yaitu respon yang lambat dari manajer proyek ketika melakukan konfirmasi atau melaporkan kendala dan komu-nikasi yang minim di dalam tim. Penulis memang belum pernah benar-benar merasakan nuansa proyek, namun, penulis pun merasa komunikasi di dalam proyek *Jitsi Bandwidth Reduction* tidak maksimal. Adapun, saat penulis bertanya tentang hal ini kepada seorang rekan tim dari perusahaan, penulis diceritakan bahwa komu-nikasi di dalam tim biasanya lebih kondusif daripada yang dialami di proyek *Jitsi Bandwidth Reduction*. Adapun hal itu karena terdapat *team leader* untuk mengini-siasi hal tersebut.

Sehubungan dengan ini, penulis pun tidak yakin siapa *team leader* di proyek ini (penulis memiliki asumsi siapa *team leader*, namun sosok tersebut mendadak hilang di tengah pengerjaan proyek). Barangkali ada baiknya jika di awal pengerjaan proyek diadakan diperkenalkan antara semua unsur di dalam tim proyek beserta peran masing-masing untuk memudahkan kerja sama dan komunikasi.

### Jurusan Teknik Komputer dan Informatika Polban

Saran dari penulis: tingkatkan pemantauan kegiatan mahasiswa selama pelaksanaan PKL di perusahaan. Jika mungkin, evaluasi kinerja mahasiswa secara berkala tidak hanya dari pihak perusahaan, namun juga dari dosen pembimbing. Selain itu, dosen pembimbing dapat lebih terlibat dengan memberikan saran serta masukan untuk meningkatkan kualitas pekerjaaan mahasiswa.

# DAFTAR PUSTAKA

*Add ability to invite people by JID · Issue #6235 · jitsi/jitsi-meet · GitHub*. (n.d.).  
Tersedia: <https://github.com/jitsi/jitsi-meet/issues/6235> (Diakses 29 September 2020)

*AirTabel - Crunchbase Company Profile & Funding*. (n.d.).   
Tersedia: [https://www.crunchbase.com/organization/airTabel#/entity](https://www.crunchbase.com/organization/airtable#/entity) (Diakses 8 Desember 2020)

*AnyDesk | What is AnyDesk – Javatpoint*. (n.d.).  
Tersedia: <https://www.javatpoint.com/anydesk> (Diakses 2 Desember 2020)

*Architecture · Jitsi Meet Handbook*. (n.d.).  
Tersedia: <https://jitsi.github.io/handbook/docs/architecture> (Diakses 23 November 2020)

Axios Community. (2020). *axios/axios: Promise based HTTP client for the browser and node.js*.  
Tersedia: <https://github.com/axios/axios>

*Berkenalan Dengan IntelliJ IDEA - Blog | Dicoding Indonesia | Blog | Dicoding Indonesia*. (n.d.).  
Tersedia: <https://www.dicoding.com/blog/berkenalan-dengan-intellij-idea/> (Diakses 27 November 2020)

D’Arcy, H. (2009). IntelliJ IDEA. *RefCardz*.  
Tersedia: <https://www.jetbrains.com/help/idea/discover-intellij-idea.html#IntelliJ-IDEA-editions>

*Developer Guide (Web) · Jitsi Meet Handbook*. (n.d.).  
Tersedia: <https://jitsi.github.io/handbook/docs/dev-guide/dev-guide-web> (Diakses September 2020)

*DevOps Platform Delivered as a Single Application | GitLab*. (n.d.). Tersedia: <https://about.gitlab.com/> (Diakses 2 Desember 2020)

Devosa, I., & Csallner, A. (2010). *Introduction to Java Programming Language.*Tersedia:[https://medium.com/@rinu.gour123/introduction-to-java-programming-language-389033edb91f](https://medium.com/@rinu.gour123/introduction-to-java-programming-language-389033edb91f%20)

Examiner, P., & Chan, W. F. (1998). *United States Patent 19 V V v*.

Facebook. (2020). *Tutorial: Intro to React – React*.  
Tersedia: <https://reactjs.org/tutorial/tutorial.html#what-is-react>

Framework, B. A. R., & Tips, A. (2010). What is a REST API? *MuleSoft*, *13*, 1 –12.  
Tersedia: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>

*GlobalProtect - Palo Alto Networks*. (n.d.).  
Tersedia: <https://www.paloaltonetworks.com/products/globalprotect> (Diakses 3 Desember 2020)

Google. (2019). *Google Docs: Free Online Documents for Personal Use*. Tersedia: <https://www.google.com/docs/about/>

Google. (2019). *Google Docs: Free Online Documents for Personal Use*.  
Tersedia: <https://www.google.com/docs/about/>

