

Rancang Bangun Aplikasi Latihan *Public Speaking* Berbasis *iOS* Menggunakan Teknologi *Live Streaming Video Agora.io*

Design and Build an *iOS*-Based Public Speaking Practice Application Using Live Streaming Video Technology *Agora.io*

Reinaldo Lewis Lordianto, Mychael Maoeretz Engel*

Program Studi Informatika, Universitas Ciputra, Surabaya 60219, Indonesia

(*Email Korespondensi: mychael.engel@ciputra.ac.id)

Abstrak: Berbicara di depan umum (*Public Speaking*) adalah kegiatan presentasi secara langsung yang diberikan di depan audiens. Berbicara di depan umum bisa menjadi hal yang sangat ampuh jika digunakan dengan benar. Kita dapat membujuk dan meyakinkan orang lain, mendapatkan kepercayaan mereka, membagikan ide dan pengetahuan kita dengan percaya diri kepada mereka. Namun, sayangnya banyak orang menderita kecemasan berbicara atau takut berbicara di depan umum (*Public Speaking Anxiety / Speech Anxiety*) karena pengalaman yang tidak menyenangkan atau bahkan diminta untuk tampil di tempat tanpa persiapan. Pusat Nasional Neurologi dan Psikiatri melaporkan bahwa ketakutan berbicara dan berbicara di depan umum mempengaruhi sekitar 73% populasi. Sehingga, untuk mengurangi masalah tersebut maka perlu dibuatkan suatu aplikasi yang berguna untuk melatih kemampuan *public speaking* secara online yang mudah digunakan dan dapat mengundang komunitas atau teman yang juga menggunakan aplikasi agar menjadi lebih terbiasa dan tidak takut lagi untuk melakukan *public speaking* dengan memanfaatkan fitur *live video streaming* yang disediakan oleh platform *third party* yaitu *agora.io*. Aplikasi ini bisa digunakan oleh banyak orang. berdasarkan data sebagian besar orang pasti mengalami ketakutan berbicara didepan umum terutama saat menginjak masa perkuliahan yang diharuskan untuk siap melakukan presentasi saat adanya *project*. Aplikasi ini dapat digunakan untuk latihan didepan teman yang kita kenal atau komunitas yang tersedia di dalam aplikasi ini, dan dapat saling menerima dan memberikan feedback satu sama lain.

Kata Kunci: Public Speaking, Anxiety, Speech, Live, Video, Glossophobia

Abstract: Public speaking is a live presentation activity given in front of an audience. Public speaking can be a very powerful thing if used properly. We can persuade and convince others, earn their trust, share our ideas and knowledge with confidence with them. However, unfortunately many people suffer from speech anxiety or are afraid to speak in public (*Public Speaking Anxiety / Speech anxiety*) due to unpleasant experiences or even being asked to appear in a place without preparation. The National Institute of Mental Health reviews that public talking anxiety, or glossophobia, impacts approximately 73% of the population. So, to reduce this problem, it is necessary to make an application that is useful for practicing online public speaking skills that is easy to use and can invite communities or friends who also use the application to become more familiar and not afraid to do public speaking anymore by utilizing the live video streaming feature that is available. provided by a third-party platform, namely *agora.io*. This application can be used by many people because based on the data, most people must be afraid of speaking in public, especially when stepping on the lecture period which is required to be ready to make presentations when there is a project. This application can be used for practice in front of friends we know or the community available in this application. Also enable to receive and provide feedback to each other.

Keywords: Public Speaking, Anxiety, Speech, Live, Video, Glossophobia



1. Pendahuluan

1.1 Latar Belakang

Berbicara di depan umum (*Public Speaking*) adalah kegiatan presentasi secara langsung yang diberikan di depan audiens. Berbicara di depan umum bisa menjadi hal yang sangat ampuh jika digunakan dengan benar. Kita dapat membujuk dan meyakinkan orang lain, mendapatkan kepercayaan mereka, membagikan ide dan pengetahuan kita dengan percaya diri kepada mereka. Hal ini dapat membantu kita dalam interaksi sehari-hari yang paling sederhana sekalipun untuk memiliki dampak besar pada jalur karier dan tingkat kesuksesan di industri.

Namun, sayangnya banyak orang menderita kecemasan berbicara atau takut berbicara di depan umum (*Public Speaking Anxiety / Speech anxiety*) karena pengalaman yang tidak menyenangkan atau bahkan diminta untuk tampil di tempat tanpa persiapan (Fadli, 2020). Takut berbicara di depan umum adalah fobia paling umum menjelang kematian, laba-laba, atau ketinggian. National Institute of Mental Health melaporkan bahwa kecemasan berbicara di depan umum, atau glossophobia, mempengaruhi sekitar 73% populasi. di depan umum, *Public Speaking Anxiety* mungkin akan membuat seseorang khawatir berminggu-minggu atau berbulan-bulan sebelum pidato atau presentasi, dan orang tersebut mungkin mengalami gejala fisik kecemasan yang parah selama pidato, seperti, gemetar, malu, jantung berdebar, suara bergetar, sesak nafas, pusing, dan bahkan sakit perut.

Ketakutan berbicara di depan umum ini tidak dapat disembuhkan, tetapi dapat dikurangi dengan banyak cara. Dengan kata lain, jelajahi ketakutan dan pikiran negatif yang mengganggu Anda: "Ketika berbicara di depan umum, pastikan semua orang menerima kesalahan dan kelalaian. Tidak masalah. Sebagian besar pemirsa mungkin tidak menyadarinya. Semua orang akan berpikir saya tidak kompeten. Hindari berpikir, dan fokuslah pada kenyataan bahwa audiens Anda ingin Anda sukses. Kedua, ingat bahwa materi yang disiapkan sudah cukup baik dan dikuasai. Setelah mengidentifikasi rasa takut, berlatihlah mempresentasikannya kepada kelompok pendukung kecil. Jika Anda menjadi lebih percaya diri dalam berbicara di depan kelompok kecil, bukan tidak mungkin untuk membangun kepercayaan diri itu di hadapan audiens yang lebih besar.

