

## **REAL TIME ONLINE EXAM PROCTORING SYSTEM IN HIGHER EDUCATION USING WEBRTC TECHNOLOGY**

Jeems Terri Agustinus<sup>1</sup>, Mychael Maoeretz Engel<sup>\*2</sup>

<sup>1,2</sup>Informatics, School of Information Technology, Universitas Ciputra Surabaya, Indonesia  
Email: [jagustinus@student.ciputra.ac.id](mailto:jagustinus@student.ciputra.ac.id), [mychael.engel@ciputra.ac.id](mailto:mychael.engel@ciputra.ac.id)

(Article received: December 04, 2023; Revision: January 03, 2024; published: January 06, 2024)

### **Abstract**

*The low level of trust in online exam results from students is a major problem because it is difficult to monitor whether test takers are taking the exam honestly according to their own abilities. Even though it has been assisted by the presence of video conferencing applications such as Zoom, Google Meet, Cisco Webex and similar applications, online exam proctoring is still unable to run effectively. Cheating in online exams, such as using dual monitors, is very possible for exam participants. Therefore, as a future preventive measure in the online exam process, a system is needed that can accommodate this concern. This research will create an online exam supervision system with WebRTC technology which has features to accommodate real-time supervision. The System Development Life Cycle method will be used in software development with 5 main stages, namely Requirement Analysis, Design, Development, Testing, and Maintenance. Implementation of the system was carried out during the online examination process for a class at one of the universities in Surabaya. Finally, the test results show that features such as: Live Proctoring get a score of 4.5; Attention Alert gets a score of 5; Exam Lock scored 4.5; Live Alert scored 4.5; and Tab & Window Detection got a score of 4; shows that this system has succeeded in providing a solution in online exam proctoring needs.*

**Keywords:** *live proctoring, online exam, real-time, webrtc.*

## **SISTEM PENGAWASAN UJIAN ONLINE SECARA REAL TIME PADA PERGURUAN TINGGI MENGGUNAKAN TEKNOLOGI WEBRTC**

### **Abstrak**

Rendahnya tingkat kepercayaan terhadap hasil ujian secara online dari mahasiswa menjadi masalah utama karena kesulitan mengawasi apakah para peserta ujian mengerjakan ujian secara jujur dengan kemampuan sendiri. Meski telah dibantu dengan kehadiran aplikasi konferensi video seperti Zoom, Google Meet, Cisco Webex dan sejenisnya, pengawasan ujian online tetap tidak mampu berjalan secara efektif. Tindak kecurangan dalam ujian online seperti penggunaan dual monitor sangatlah mungkin dilakukan oleh para peserta ujian. Oleh karena itu, sebagai tindakan preventif kedepan dalam proses ujian online, dibutuhkan suatu sistem yang dapat mengakomodasi keresahan ini. Penelitian ini akan menghasilkan sebuah sistem pengawasan ujian online dengan teknologi WebRTC yang memiliki fitur-fitur untuk mengakomodasi pengawasan secara real-time. Metode System Development Life Cycle akan digunakan pada pengembangan perangkat lunak dengan 5 tahapan utama yaitu Requirement Analysis, Design, Development, Testing, dan Maintenance. Pengimplementasian sistem dilakukan pada proses ujian online suatu kelas yang terdapat pada salah satu Universitas di Surabaya. Akhirnya, hasil pengujian menunjukkan bahwa fitur seperti: Live Proctoring mendapatkan skor 4,5; Attention Alert mendapatkan skor 5; Exam Lock mendapatkan skor 4,5; Live Alert mendapatkan skor 4,5; dan Tab & Window Detection mendapatkan skor 4; menunjukkan bahwa sistem ini telah berhasil memberikan solusi dalam memenuhi kebutuhan pengawasan ujian online.

**Kata kunci:** *pengawasan langsung, real-time, ujian online, webrtc.*

### **1. PENDAHULUAN**

Dalam menanggapi pandemi COVID-19, penggunaan sistem ujian yang diawasi secara *online* semakin populer [1][2]. Berkembangnya ujian *online* untuk mengevaluasi kinerja siswa menimbulkan

ancaman yang lebih besar terhadap integritas akademis daripada ujian *onsite* karena diduga dengan adanya penggunaan internet, lebih banyak kesempatan bagi siswa untuk melakukan kecurangan dalam ujian *online* [3][4]. Selain itu, siswa lebih

cenderung bertindak tidak jujur saat belajar dalam situasi *online* karena tindakan curang lebih umum terjadi dalam tes *online* tanpa rekaman (dan lebih jarang dalam ujian daring yang diawasi) [5]. Hal ini menunjukkan betapa pentingnya menjaga integritas akademik dengan memantau dan mengawasi ujian *online* dengan baik. Namun demikian, ada kekhawatiran tentang kemungkinan dampak negatif dari ujian yang diawasi. Kekhawatiran ini mencakup kecemasan yang meningkat terkait dengan ujian, persepsi kesulitan ujian, dan hasil performa [6].