Grozev, B., Marinov, L., Singh, V., & Ivov, E. (2015). Last N: Relevance-based selectivity for forwarding video in multimedia conferences. *Proceedings of* the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 2015, 19 – 24.  
<https://doi.org/10.1145/2736084.2736094>

*Helpful Features for Your Browser – Google Chrome.* (n.d.). Tersedia: <https://www.google.com/chrome/browser-features/> (Diakses 3 Desember 2020)

*How to match the conference ID between Jitsi-meet and Rest Colibri? - Developers - Jitsi Community Forum - developers & users*. (n.d.).  
Tersedia: <https://community.jitsi.org/t/how-to-match-the-conference-id-between-jitsi-meet-and-rest-colibri/14360> (Diakses 12 Oktober 2020)

*How to Use Google Meet Video Conferencing | Google Meet*. (n.d.).   
Tersedia: <https://apps.google.com/meet/how-it-works/> (Diakses 5 Desember 2020)

*Jisti Conference with Up To 100 Participant - Install & Config - Jitsi Community Forum - developers & users*. (n.d.).  
Tersedia: <https://community.jitsi.org/t/jisti-conference-with-up-to-100-participant/37613> (Diakses 6 November 2020)

*Lua: About*. (2018). Lua.  
Tersedia: <https://www.lua.org/about.html>

Luis, A., Negi, S., & Mills, C. (2019). *Getting Started - Developer guides | MDN*.  
Tersedia: [https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/  
Getting\_Started](https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started)

Microsoft. (2018). *FAQs about Windows Subsystem for Linux | Microsoft Docs*. Tersedia: <https://docs.microsoft.com/en-us/windows/wsl/about>

*Mute video - Developers - Jitsi Community Forum - developers & users*. (n.d.). Tersedia: <https://community.jitsi.org/t/mute-video/14885/6> (Diakses 10 Oktober 2020)

*Mute Video for Remote User - Install & Config - Jitsi Community Forum - developers & users*. (n.d.).  
Tersedia: <https://community.jitsi.org/t/mute-video-for-remote-user/58695> (Diakses 10 Oktober 2020)

*Node.js – Introduction – Tutorialspoint*. (2019).  
Tersedia: <https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm>

Peham, T. (2020). *GitLab vs GitHub: What Are the Key Differences?*  
Tersedia: https://usersnap.com/blog/gitlab-github/

*Pengertian Bandwidth dan Fungsinya (Lengkap) - Niagahsoter*. (n.d.).  
Tersedia:<https://www.niagahoster.co.id/blog/pengertian-bandwidth/#Apa_Itu_Bandwidth_Internet> (Diakses 25 Desember 2020)

Point, T. (n.d.). *Ubuntu - Overview - Tutorialspoint*.  
Tersedia: [https://www.tutorialspoint.com/ubuntu/ubuntu\_overview.htm](https://www.tutorialspoint.com/ubuntu/ubuntu_overview.html) (Diakses 30 November 2020)

ssh.com. (2020). SSH Protocol – Secure Remote Login and File Transfer. In *SSH Communications Security*.  
Tersedia: <https://www.ssh.com/ssh/protocol/>

Tersedia: [https://prosody.im/doc/xmpp](https://prosody.im/doc/xmpp%20)

Tersedia: <https://restfulapi.net/> (Diakses 25 November 2020)

Visual Studio Code. (2015). *Documentation for Visual Studio Code*. Visual Studio Code Documentation.  
Tersedia: <https://code.visualstudio.com/docs>

*WebRTC*. (n.d.).  
Tersedia: <https://webrtc.org/> (Diakses 26 Desember 2020)

*What is JavaScript? - Learn web development | MDN*. (2019).  
Tersedia: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript>

*What is Jitsi? – open-source video conferencing API's, SDKs, and installers.* (n.d.).  
Tersedia: <https://jitsi.org/what-is-jitsi/> (Diakses 10 Desember 2020)

*What is REST – REST API Tutorial*. (n.d.). restfulapi.net.

WhatsApp Inc. (2018). *About WhatsApp*. WhatsApp.com.  
Tersedia: <https://www.whatsapp.com/about/>

*XMPP – Prosody IM*. (2018).  
Tersedia: <https://prosody.im/doc/xmpp>

# LAMPIRAN

**Lampiran 1**

Rencana Penyelesaian Pekerjaan Proyek Jitsi – Bandwidth Reduction

PT. Padepokan Tujuh Sembilan

**PKL MAHASISWA**

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**TAHUN 2020**

****

**RENCANA PENYELESAIAN PEKERJAAN**

**Oleh**

Sophia Gianina Daeli NIM 171524029

**RENCANA PENYELESAIAN PEKERJAAN**

1. **Judul Pekerjaan**

Proyek *Jitsi Bandwidth Reduction* adalah proyek yang bertujuan untuk mengurangi penggunaan *bandwidth* pada konferensi video dengan menggunakan aplikasi Jitsi. Hal ini dilakukan untuk menjaga performa konferensi video sehingga komunikasi antar pengguna tidak terhambat. Untuk mencapai tujuan tersebut, klien berupaya untuk mengurangi penggunaan *bandwidth* dengan membatasi jumlah data (audio dan video) yang dapat diterima pengguna dalam kurun waktu tertentu.