Dalam penelitian ini, akan berfokus untuk mengurangi *Public Speaking Anxiety* dengan cara mengembangkan sebuah aplikasi yang dapat digunakan untuk berlatih kemampuan *public speaking* dengan menggunakan fitur *live video streaming* dan dapat mengundang komunitas untuk melihat dan melakukan penilaian berdasarkan performa dari hasil *public speaking* yang telah dilakukan.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah bagaimana merancang dan membangun aplikasi latihan *public speaking* berbasis *iOS* menggunakan teknologi *live streaming video Agora.io*.

1.3 Tujuan

Tujuan dari penelitian ini adalah merancang dan membangun aplikasi latihan *public speaking* berbasis *iOS* menggunakan teknologi *live streaming video Agora.io*.

1.4 Manfaat

1.4.1 Teoritis

Hasil penelitian diharapkan dapat menambah ilmu pengetahuan tentang pentingnya melatih kemampuan *public speaking* dan cara mengurangi rasa takut saat akan melakukan *public speaking*.

1.4.2 Praktis

1. Bagi penulis dapat menambah wawasan dan pengalaman langsung tentang cara meningkatkan kemampuan *public speaking*.
2. Bagi pengguna aplikasi (mahasiswa/mahasiswi) bisa digunakan sebagai tempat latihan untuk meningkatkan kemampuan *public speaking*.

1.5 Ruang Lingkup

Penelitian ini memiliki ruang lingkup:

1. Aplikasi ini hanya tersedia untuk device berbasis *IOS*
Aplikasi ini dikembangkan menggunakan *Xcode* dengan bahasa pemrograman *Swift (SwiftUI)*. Dimana hanya dapat menghasilkan aplikasi yang dapat dijalankan ke perangkat *IOS*.
2. Menyimpan data *recording video* menggunakan *third-party*
Hasil *Recording live video* yang dihasilkan dapat disimpan menggunakan *third-party database* dimana pada penelitian ini akan menggunakan sistem *Firebase*.

2. Kajian Pustaka

2.1 Studi Terdahulu

Ada beberapa penelitian terdahulu yang membahas mengenai *live streaming video*. Artikel (Wang, et al., 2020) membuat sebuah sistem *live streaming* yang digunakan untuk *multi-party interactive live streaming* agar saat melakukan *live streaming* dengan fitur yang dapat penontonnya lebih interaktif dengan *host* dengan membuat *quality of experience (QoE)* model untuk *multi-party live streaming*, dan menggunakan pengelompokan *bitrate* untuk membuat performa *live streaming* lebih yang akan dikirim dari streamer menjadi lebih lancar. Mereka menghasilkan fitur yang meningkatkan kecepatan dua sampai lima kali lipat dari *QoE* rata-rata. Selanjutnya, penundaan ujung-ke-ujung telah dikurangi menjadi sekitar 100 ms, yang jauh lebih rendah dari 400 ms yang digunakan sebagai ambang batas penundaan dalam skema konferensi video yang ada.

Artikel (Rodriguez-Gil, et al., 2017) mengatakan bahwa sistem *live streaming* sekarang sudah baik terutama pada aplikasi-aplikasi besar seperti Youtube Live, Twitch TV, Instagram Livestream, dan Facebook Live. tetapi, mereka memiliki kekurangan yaitu sistem mereka yang memiliki limitasi yang membuat fitur mereka kurang interaktif dengan penontonnya. Limitasi ini seperti, adanya penundaan (*delay*) antara penonton dan *host* yang relatif tinggi setidaknya beberapa detik. Untuk mengatasi masalah di atas, mereka menggunakan sebuah platform yang bernama WILSP, yang didesain untuk membuat sistem *live streaming* yang lebih interaktif dan mengurangi limitasi penundaan (*delay*) yang ada. Dan kelebihan dari platform ini yaitu bersifat *Open-Source*.

Artikel (Shang, et al., 2021) membuat penargetan basis data kualitas video skala besar gerakan tinggi, dalam skenario *streaming* langsung. Sumber daya baru mencakup 45 urutan sumber dari 33 konten asli dan 6 jenis distorsi yang berbeda. Basis data baru dapat digunakan untuk membuat, menguji, dan membandingkan model NR dan FR VQA (*Video Quality Assessment*). Mereka membuat database *LIVE Livestream* baru tersedia untuk umum. Langkah-langkah masa depan termasuk mengembangkan model NR VQA baru menggunakan database yang diusulkan.

2.2 SwiftUI

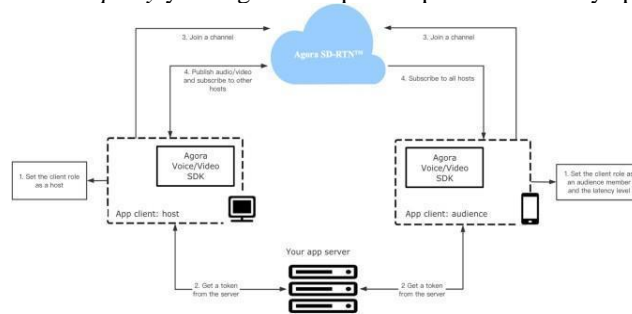
Swift adalah bahasa pemrograman yang kuat dan intuitif untuk macOS, iOS, watchOS, dan tvOS (Ham, 2019). Menulis kode Swift bersifat interaktif dan menyenangkan, sintaksnya ringkas namun ekspresif, dan Swift menyertakan fitur terbaru yang disukai pengembang kode Swift dan menghasilkan perangkat lunak yang berjalan sangat cepat.

SwiftUI dapat membangun aplikasi yang dapat bekerja di semua platform *Apple* dengan menggunakan *Swift* dan *code* sesedikit mungkin. Dengan *SwiftUI*, *developer* dapat menghadirkan pengalaman yang lebih baik lagi bagi semua pengguna, di perangkat *Apple* apa pun. Untuk memulai coding *SwiftUI*, harus menggunakan aplikasi *Xcode*. Membangun tampilan dan sistem aplikasi menjadi lebih mudah dengan menggunakan aplikasi ini karena sistem codenya yang lebih mudah dan sedikit, terdapat sistem *drag and drop* untuk mengatur warna, huruf, dan peletakan komponen. Kemudian yang paling menarik yaitu adanya sistem *preview*, dimana *developer* dapat melihat dengan *real-time* bagaimana hasil dari aplikasi yang sedang dikembangkan.