Meskipun masih dalam debat dan perilaku curang cenderung meluas ketika tanpa adanya peringatan dalam pengawasan, mungkin saja perilaku curang dapat dikurangi secara substansial dengan metode yang lebih tidak invasif, mahal, atau rumit [7]. Sehingga solusi untuk melakukan pengawasan yang lebih ketat dengan mewajibkan peserta ujian untuk menyalakan *webcam* selama ujian dianggap sebagai jawaban paling tepat [8]. Salah satu hasil dari pergeseran cepat COVID-19 ke pembelajaran daring, beberapa instansi berusaha dengan cepat mengadopsi *video conferencing applications* (VCA), seperti *Zoom* untuk mengisi celah sebagai layanan *proctoring* ujian yang dapat diandalkan, efisien dan terjangkau [9].

Aplikasi konferensi video seperti *Zoom*, *Google Meet*, dan *Cisco Webex* memiliki batasan dalam menjalankan pengawasan ujian daring yang efisien. Seperti yang dialami oleh dua dosen di salah satu Universitas di Surabaya. Rendahnya tingkat kepercayaan dosen terhadap hasil ujian dari mahasiswa menjadi masalah utama. Karena sulitnya dosen dalam mengawasi apakah para peserta ujian mengerjakan ujian secara jujur dan dengan kemampuan sendiri. Hal ini diprediksi dapat menyebabkan menurunnya kualitas kompetensi dari mahasiswa yang berimbas terhadap kualitas lulusan. Selain itu aplikasi komersial lainnya berfokus pada deteksi kecurangan, dimana tindakan kecurangan sudah / sedang terjadi. Oleh karena itu sebagai tindakan preventif kedepannya, dibutuhkan suatu sistem yang mampu mengakomodasi keresahan ini.

Berbeda dengan aplikasi konferensi video, layanan *proctoring online* seperti *Examity 22*, *Proctortrack TM23*, *Proctorio*, *Honorlock*, dan *ProctorU*, serta produk lokal *e-exam.com* menawarkan beberapa fitur yang bertujuan untuk mencegah kecurangan saat ujian *online* [10]. *Proctorio*, misalnya, menyediakan tutorial berbasis AI, yang mencakup fitur-fitur seperti mengunci *browser* dan merekam video dan audio untuk memantau kemajuan ujian. *Honorlock* menawarkan fitur deteksi multi-perangkat yang mendeteksi jika siswa menggunakan perangkat lain selama ujian. *ProctorU*, di sisi lain, menawarkan pengawasan langsung, di mana pengawas langsung memantau ujian melalui *webcam* dan *sharing* layar. *E-exam.com* menyediakan fitur-fitur seperti peringatan perhatian, peringatan langsung, dan deteksi tab dan jendela aktif.

Terlepas dari fitur pencegahannya, layanan pengawasan *online* ini juga memiliki keterbatasan. Misalnya, *Proctorio* mendapatkan kritik karena masalah privasi terkait pengumpulan data, dan *Honorlock* dituduh menggunakan teknologi pengenalan wajah tanpa persetujuan. Selain itu, beberapa siswa mungkin menganggap proses pemantauan mengganggu, menyebabkan kurangnya kepercayaan antara siswa dan institusi. Oleh karena itu, sebagai tindakan pencegahan untuk ujian *online*, disarankan untuk menyertakan fitur seperti peringatan perhatian, peringatan langsung, tab deteksi dan jendela aktif, dan mengunci ujian untuk memastikan siswa tetap fokus selama ujian dan untuk mencegah mereka mengakses materi yang dilarang.

Banyak penelitian yang telah dilakukan dalam beberapa tahun terakhir mengenai penggunaan *WebRTC* dan performanya. Salah satu penelitian terkait, menggunakan *WebRTC* untuk mengembangkan aplikasi ujian *online* berbasis *web* yang dapat diawasi oleh pengawas ujian secara langsung di SMA PGRI Bayah. Dengan mengembangkan fitur *video call* yang dapat digunakan saat siswa mengerjakan ujian, hal ini memungkinkan pengawas dapat melihat tangkapan gambar dari *webcam* siswa. Namun fitur *video call* ini tidak begitu perlu melihat bahwa tangkapan *webcam* pengawas juga tampil di halaman siswa [11].

Penelitian lainnya yaitu dengan judul "Rancang Bangun Aplikasi *E-meeting* Menggunakan *WebRTC* (Web Real time Communication)" dilakukan oleh Rizki Ainanda & Edy Victor Haryanto pada tahun 2020. Pada penelitian tersebut, arsitektur koneksi *WebRTC* secara *peer to peer* digunakan agar dapat melakukan *online meeting* langsung melalui *web browser*. Pengguna tidak perlu lagi menginstall *software* spesifik atau plugin sehingga penggunaan menjadi lebih praktis. Namun penggunaan aplikasi *online meeting* ini masih terbatas, karena 1 ruangan (*room*) *meeting* hanya dapat diisi oleh 2 pengguna. Hal ini disebabkan karena arsitektur koneksi *WebRTC* yang digunakan hanya secara *peer to peer* [12].

Pada penelitian lainnya dengan judul "Implementasi *Video Conference* dengan *File Sharing* menggunakan *WebRTC*" dilakukan pada tahun 2019 oleh Fathi Nur Azzam, dkk. Penelitian tersebut berfokus untuk mengembangkan aplikasi *Video Conference* dengan *File sharing* menggunakan teknologi *WebRTC*. Fokus utama penelitian ini adalah melihat hubungan antara fitur *file sharing* dengan *video conference*. Untuk dapat melewati *NAT* secara *peer to peer*, dibutuhkan 2 *server* tambahan yaitu *STUN* dan *TURN*. Kemudian dilakukan pengujian dengan menggunakan 6 tipe koneksi dan 3 jenis ukuran file. Hasil yang didapat adalah fitur *file sharing* tidak berpengaruh terhadap kualitas *video conference* karena keduanya berjalan pada *port* yang berbeda. Namun, penggunaan sumber daya komputer

meningkat pada *memory* dan secara signifikan pada *processor* [13].