1. **Peran Kelompok PKL dalam Pekerjaan**

Penulis terlibat dalam pengerjaan proyek *Jitsi Bandwidth Reduction* sebagai *back end developer*. Adapun, cakupan tugas yang diberikan, yaitu: analisis metode untuk mengurangi penggunaan *band-width* pada konferensi video dengan menggunakan aplikasi Jitsi, implementasi modifikasi dan penambahan fungsi, dan menguji serta *bug-fixing* hasil modifikasi dan fungsi yang ditambahkan.

1. **Deskripsi Tugas**

Tugas yang diberikan kepada tim proyek adalah mengembangkan aplikasi Jitsi dengan menambahkan fungsi *paging* untuk membatasi pengiriman data (audio dan video) dari partisipan *remote* yang dapat diterima partisipan lokal. Untuk mencapai tujuan tersebut, tim melakukan analisis terhadap beberapa komponen yang menyu-sun aplikasi Jitsi, yaitu: Jitsi Meet, Jitsi Videobridge (JVB), dan Jitsi Conference Focus (Jicofo) untuk mambuat rancangan alur fungsi *pagination*. Kemudian, tim melakukan implementasi fungsi dan modifikasi sesuai kebutuhan proyek, dan ter-akhir, tim melakukan pengujian hasil implementasi untuk memastikan fungsi yang ditambahkan dan modifikasi yang dibuat sesuai dengan kebutuhan tim dan berhasil mengurangi penggunaan *bandwidth* pada konferensi.

1. **Deliverables**

Pada kegiatan ini, *deliverables* yang harus diberikan kepada pembimbing PKL instansi PT. Padepokan Tujuh Sembilan dijabarkan pada Tabel 1.

Tabel 1. Deliverables dan tenggat waktu

|  |  |
| --- | --- |
| **Deliverables** | **Due date** |
| Hasil analisis elemen dan atribut XML XMPP dari log prosody konferensi Jitsi | 8 September 2020 |
| Hasil simulasi bandwidth | 17 September 2020 |
| Hasil implementasi fungsi RESTful API di proyek JVB | 23 September 2020 |
| Hasil analisis proyek Jitsi Meet dan *library* lib-jitsi-meet untuk pengiriman Stanza IQ *custom* di Prosody | 29 September 2020 |
| Hasil implementasi fungsi pemangilan RESTful API dengan menggunakan AJAX di proyek Jitsi Meet | 14 Oktober 2020 |
| Hasil analisis proyek Jicofo untuk mendapatkan semua partisipan di sebuah konferensi | 22 Oktober 2020 |
| Hasil analisis proyek Jicofo untuk proses pengiriman paket audio dan video participant | 4 November 2020 |
| Hasil stress test Jitsi untuk mendapatkan octo participant pada konferensi | 16 November 2020 |
| Hasil implementasi fungsi *pagination* di proyek Jitsi Meet | 19 November 2020 |
| Laporan dokumentasi progres untuk klien | 4 Desember 2020 |

1. **Milestones**

Milestones yang ditetapkan selama waktu pengerjaan proyek dijabarkan pada Tabel 2.

Tabel 2. Tabel milestones

|  |  |  |
| --- | --- | --- |
| **Milestones** | **Tanggal** | **Deskripsi** |
| a | 23 September 2020 | Implementasi fungsi RESTful API di proyek JVB |
| b | 13 Oktober 2020 | Implementasi fungsi pemangilan RESTful API dengan menggunakan AJAX di proyek Jitsi Meet |
| c | 19 November 2020 | Implementasi fungsi *pagination* di proyek Jitsi Meet |

1. **Rencana Capaian Per Minggu**

Rencana capaian per minggu yang ditetapkan selama waktu pengerjaan proyek dijabarkan pada Tabel 3.