2.3 Agora.io

Agora.io adalah platform yang memudahkan penerapan panggilan video dan mengirimkannya langsung ke aplikasi dan situs web iOS/Android. Platform Agora.io adalah platform berbasis cloud yang sesuai dengan HIPAA, mendukung WebRTC, dan memiliki fitur seperti panggilan grup, obrolan video, obrolan video grup, siaran interaktif, dan fitur komunikasi lainnya. Pengembang dapat menggunakan Agora SDK dan WebRTC API untuk

mengintegrasikan aplikasi web dan seluler dengan produk komunikasi yang dikembangkan oleh Agora.io. *Streaming* yang disediakan oleh platform *third-party* yaitu agora.io seperti dapat dilihat alurnya pada Gambar 1.



Gambar 1. Flowchart Agora Live Streaming Video

2.4 Firebase (Database)

Firebase Realtime Database adalah database yang dihosting di cloud (Firebase, 2022). Data disimpan sebagai JSON dan disinkronkan secara real time dengan setiap klien yang terhubung. Saat Anda membuat aplikasi lintas platform menggunakan SDK platform Apple, Android, dan JavaScript, semua klien berbagi instansi Realtime Database yang sama dan secara otomatis mendapatkan perubahan data terbaru.

2.5 Swift

Swift adalah bahasa pemrograman yang kuat dan intuitif yang dikembangkan oleh *Apple* untuk mengembangkan aplikasi untuk *iOS*, *Mac*, *Apple TV*, dan *Apple Watch* (Apple, n.d.). Bahasa ini dirancang untuk memberi pengembang lebih banyak kebebasan. *Swift* mudah digunakan dan *open source*, jadi siapa pun yang memiliki ide dapat membuat hal-hal hebat. *Swift* adalah bahasa yang cepat dan efisien yang menyediakan pelaporan waktu nyata dan terintegrasi secara mulus dengan *Code Objective C* yang ada. Ini memungkinkan pengembang untuk menulis kode yang lebih aman dan andal, menghemat waktu, dan menciptakan pengalaman aplikasi yang lebih kaya.

2.6 XCode

Xcode adalah IDE untuk aplikasi iOS (Nosuke, 2020). Xcode memiliki banyak alat untuk proses pengembangan, termasuk manajemen perangkat, debugger, pembuat UI, alat profil, kontrol sumber, dan dokumentasi. Sayangnya, Xcode hanya dapat dijalankan di sistem macOS. Saat ini, tutorial Xcode alternatif mungkin tersedia di platform non-macOS, tetapi Xcode biasanya tidak optimal.

3. Metode Penelitian

3.1 Metode Analisis

3.1.1 Analisis Permasalahan

Public Speaking adalah kegiatan presentasi secara langsung yang diberikan di depan audiens. Berbicara di depan umum bisa menjadi hal yang sangat ampuh jika digunakan dengan benar. Kita dapat membujuk dan meyakinkan orang lain, mendapatkan kepercayaan mereka, membagikan ide dan pengetahuan kita dengan percaya diri kepada mereka. Namun, sayangnya banyak orang menderita kecemasan berbicara atau takut berbicara di depan umum (*Public Speaking Anxiety / Speech Anxiety*) karena pengalaman yang tidak menyenangkan atau bahkan diminta untuk tampil di tempat tanpa persiapan. Takut berbicara di depan umum adalah ketakutan yang paling umum akan kematian, laba-laba, atau ketinggian. Pusat Nasional Neurologi dan Psikiatri melaporkan bahwa ketakutan berbicara dan berbicara di depan umum mempengaruhi sekitar 73% populasi. di depan umum, *Public Speaking Anxiety* mungkin akan membuat seseorang khawatir berminggu-minggu atau berbulan-bulan sebelum pidato atau presentasi, dan orang tersebut mungkin mengalami gejala fisik kecemasan yang parah selama pidato, seperti, gemetar, malu, jantung berdebar, suara bergetar, sesak nafas, pusing, dan bahkan sakit perut.

Untuk mengurangi masalah tersebut maka perlu dibuatkan suatu aplikasi yang berguna untuk melatih

kemampuan *public speaking* secara online yang mudah digunakan dan dapat mengundang komunitas atau teman yang juga menggunakan aplikasi agar menjadi lebih terbiasa dan tidak takut lagi untuk melakukan *public speaking* dengan memanfaatkan fitur *live video*

Agora.io adalah platform yang memudahkan penerapan panggilan video dan mengirimkannya langsung ke aplikasi dan situs web iOS / Android. Platform Agora.io adalah platform berbasis cloud yang sesuai dengan HIPAA, mendukung WebRTC, dan memiliki fitur seperti panggilan grup, obrolan video, obrolan video grup, siaran interaktif, dan fitur komunikasi lainnya. Pengembang dapat menggunakan Agora SDK dan WebRTC API untuk mengintegrasikan aplikasi web dan seluler dengan produk komunikasi yang dikembangkan oleh Agora.io.

Metode pengembangan yang digunakan dalam penelitian ini untuk mengatasi permasalahan di atas adalah dengan metode agile karena, hal ini sangat berguna agar aplikasi yang dikembangkan bisa lebih membantu apa saja yang menjadi masalah utama masyarakat yang dapat diselesaikan dengan aplikasi ini.

3.1.2 Analisis Kebutuhan

Analisis kebutuhan sistem sangat penting untuk mendukung kinerja aplikasi terlepas dari apakah aplikasi memenuhi persyaratan. Persyaratan sistem adalah untuk membantu Anda mencapai tujuan aplikasi Anda.

1. Persyaratan perangkat keras Sistem bekerja dengan baik dan berfungsi penuh. Perangkat keras yang dibutuhkan untuk membangun aplikasi ini adalah :
 - MacBook
 - Processor Intel Core 7
 - RAM 8 GB
 - SSD 320 GB
2. Persyaratan perangkat lunak Perangkat lunak yang digunakan untuk mendukung pembuatan dan pengoperasian program aplikasi ini adalah :
 - Sistem operasi MacOS
 - XCode
 - Agora.io (*Third-Party Platform*)

3.1.2 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan untuk pengembangan aplikasi ini adalah wawancara, karena kita dapat dengan langsung bertanya lebih dalam jika terdapat pendapat yang menarik dari respondennya.