Pada penelitian lainnya dengan judul "*Psychological Support Application Using The Real-Time Webrtc: A Case Study Of Undergraduate In Malaysian And Indonesian Higher Learning Institutions*" dilakukan pada tahun 2021 oleh Mohamad Asrol Arshad, dkk. Penelitian ini membahas soal pengembangan aplikasi dukungan psikologis menggunakan *WebRTC* untuk membantu mahasiswa menghadapi masalah kesehatan mental selama pandemi Covid-19. Dengan menggunakan *WebRTC*, mahasiswa dapat melakukan panggilan secara *anonymous* ke mahasiswa lainnya. Hasil dari penelitian ini menyatakan bahwa aplikasi yang dikembangkan berhasil menjadi solusi masalah mahasiswa dalam menyediakan *platform* berbagi dan konsultasi kesehatan mental mereka [14].

Hasil sebuah studi yang dilakukan Jung Wang Lee dalam mengidentifikasi dampak lingkungan pengawasan terhadap kinerja siswa dalam ujian yang diawasi secara *online* dibandingkan dengan ujian yang dilakukan secara *offline* memberikan wawasan tentang bagaimana berbagai format pengawasan dapat digunakan untuk melakukan penilaian dalam pendidikan *online* [15]. Layanan pengawasan *online* dan *real-time* semakin populer, menunjukkan bahwa ada permintaan yang meningkat untuk produk seperti itu di dunia pendidikan.

Sehingga, berdasarkan permasalahan yang telah dijabarkan di atas, maka tujuan dari penelitian ini adalah membangun sebuah *website* pengawasan ujian *online* secara *real time* menggunakan teknologi *WebRTC* dengan menerapkan fitur-fitur yang dapat

menjadi tindakan preventif kecurangan pada ujian *online*.

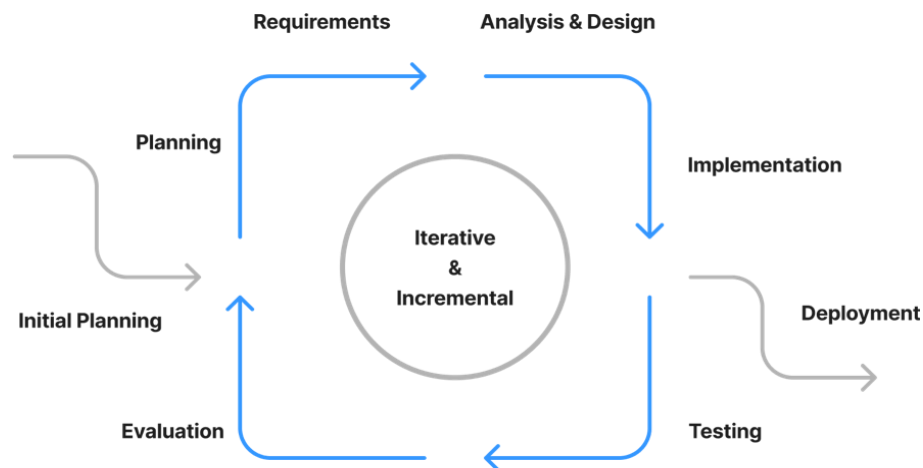
## 2. METODE PENELITIAN

Pada bagian ini akan menjelaskan metode pengembangan *software* yang digunakan, analisa kebutuhan, pengumpulan data, desain arsitektur, *use case diagram* dan *class diagram* dari sistem.

### 2.1. Metode yang Digunakan

Metode *Software Development Life Cycle* (SDLC) merupakan sebuah proses membuat dan memodifikasi sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem rekayasa perangkat lunak. Metode *SDLC* hadir untuk membantu dalam proses pengembangan suatu produk. *SDLC* dapat disebut sebagai model klasik yang sifatnya sistematis dan berurutan dalam membangun perangkat lunak. Fungsi dari *SDLC* adalah mengakomodasi setiap kebutuhan pengguna yang berkaitan dengan sistem yang ada. Kebutuhan pengembangan sistem tersebut dapat dilihat dari perubahan atau penciptaan aplikasi baru. Selain hal tersebut melalui *SDLC* pengembang perangkat lunak dapat memperkirakan umur sebuah perangkat lunak diciptakan ataupun digunakan.

Dalam penelitian ini terdapat beberapa tahapan dari metode *SDLC* sebagai berikut: (1) Requirement Analysis, (2) Design, (3) Development, (4) Testing, dan (5) Maintenance seperti pada gambar Gambar 1 di bawah ini.



Gambar 1. Skema SDLC

Tahap *requirement Analysis* merupakan tahap saat mengumpulkan kumpulan informasi yang bertujuan untuk memecahkan suatu permasalahan. Setiap informasi yang didapatkan akan di pelajari dan di-*extract* untuk mendapatkan hasil analisis kebutuhan. Tahap *design* merupakan tahap rancangan desain aplikasi yang berlandaskan hasil analisis dari tahap sebelumnya. Pada tahap ini akan menggambarkan secara detail terkait pola-pola

desain. Tahap *development* merupakan tahap yang panjang dalam metode *SDLC* karena tahap ini merupakan tahap implementasi desain *High-Fidelity* ke *coding*.