Tabel 3. Rencana capaian

|  |  |
| --- | --- |
| **Minggu ke-** | **Rencana Capaian** |
| 5 | Sprint 1:   * *Setting environment* proyek Jitsi Meet * Mempelajari dan analisis metode Last-N untuk mengurangi jumlah transmisi data pada konferensi video |
| 6 | Sprint 1:   * Analisis dan implementasi RESTful API di proyek JVB * Analisis proyek Jitsi Meet untuk pengiriman Stanza IQ *custom* di Prosody * Simulasi bandwidth |
| 7 | Sprint 2:   * Analisis dan implementasi RESTful API di proyek JVB * Analisis proyek Jitsi Meet untuk pengiriman Stanza IQ *custom* di Prosody * Analisis *library* lib-jitsi-meet untuk pengiriman Stanza IQ *custom* di Prosody |
| 8 | Sprint 2:   * Analisis *library* lib-jitsi-meet untuk pengiriman Stanza IQ *custom* di Prosody * Analisis proyek Jitsi Meet untuk melakukan execute HTTP request API ke proyek JVB atau Jicofo |
| 9 | Sprint 3:   * Implementasi fungsi pemangilan RESTful API dengan menggunakan AJAX di proyek Jitsi Meet |
| 10 | Sprint 3:   * Implementasi dan menguji fungsi pemangilan RESTful API dengan menggunakan AJAX di proyek Jitsi Meet |
| 11 | Sprint 4:   * Analisis proyek Jicofo untuk mendapatkan semua partisipan di sebuah konferensi * Implementasi dan menguji fungsi untuk mendapatkan semua partisipan di sebuah konferensi di proyek Jicofo |
| 12 | Sprint 4:   * Analisis proyek Jicofo untuk proses pengiriman paket audio dan video participant |
| 13 | Sprint 5:   * Analisis proyek Jicofo untuk proses pengiriman paket audio dan video participant * Stress test Jitsi untuk mendapatkan octo participant pada konferensi |
| 14 | Sprint 5:   * Analisis proyek Jicofo untuk proses pengiriman paket audio dan video participant * Stress test Jitsi untuk mendapatkan octo participant pada konferensi   Analisis proyek JVB untuk menerapakan meteode *pagination* |
| 15 | Sprint 5:   * Stress test Jitsi untuk mendapatkan octo participant pada konferensi * Analisis proyek JVB untuk menerapakan meteode *pagination* * Analisis dan implementasi fungsi *pagination* di proyek Jitsi Meet |
| 16 | Sprint 6:   * Modifikasi fungsi *pagination* di proyek Jitsi Meet agar sesuai dengan tampilan *front end* yang telah dibuat oleh klien |
| 17 | Sprint 6:   * Menyusun dokumentasi progres untuk klien |

1. **Jadwal Penyelesaian Pekerjaan Keseluruhan**

Pengembangan proyek Jitsi dimulai pada minggu ke-5 tanggal 7 September 2020, dan selesai pada minggu ke-15 tanggal 4 Desember 2020. Jadwal penyelesaian pekerjaan keseluruhan dapat dilihat pada Tabel 4.

Tabel 4. Jadwal penyelesaian pekerjaan keseluruhan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Butir Pekerjaan** | **Mimggu ke-** | | | | | | | | | | | | |
| **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| *Setting environment* proyek Jitsi Meet |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Mempelajari dan analisis metode Last-N untuk mengurangi jumlah transmisi data pada konferensi video |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis dan implementasi RESTful API di proyek JVB |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis proyek Jitsi Meet untuk pengiriman Stanza IQ *custom* di Prosody |  |  | **milestone a** |  |  |  |  |  |  |  |  |  |  |
| Simulasi bandwidth |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis *library* lib-jitsi-meet untuk pengiriman Stanza IQ *custom* di Prosody |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis proyek Jitsi Meet untuk melakukan execute HTTP request API ke proyek JVB atau Jicofo |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi fungsi pemangilan RESTful API dengan menggunakan AJAX di proyek Jitsi Meet |  |  |  |  |  | **milestone b** |  |  |  |  |  |  |  |
| Analisis proyek Jicofo untuk mendapatkan semua partisipan di sebuah konferensi |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi dan menguji fungsi untuk mendapatkan semua partisipan di sebuah konferensi di proyek Jicofo |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis proyek Jicofo untuk proses pengiriman paket audio dan video participant |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Stress test Jitsi untuk mendapatkan octo participant pada konferensi |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis proyek JVB untuk menerapakan meteode *pagination* |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis dan implementasi fungsi *pagination* di proyek Jitsi Meet |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Modifikasi fungsi *pagination* di proyek Jitsi Meet agar sesuai dengan tampilan *front end* yang telah dibuat oleh klien |  |  |  |  |  |  |  |  |  |  |  | **milestone c** |  |
| Menyusun dokumentasi progres untuk klien |  |  |  |  |  |  |  |  |  |  |  |  |  |