3.1.3 Responden

Mahasiswa/mahasiswi yang tidak percaya diri, merasa gugup dan gelisah ketika harus presentasi.

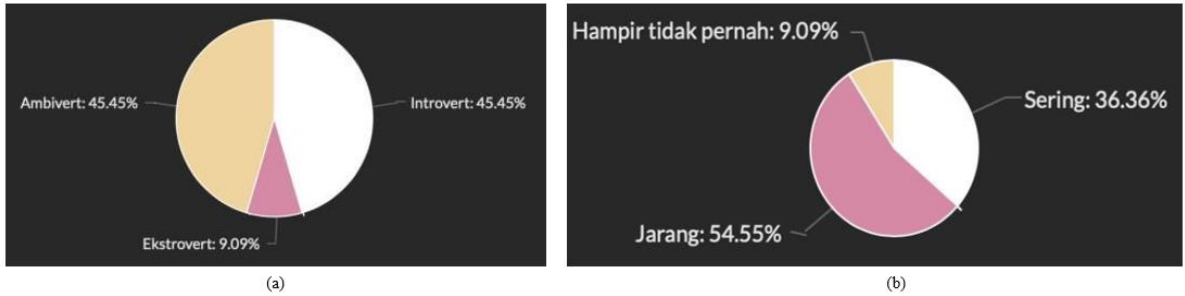
3.1.4 Instrumen Penelitian / Pertanyaan

Melakukan wawancara dengan mahasiswa/mahasiswi yang pernah atau sering melakukan *public speaking* dengan pertanyaan-pertanyaan sebagai berikut:

1. Dalam tipe berkomunikasi, apakah Anda termasuk introvert atau ekstrovert?
2. Bagaimana perasaan Anda dan kesulitan apa saja yang pernah dihadapi saat *public speaking* waktu itu?
3. Apa tujuan Anda melakukan *public speaking*? Kapan Anda membutuhkan *public speaking*?
4. Apakah Anda pernah mempelajari tentang *public speaking*? Dari mana Anda mempelajarinya?
5. Saat mempelajari *public speaking*, cara apa saja yang membuat Anda nyaman sehingga lebih mudah mengerti *public speaking*?
6. Jika Anda diberi kesempatan untuk *public speaking* lagi, apa Anda akan melakukannya?

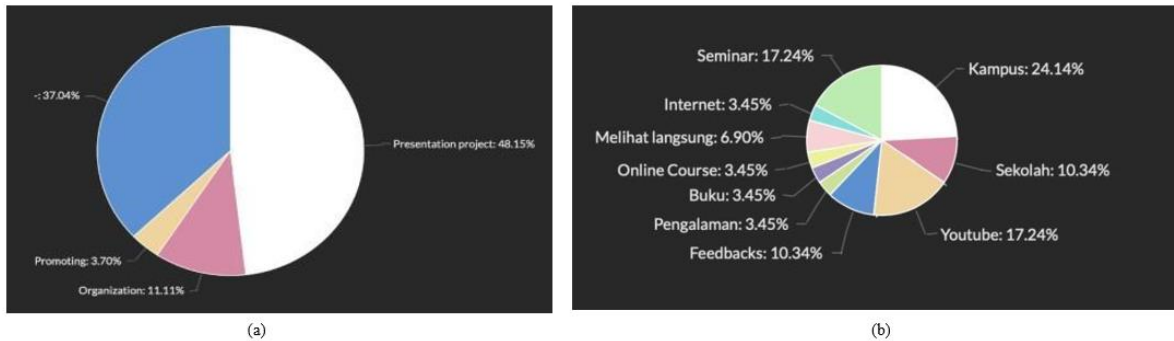
3.1.5 Hasil yang Diperoleh

Pada Gambar 2 (a), dapat dilihat bahwa sebagian besar dari responden adalah introvert atau ambivert. Dari Gambar 2 (b), dapat dilihat bahwa sekitar 36% sering melakukan *public speaking*, 54% jarang melakukannya, dan 9% hampir atau bahkan tidak pernah melakukan *public speaking*.



Gambar 2. Hasil jawaban responden tentang tipe berkomunikasi (a) dan frekuensi public speaking mereka (b)

Dari Gambar 3 (a), dapat dilihat bahwa sekitar 48% responden menjawab bahwa mereka melakukan presentasi untuk kepentingan proyek kuliah atau tugas-tugas kuliah dan sisanya ada yang untuk organisasi maupun promosi. Gambar 3 (b) menunjukkan bahwa sekitar 24% responden menjawab bahwa mereka belajar mengenai *public speaking* dari kampus. Sisanya ada yang belajar dari seminar, *Youtube*, sekolah, dan lain- lain.



Gambar 3. Hasil jawaban responden tentang tujuan mereka melakukan public speaking (a) dan dari mana mereka belajar *public speaking* (b)

Dari Gambar 4 (a), dapat dilihat bahwa 30% dari responden menjawab untuk mengerti bagaimana cara *public speaking* yang baik harus latihan dan 30% lainnya menjawab belajar lebih dalam mengenai *public speaking*, lalu sisanya ada yang menjawab belajar dari ahli maupun mendapatkan masukan dari orang lain. Dari Gambar 4 (b), dapat disimpulkan bahwa 80% dari responden masih menginginkan untuk mencoba lagi melakukan *public speaking* untuk dapat meningkatkan kemampuan mereka dalam berbicara di depan umum.



Gambar 4. Hasil jawaban responden tentang cara meningkatkan *public speaking* (a) dan keinginan mereka untuk mencoba lagi public speaking (b)

3.1.7 Hasil Analisa

Dari hasil wawancara dengan responden dapat dilihat bahwa aplikasi ini membutuhkan fitur login register akun, fitur feedback dari penonton, fitur untuk mengundang teman untuk masuk ke dalam sesi *live streaming video* dan harus memberikan UI/UX yang menarik karena yang akan menggunakannya adalah mahasiswa/mahasiswi.