Proses *development* untuk aplikasi ini menggunakan *tools github* untuk repositori pembuatan aplikasi, serta menggunakan *framework* *Laravel* sebagai bahasa implementasi *coding*. Lebih jelasnya, tahap ini merupakan tahap pembuatan

aplikasi dengan desain yang sama dengan tahap sebelumnya. Tahap *testing* merupakan tahap untuk menguji apakah aplikasi yang sudah dibuat merupakan sistem yang sudah layak digunakan. Tahap terakhir yaitu tahap pemeliharaan/*maintenance* termasuk dalam tahap memperbaiki kesalahan yang ditemukan ketika melakukan testing di tahap sebelumnya.

## 2.2. Analisa Kebutuhan

Berdasarkan permasalahan yang dialami karena ketidakpuasan pada validitas ujian *online* yang dilakukan selama ini, dapat disimpulkan bahwa dalam melakukan ujian *online*, cukup sulit untuk melakukan pengawasan terhadap peserta ujian. Berdasarkan hasil wawancara, dibutuhkan sistem yang dapat membantu melakukan pengawasan ujian *online* karena selama ini masih menggunakan aplikasi *video conference* namun masih belum maksimal. Diharapkan sistem yang dikembangkan oleh peneliti dapat melakukan pengawasan menggunakan webcam peserta dan layar peserta sehingga pengawas dapat

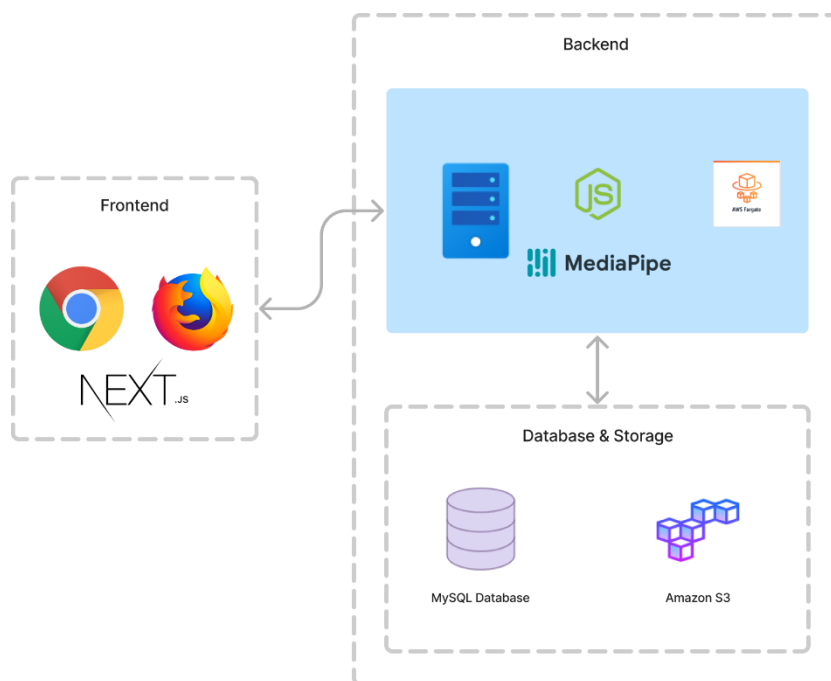
melihat gerak gerak serta tampilan layar peserta. Kemudian pengawas juga membutuhkan kewenangan untuk melakukan peringatan maupun mendiskualifikasi peserta. Untuk mengidentifikasi kebutuhan agar dapat selaras dengan permasalahan yang ada, maka dilakukan analisis hasil dari wawancara yang sudah dilakukan.

## 2.3. Pengumpulan Data

Proses pengumpulan data yang digunakan selain studi literatur adalah menggunakan teknik atau metode wawancara. Beberapa diantaranya adalah dosen pada sebuah Universitas di Surabaya yang pernah melakukan ujian secara *online*. Peneliti telah menyusun kumpulan pertanyaan yang rinci mengenai masalah, proses ujian *online*, jenis kecurangan, hingga fitur apa saja yang diinginkan.

## 2.4. Desain Arsitektur Sistem

Arsitektur dari sistem yang dikembangkan dapat dilihat pada Gambar 2 di bawah ini.