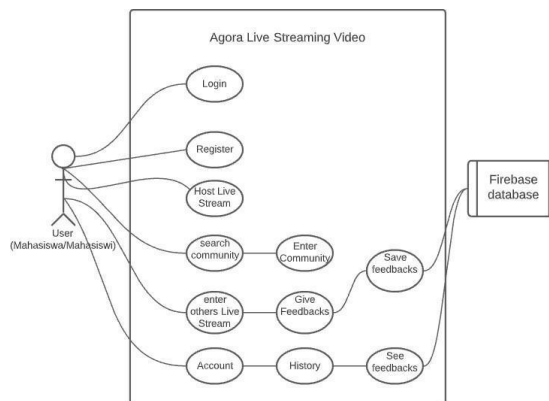
3.2 Desain Sistem

3.2.1 Desain UML

UML (*Unified Modeling Language*) adalah teknik pemodelan visual yang digunakan sebagai sarana untuk merancang sistem berorientasi objek (Juliarto, 2021). Awalnya UML dibuat oleh Object Management Group pada Januari 1997 dengan versi pertama 1.0. UML juga dapat digambarkan sebagai bahasa standar untuk memvisualisasikan, merancang, dan mendokumentasikan sistem, atau sebagai bahasa standar untuk membuat cetak biru perangkat lunak. Terdapat berbagai macam jenis UML yaitu use case diagram, activity diagram, sequence diagram, class diagram, state machine diagram, component diagram, dan deployment diagram.

3.2.1.1 Use case Diagram

Use case diagram adalah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor (Juliarto, 2021). *Use Case* dapat mendeskripsikan tipe interaksi antara pengguna sistem dengan sistemnya. Use case diagram artikel ini ditunjukkan pada Gambar 5.



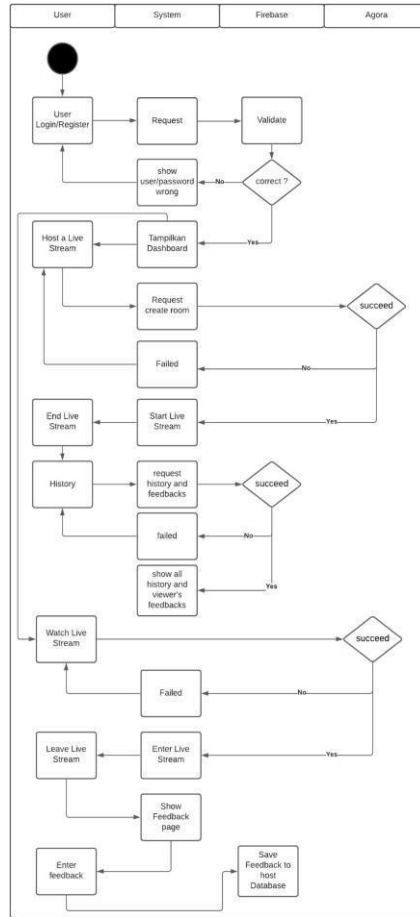
Gambar 5. Use Case Diagram

3.2.1.2 Activity Diagram

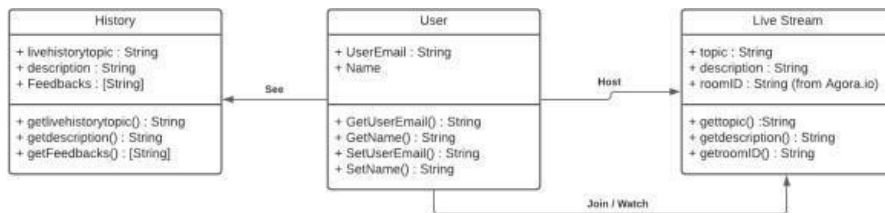
Activity diagram, yaitu diagram yang bisa memodelkan proses-proses yang terjadi dalam sebuah sistem (Juliarto, 2021). Runtutan proses berdasarkan suatu sistem digambarkan secara vertikal. Activity diagram adalah pengembangan berdasarkan use case yang mempunyai alur kegiatan. Alur atau kegiatan berupa mampu berupa runtutan hidangan-hidangan atau proses usaha yang masih ada pada sistem tersebut. Dalam kitab *Rekayasa Perangkat Lunak* karangan Rosa A.S mengatakan, “Diagram kegiatan nir mengungkapkan kelakuan aktor. Dapat diartikan bahwa pada pembuatan activity diagram hanya bisa digunakan buat mendeskripsikan alur kerja atau kegiatan sistem saja.” Activity diagram artikel ini dapat dilihat pada Gambar 6.

3.2.1.3 Class Diagram

Diagram kelas adalah jenis diagram struktural UML yang secara jelas menggambarkan struktur dan deskripsi kelas, atribut, metode, dan hubungan setiap objek (Juliarto, 2021). Diagram kelas bersifat statis dalam arti bahwa diagram tersebut menjelaskan hubungan apa yang terjadi, daripada menjelaskan apa yang terjadi ketika kelas terkait. Diagram kelas ini sangat mudah digunakan dan berguna untuk implementasi dalam proyek berorientasi objek. Diagram kelas artikel ini dapat dilihat pada Gambar 7.



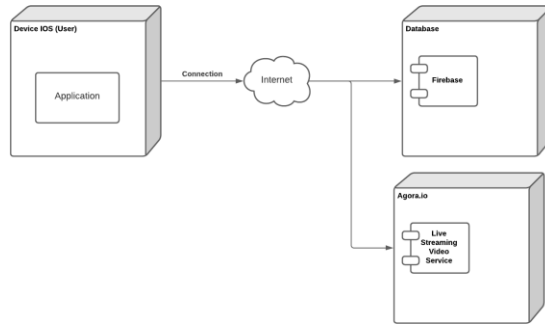
Gambar 6. Activity Diagram



Gambar 7. Class Diagram

3.2.1.4 Deployment Diagram

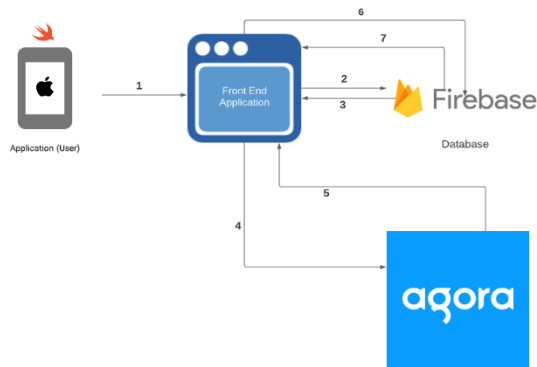
Deployment diagram adalah salah satu model diagram UML untuk menempatkan artefak pada sebuah node (Vinashaw, 2018). Layout digunakan untuk memvisualisasikan, menentukan, dan mendokumentasikan proses yang terjadi pada sistem perangkat lunak berorientasi objek yang sedang dibangun. Maksud atau fungsi dari layout adalah untuk menjelaskan/memvisualisasikan proses yang sedang berlangsung dalam sistem/software secara umum. Node UML mewakili lingkungan eksekusi perangkat keras atau perangkat lunak. Node dapat terhubung melalui jalur komunikasi untuk membuat sistem jaringan yang kompleks. Artefak UML mewakili spesifikasi bentuk fisik informasi yang digunakan atau dihasilkan selama proses pengembangan. Deployment diagram artikel ini ditunjukkan pada Gambar 8.



Gambar 8. Deployment Diagram

3.2.2 Desain Arsitektur

Pertama-tama user login/register akun lalu dapat menggunakan fitur *host livestream* untuk memulai proses *live streaming video* dan setelah selesai bisa melihat *history* dan *feedback* dari user lain. Selain itu, user juga bisa masuk ke dalam *livestream* user dan bisa memberikan *feedback* kepada user tersebut seperti pada Gambar 9.



Gambar 9. Desain Arsitektur

3.2.3 Desain Mockup

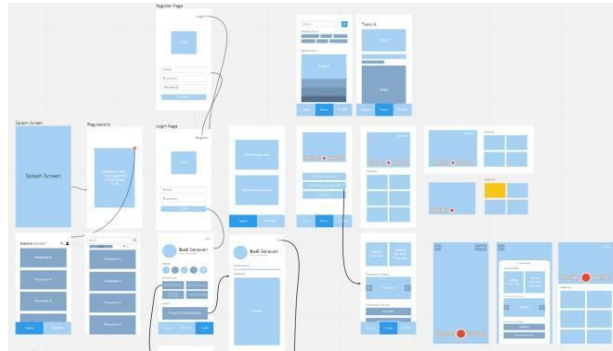
Desain aplikasi yang dibuat simple dan mudah digunakan agar semua orang dapat menggunakannya dan memiliki desain yang lumayan menarik karena yang akan menggunakannya adalah mahasiswa/mahasiswi. Berikut merupakan beberapa contoh desain mockup dari pengembangan aplikasi ini pada Gambar 10 dan Gambar 11.

4. Hasil dan Pembahasan

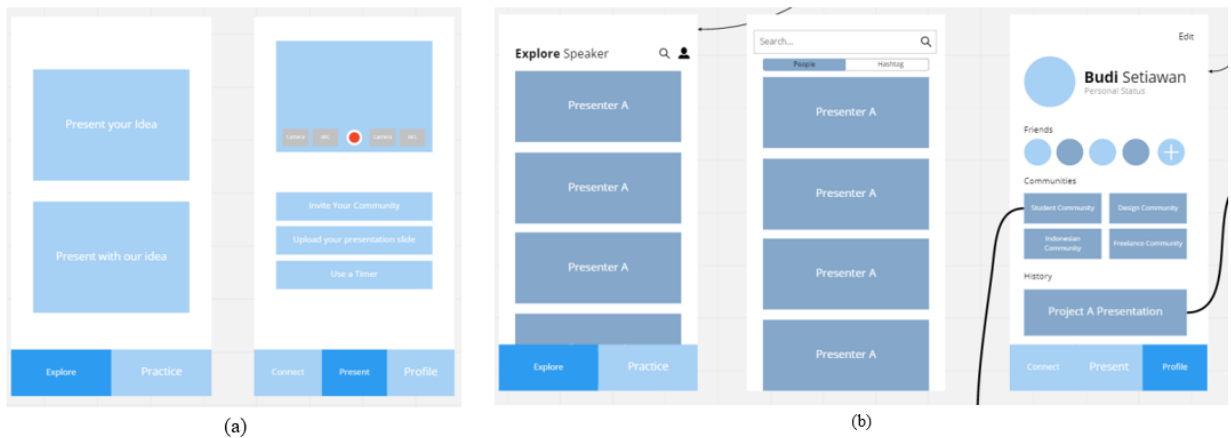
Bab ini menjelaskan mengenai hasil pengujian yang telah dilakukan pada sistem aplikasi *public speaking* yang merupakan aplikasi berbasis iOS. Hasil pengujian yang dilakukan untuk mengetahui apakah sistem aplikasi yang dirancang dapat berjalan dengan baik. Pengujian yang dilakukan adalah pengujian terdapat penerapan implementasi *live streaming* dari *Agora.io*, sistem penyimpanan data pengguna dan sistem otentikasi pengguna yang menggunakan sistem *Firebase*.

4.1 Hasil Pengujian

Pengujian aplikasi ini dilakukan untuk mencari bug atau permasalahan yang bisa saja muncul saat aplikasi sedang digunakan oleh pengguna, dan juga digunakan untuk mengetahui kekurangan dari aplikasi yang harus dikembangkan lebih jauh (Okriwina, 2021). Pada penelitian ini pengujian yang digunakan adalah pengujian *Alpha* dan *Beta*.



Gambar 10. Desain mockup aplikasi



Gambar 11. Cuplikan detail desain mockup aplikasi

4.1.1 Pengujian Alpha

Tes Alpha adalah tes yang dirancang untuk memastikan bahwa aplikasi yang diuji berjalan dengan lancar tanpa bug. Dalam penelitian ini, tes White Box dan Black Box digunakan sebagai tes Alpha.