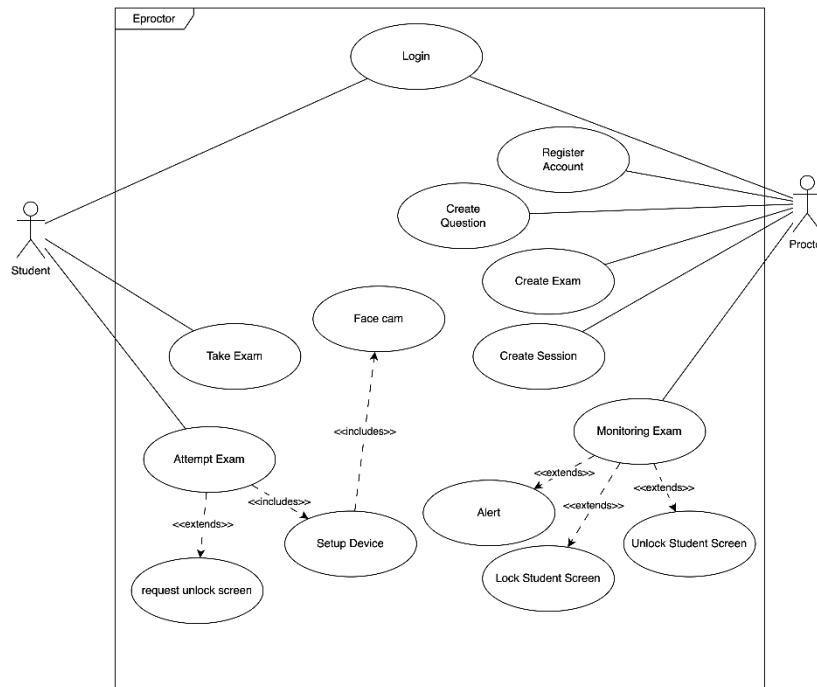
Gambar 2. Desain arsitektur sistem *online proctoring*

Peserta ujian akan menggunakan kamera atau *webcam* pada bagian frontend untuk input video yang nantinya akan di proses oleh sistem untuk dikirimkan ke *client proctor*. Sistem juga akan menyimpan data stream tersebut ke dalam *storage S3*. *Attention score* peserta juga akan digunakan untuk memberikan peringatan kepada peserta dan *proctor*. Apabila peserta melakukan penambahan tab atau pindah *window*, maka sistem *frontend* akan mengirimkan pesan ke *backend* dan menampilkan data kepada pengawas bahwa telah terjadi penambahan tab dan pemindahan *window*. Pengawas atau tenaga pendidik dapat melihat report dari hasil pengawasan yang

dilakukan oleh sistem. Data akan disimpan ke dalam *database MySQL / MariaDB*.

## 2.5. Use Case Diagram Sistem

*Use case diagram* adalah gambaran interaksi yang terjadi oleh pengguna atau *stakeholder* dengan sistem. Pada sistem pengawasan ujian *online* ini, terdapat 2 fitur besar yaitu: sistem ujian dan sistem pengawasan. Masing-masing memiliki *micro-activities* yang akan dijelaskan sesuai dengan Gambar 3 di bawah ini.

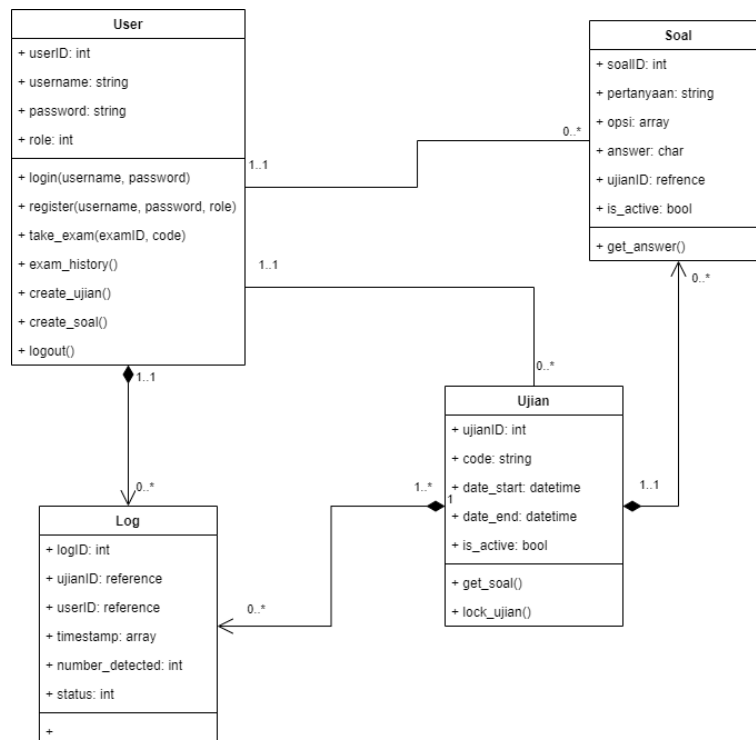


Gambar 3. Use case diagram sistem online proctoring

Terdapat dua tipe pengguna yaitu: peserta ujian dan pengawas / tenaga pendidik. Pengawas dapat membuat ujian dengan soal ujian sederhana yang nantinya dapat diakses oleh peserta ujian. Peserta ujian akan menggunakan kode ujian untuk dapat masuk ke dalam ujian dan mulai mengerjakan. Terdapat beberapa tahapan yang harus dilalui oleh peserta ujian seperti: pengenalan wajah, akses webcam, akses share screen, dan instruksi aturan selama ujian.

### 2.6. Class Diagram Sistem

Class diagram digunakan untuk menggambarkan struktur dari sistem dengan membuat permodelan dari class, attribute, metode, dan hubungan antar object. Desain class diagram dari sistem online proctoring yang dikembangkan dapat dilihat pada Gambar 4 di bawah ini.