4.1.1.1 Pengujian White Box

White box testing adalah pengujian perangkat lunak pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan (Pratala, et al., 2020). Terdapat beberapa kelebihan dalam penggunaan pengujian *white box* ini seperti:

1. Efisien dalam menemukan kesalahan dan masalah
2. Bermanfaat dalam pengujian penuh yang teratur
3. Dapat menemukan kesalahan-kesalahan kecil
4. Membantu mengoptimalkan kode program

Tetapi terdapat juga beberapa kekurangan dalam penggunaan *white box* yaitu:

1. Pada aplikasi yang besar pengujian ini akan boros karena membutuhkan sumber daya yang besar
2. Membutuhkan sebuah pengetahuan khusus mengenai perangkat lunak internal yang sedang diujikan

Pengujian *white box* dalam penelitian ini yaitu menangani kesulitan fitur *Host/Join Live Streaming*. Beberapa metode telah dicoba agar fitur ini dapat bekerja dengan baik, dengan menggunakan implementasi *Agora.io* dan implementasi *Firebase*. Penulis menggunakan *Agora.io* karena fitur yang diberikan cukup efisien untuk digunakan dalam aplikasi ini dan *Firebase* digunakan untuk menyimpan data pengguna saat masuk ke *live streaming* pengguna lain.

4.1.1.2 Pengujian Black Box

Pengujian Black Box atau pengujian operasional, adalah pengujian yang dilakukan untuk memantau input dan

output perangkat lunak tanpa mengetahui struktur kode perangkat lunak (Setiawan, 2021). Tes ini dijalankan pada akhir pembuatan perangkat lunak untuk memastikan bahwa perangkat lunak berfungsi dengan baik. Pengujian dapat dilihat pada Tabel 1. Terdapat beberapa kelebihan dalam penggunaan pengujian *black box* ini seperti:

1. Efisien untuk menguji aplikasi yang besar
2. Penguji tidak memerlukan akses kode

Tetapi terdapat juga beberapa kekurangan dalam penggunaan *black box* yaitu:

1. Penguji mempunyai keterbatasan dalam karena hanya sebagian kecil dari skenario yang dilakukan
2. Pengujian yang tidak efisien karena kurangnya pengetahuan *tester* mengenai kode

Tabel 1. Pengujian Black Box

No	Fungsi Aplikasi	Kondisi	Aksi yang diharapkan	Aksi yang terjadi	Status Pengujian
1	<i>Splashscreen</i>	Menampilkan logo aplikasi	Dapat menampilkan logo aplikasi dan pindah ke halaman <i>sign in</i> atau ke menu utama aplikasi	Dapat menampilkan logo aplikasi dan pindah ke halaman <i>sign in</i> atau ke menu utama aplikasi	Valid
2	<i>Register</i> aplikasi	Semua data pengguna yang dibutuhkan terisi	Pindah ke halaman menu utama aplikasi	Pindah ke halaman menu utama aplikasi	
3	<i>Register</i> aplikasi	Semua data pengguna yang dibutuhkan terisi	Pindah ke halaman menu utama aplikasi	Aplikasi tertutup paksa dan pada Xcode terdapat error	Tidak Valid
4	<i>Login</i> aplikasi	Email dan password benar	Pindah ke halaman menu utama aplikasi	Pindah ke halaman menu utama aplikasi	Valid
5	<i>Host Live Streaming</i>	Terdapat Token dan channel yang tersedia	Pindah ke halaman <i>live streaming</i> dan muncul video	Pindah ke halaman <i>live streaming</i> tetapi video tidak muncul	Tidak Valid
6	<i>Host Live Streaming</i>	Terdapat Token dan channel yang tersedia	Pindah ke halaman <i>live streaming</i> dan muncul video	Pindah ke halaman <i>live streaming</i> dan muncul video	Valid
7	<i>Join Live Streaming</i>	Terdapat Token dan channel yang tersedia	Pindah ke halaman <i>live streaming</i> dan muncul video	Pindah ke halaman <i>live streaming</i> dan muncul video	Valid
8	<i>History Feedback</i>	Mengambil data <i>history</i> dari <i>database</i>	Menampilkan <i>history</i> dan <i>feedback</i> dari hasil <i>live</i>	Menampilkan <i>history</i> dan <i>feedback</i> dari hasil <i>live</i>	Valid

4.1.2 Pengujian Beta

Pengujian beta adalah serangkaian pengujian penerimaan pengguna yang dijalankan sebelum produk dirilis ke publik (Beta Test, 2020). Tujuan dari pengujian ini adalah untuk menemukan dan mengidentifikasi sebanyak mungkin kesalahan atau masalah dalam penggunaan sistem dan produk Anda. Pengujian beta melibatkan sejumlah besar pengguna, yang disebut penguji beta.

Pada penelitian ini jenis pengujian beta yang digunakan adalah close beta testing yang hanya melibatkan orang tertentu saja dengan melakukan wawancara dan pengujian sistem ke tiga target pengguna (Evelyn Wijaya, Enrico

Christopher Rano, dan Jackson Thungadi) dengan mendapatkan hasil pengujian serta mendapatkan saran terdapat aplikasi yang telah dikembangkan.

4.2 Pembahasan

4.2.1 Pengujian Alpha

4.2.1.1 Pengujian White Box

Pengujian *white box* untuk fitur *Host/Join Live Streaming* ini memiliki tingkat keberhasilan 8/10 dimana fitur ini mengalami kesalahan saat akan masuk ke halaman *live streaming* seperti, tidak munculnya video atau pengguna lain tidak bisa masuk ke *live streaming* yang sama dengan *host*.

4.2.1.2 Pengujian Black Box

Hasil pengujian Black Box dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Black Box

No	Fungsi Aplikasi	Kondisi	Aksi yang diharapkan	Aksi yang terjadi	Status Pengujian
1	<i>Splashscreen</i>	Menampilkan logo aplikasi	Dapat menampilkan logo aplikasi dan pindah ke halaman <i>sign in</i> atau ke menu utama aplikasi	Dapat menampilkan logo aplikasi dan pindah ke halaman <i>sign in</i> atau ke menu utama aplikasi	Berhasil 10/10 percobaan
2	<i>Register</i> aplikasi	Semua data pengguna yang dibutuhkan terisi	Pindah ke halaman menu utama aplikasi	Pindah ke halaman menu utama aplikasi	Berhasil 8/10 percobaan
3	<i>Login</i> aplikasi	Email dan password benar	Pindah ke halaman menu utama aplikasi	Pindah ke halaman menu utama aplikasi	Berhasil 10/10 percobaan
4	<i>Host Live Streaming</i>	Terdapat Token dan channel yang tersedia	Pindah ke halaman <i>live streaming</i> dan muncul video	Pindah ke halaman <i>live streaming</i> dan muncul video	Berhasil 6/10 percobaan
5	<i>Join Live Streaming</i>	Terdapat Token dan channel yang tersedia	Pindah ke halaman <i>live streaming</i> dan muncul video	Pindah ke halaman <i>live streaming</i> dan muncul video	Berhasil 10/10 percobaan
6	<i>History Feedback</i>	Mengambil data <i>history</i> dari <i>database</i>	Menampilkan <i>history</i> dan <i>feedback</i> dari hasil <i>live</i>	Menampilkan <i>history</i> dan <i>feedback</i> dari hasil <i>live</i>	Berhasil 10/10 percobaan

4.2.2 Pengujian Beta

Dari penelitian beta, mendapatkan beberapa hasil dan komentar dari para *tester*. Hasil yang penulis dapatkan

adalah saat pengguna terlalu cepat menekan tombol berhenti pada halaman *live streaming* maka *live streaming* akan terus berjalan di *background* sampai aplikasi ditutup dan mendapatkan beberapa saran yang dapat digunakan untuk perkembangan aplikasi ini selanjutnya.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab-bab sebelumnya, dapat disimpulkan bahwa penelitian ini telah mampu menjawab penerapan rancang bangun sistem melatih *skill public speaking* yang menggunakan *third party Agora.io* dan menggunakan *Firebase* sebagai penyimpanan data pengguna dimana:

1. Penggunaan *platform Agora.io* dapat memberikan hasil *live streaming* dengan kualitas yang bagus dan mudah digunakan oleh pengguna aplikasi
2. Dalam penggunaan aplikasi, pengguna harus *register/login* akun yang akan disimpan dalam *Firebase*, agar pengguna dapat menggunakan aplikasi dengan keseluruhan.
3. Saat seorang pengguna melakukan *Live Streaming*, pengguna lain bisa masuk ke *live streaming* yang sama dan memberikan *feedback* yang dapat dilihat oleh pengguna menggunakan fitur *history*.

5.2 Saran

Berikut merupakan saran-saran yang diberikan dan dapat digunakan untuk memperbaiki ataupun mengembangkan aplikasi atau sistem selanjutnya. Saran yang diberikan sebagai berikut:

1. Menambahkan fitur yang dapat mengubah *profile picture*
2. Pada fitur *live streaming*, penempatan video sebaiknya diperbaiki
3. Tampilan UI yang masih bisa dikembangkan lebih baik lagi baik dari segi penempatan dan warna tampilan.

Daftar Pustaka

- Apple. (n.d.). *Swift*. Retrieved from Apple: <https://www.apple.com/id/swift/>
- Beta Test. (2020). Retrieved from Product Plan: <https://www.productplan.com/glossary/beta-test/>
- Fadli, R. (2020, 04 17). *Takut Bicara di Depan Orang Banyak? Mungkin Ini Alasannya*. Retrieved from halodoc: <https://www.halodoc.com/artikel/takut-bicara-di-depan-orang-banyak-mungkin-ini-alasannya>
- Firebase. (2022). *Firebase Realtime Database | Firebase Documentation*. Retrieved from Firebase: <https://firebase.google.com/docs/database?hl=id>
- Ham, H. (2019, 12 26). *Swift is Fun*. Retrieved from Binus: <https://socs.binus.ac.id/2019/12/26/swift-is-fun/>
- Juliarto, R. (2021, 03 10). *Apa itu Activity Diagram? Beserta Pengertian, Tujuan, Komponen*. Retrieved from Dicoding: <https://www.dicoding.com/blog/apa-itu-activity-diagram/>
- Juliarto, R. (2021, 05 12). *Apa itu UML? Beserta Pengertian dan Contohnya*. Retrieved from Dicoding: <https://www.dicoding.com/blog/apa-itu-uml/>
- Nosuke. (2020, 12 16). *Xcode Tutorial : Aplikasi Terbaik untuk Ngoding iOS di Mac*. Retrieved from Appkey: <https://appkey.id/pembuatan-aplikasi/aplikasi-ios/xcode-tutorial/>
- Oktriwina, A. S. (2021, 01 31). *Mengenal Alpha Testing, Tahap Pengujian Pertama sebelum Aplikasi Diluncurkan*. Retrieved from glints: <https://glints.com/id/lowongan/alpha-testing/#.YpAY5KhByUk>
- Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020). Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *J. Inform. Univ. Pamulang*, 5(2), 111.
- Rodríguez-Gil, L., García-Zubia, J., Orduna, P., & Lopez-de-Ipiña, D. (2017). An open and scalable web-based interactive live-streaming architecture: The WILSP platform. *IEEE Access*, 5, 9842-9856.
- Setiawan, R. (2021, 11 17). *Black Box Testing Untuk Menguji Perangkat Lunak*. Retrieved from Dicoding: <https://www.dicoding.com/blog/black-box-testing/>
- Shang, Z., Ebenezer, J. P., Wu, Y., Wei, H., Sethuraman, S., & Bovik, A. C. (2021). Study of the subjective and objective quality of high motion live streaming videos. *IEEE Transactions on Image Processing*, 31, 1027-1041.
- Vinashaw. (2018). *Apa yang dimaksud dengan Deployment diagram?* Retrieved from Dictio:

<https://www.dictio.id/t/apa-yang-dimaksud-dengan-deployment-diagram/15125>

Wang, Z., Cui, Y., Hu, X., Wang, X., Ooi, W. T., Cao, Z., & Li, Y. (2021). Multilive: Adaptive bitrate control for low-delay multi-party interactive live streaming. *IEEE/ACM Transactions on Networking*, 30(2), 923-938.