Gambar 4. Class diagram sistem online proctoring

### 3. HASIL DAN PEMBAHASAN

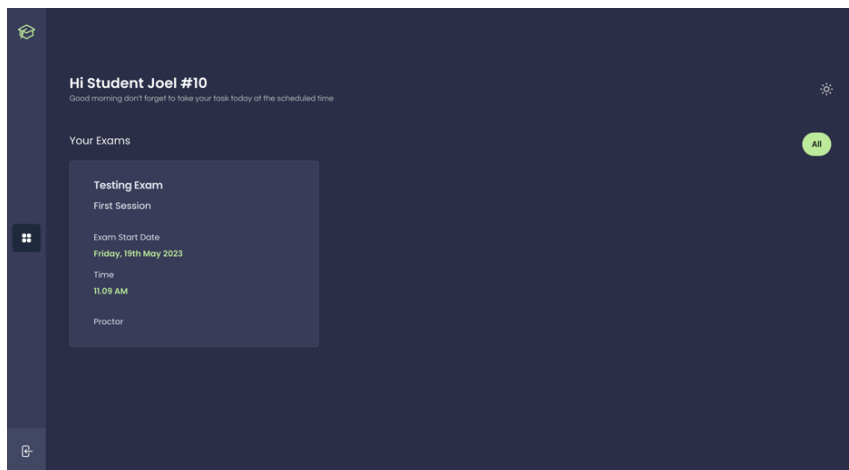
Terdapat 2 bagian pada sistem yang dibangun pada penelitian ini, yaitu sistem *backend* yang dimana bertindak sebagai sistem yang berhubungan dengan database dan menyediakan *API* yang akan digunakan oleh sistem *frontend* yang akan berhubungan langsung dengan pengguna. Sistem yang dibangun pada penelitian ini dapat diakses melalui website oleh pengguna yang nantinya dibagi menjadi 2 pengguna yaitu pengguna *Student* dan *Proctor*. Kedua pengguna tersebut akan memiliki *dashboard* masing-masing yang memiliki fitur sesuai dengan *role* masing-masing pengguna. Fokus fitur utama akan berada pada halaman pengawasan ujian yang dapat diakses oleh *Proctor* dan halaman ujian yang akan diakses oleh *Student*.

Karena *scope* dari sistem ini berfokus pada pengawasan ujian, maka dibutuhkan sistem ujian sehingga sistem ini juga menyediakan sistem ujian sederhana yang dapat mendukung ujian dengan tipe pilihan ganda. Sehingga, pembahasan implementasi akan berfokus pada bagian pengguna *student* yang berlaku sebagai pengambil ujian.

#### 3.1. Implementasi Fitur Sistem

##### 3.1.1. Implementasi *Dashboard Student*

Pada implementasi halaman *dashboard* ini, pengguna dengan *role student* akan dapat melihat informasi mengenai sesi ujian yang dimiliki dan akan berlangsung. Seperti pada Gambar 5 di bawah, pengguna dengan *role student* akan memiliki tampilan sesuai dengan fitur yang dapat diakses

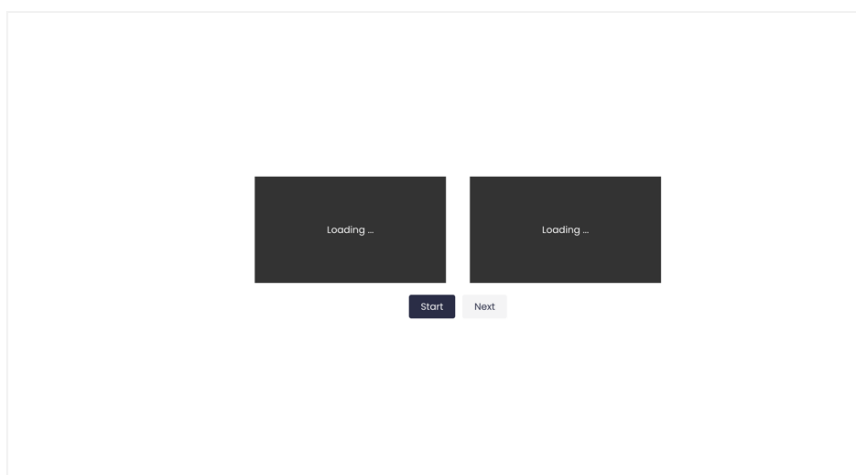


Gambar 5. Halaman *Dashboard Student*

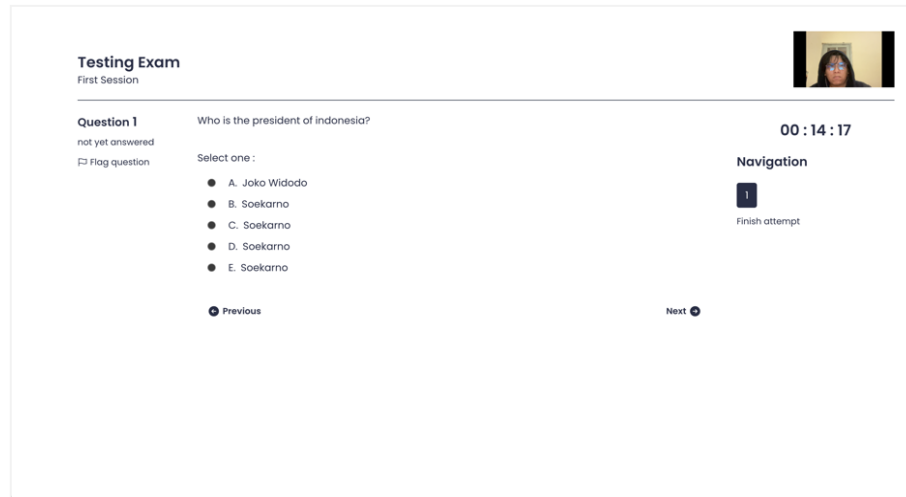
Pengguna dengan *role student* hanya memiliki satu fitur utama yaitu mengerjakan ujian. Sehingga pada halaman dashboard pengguna dengan *role student*, informasi yang akan ditampilkan hanyalah *list* ujian yang dimiliki dan yang akan berlangsung.

##### 3.1.2. Implementasi *Student Attempt Exam*

Pada implementasi fitur *attempt exam* ini, pengguna dengan *role student* akan dapat menggunakan fitur ini untuk mengerjakan ujian dan secara *real time* diawasi oleh *proctor* seperti pada Gambar 6 dan 7 di bawah ini.



Gambar 6. Halaman *setup device student*

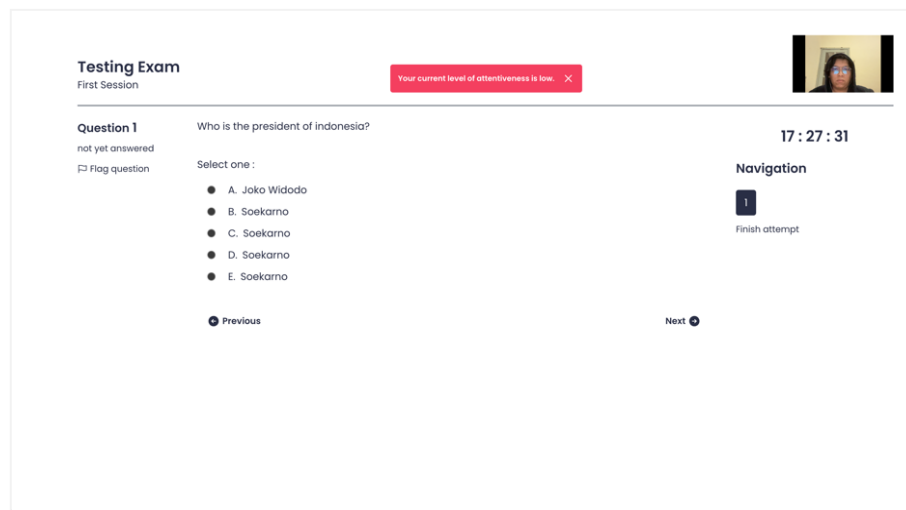
Gambar 7. Halaman ujian pada *student*

*Student* terlebih dahulu melakukan *setup device* untuk mengaktifkan *webcam* dan *share screen* dari layar. Apabila keduanya sudah aktif, maka *student* akan dapat melanjutkan masuk ke halaman pengerjaan dengan menggunakan tombol *next*. Setelah itu, *student* dapat melihat soal, opsi pilihan, *preview webcam student* dan navigasi soal. *Student* dapat memilih opsi pilihan dan *frontend* akan mengirimkan jawaban ke *backend* agar dapat disimpan ke dalam *database*. Pada fitur navigasi, *student* dapat melakukan navigasi secara bebas. *Student* juga dapat melihat *preview* dari *webcam* mereka sendiri.

Pada tahap berikutnya, Fitur *WebRTC* sendiri berfungsi sebagai *media streaming webcam* dan *share screen student* yang akan dikirimkan ke *proctor*. Penggunaan *WebRTC* dapat dilihat pada halaman ujian *student*, dimana *video stream webcam* dan *share screen student* akan dikirimkan secara *real time* ke *backend* dan akan diteruskan ke *proctor*.

### 3.1.3. Implementasi Fitur *Alert*

Pada implementasi fitur *alert* ini, pengguna dengan *role student* dapat melihat *alert* yang dikirimkan oleh *proctor* saat ujian seperti pada Gambar 8 di bawah ini.

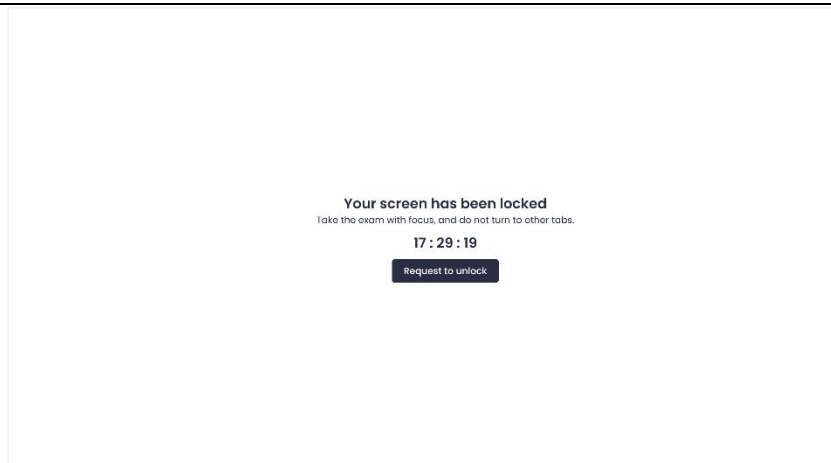
Gambar 8. Implementasi fitur *alert* pada *student*

Penanda *alert* akan muncul pada bagian atas *exam student*, sehingga *student* dapat melihat *alert* dari *proctor* mengenai apa yang harus diperhatikan atau diperbaiki.

dikunci dari ujian tersebut apabila ada perintah dari *proctor* dan skor atensi rendah. Tampilan penguncian ujian dapat dilihat seperti pada Gambar 9 di bawah ini.

### 3.1.4. Implementasi Fitur *Lock Exam*

Pada implementasi fitur *lock exam* ini, pengguna dengan *role student* saat ujian dapat

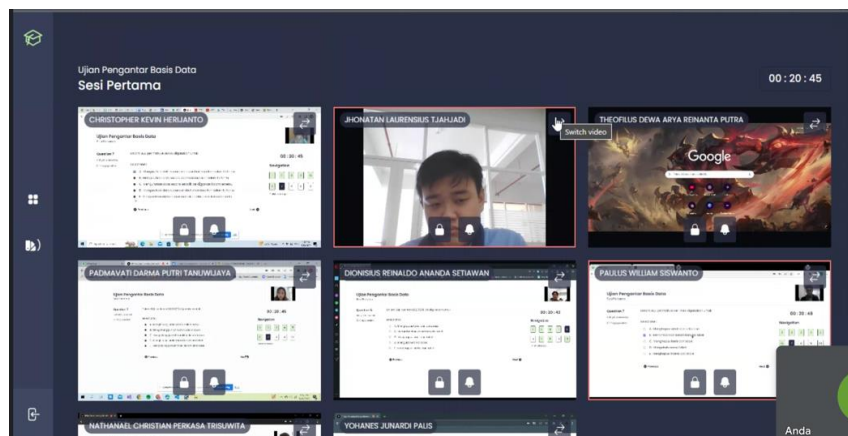


Gambar 9. Implementasi fitur *lock exam*

### 3.1.5. Implementasi Fitur *Attention Scoring*

Pada implementasi fitur *attention scoring*, atensi *student* saat mengerjakan ujian akan diperoleh dan

dikirimkan melalui *socket* ke *proctor* seperti pada Gambar 10 di bawah ini.



Gambar 10. Implementasi fitur *attention scoring*

Apabila *student* memiliki atensi skor yang rendah selama 6 kali berturut-turut, maka layar *student* akan otomatis terkunci.

## 3.2. Hasil Pengujian

Pengujian aplikasi bertujuan agar dapat menemukan *bug* atau masalah yang dapat terjadi ketika aplikasi digunakan. Selain itu, pengujian ini juga digunakan untuk mengidentifikasi aspek-aspek yang perlu diperbaiki atau dikembangkan lebih lanjut. Dalam penelitian ini, jenis pengujian yang akan dilakukan adalah pengujian *Alpha* dan *Beta*. Pengujian *alpha* dan *beta* adalah dua tahapan penting dalam siklus hidup pengembangan perangkat lunak, yang bertujuan untuk memastikan kualitas dan fungsionalitas produk perangkat lunak sebelum dirilis ke pasar [16].

### 3.2.1. Pengujian *Alpha*

Pengujian *alpha* merupakan pengujian yang dilakukan oleh perancang aplikasi dan ataupun setiap

orang yang terlibat dalam pembuatan aplikasi. Tujuannya adalah untuk menemukan masalah, *bug*, dan kekurangan yang ada dalam aplikasi sebelum aplikasi tersebut dirilis dan digunakan oleh pengguna secara publik. Pada penelitian ini pengujian *alpha* yang digunakan adalah pengujian *black box*.

Metode pengujian *black box* memiliki peran penting dalam memvalidasi dan memverifikasi fungsionalitas aplikasi atau sistem perangkat lunak. Dengan mengamati hasil output yang dihasilkan dari input yang diberikan, pengujian ini dapat mengidentifikasi potensi kesalahan atau *bug* yang mungkin terjadi dalam aplikasi atau sistem. Hal ini memungkinkan tim pengembang untuk memperbaiki dan meningkatkan performa aplikasi atau sistem agar sesuai dengan kebutuhan dan harapan pengguna. Tabel 1 di bawah ini menunjukkan hasil dari pengujian *black box* pada aplikasi *website* secara fungsionalitas.



Tabel 1. Hasil *black box testing* pada *website*

No	Fungsi Aplikasi	Kondisi	Aksi yang diharapkan	Aksi yang diperoleh	Status pengujian
1	Login / SignIn kedalam website	Username dan password sudah terisi dengan benar	Sukses melakukan autentikasi ke dalam website dan dialihkan ke dashboard sesuai dengan role pengguna	Sukses melakukan autentikasi ke dalam website dan dialihkan ke dashboard sesuai dengan role pengguna	Valid
2	Menampilkan informasi pada dashboard sesuai dengan role	Pengguna sudah dalam kondisi Logged In dan berada pada halaman dashboard pengguna	Sukses menampilkan informasi ringkas pada dashboard sesuai dengan role pengguna	Sukses menampilkan informasi ringkas pada dashboard sesuai dengan role pengguna	Valid
3	Menampilkan 5 sesi ujian yang akan berlangsung pada halaman dashboard Proctor	Proctor sudah dalam kondisi Logged In dan berada pada halaman dashboard	Sukses menampilkan list sesi ujian yang akan berlangsung pada halaman dashboard proctor	Sukses menampilkan list sesi ujian yang akan berlangsung pada halaman dashboard proctor	Valid
4	Menampilkan 5 sesi ujian yang dimiliki dan akan berlangsung pada halaman dashboard Student	Student sudah dalam kondisi Logged In dan berada pada halaman dashboard	Sukses menampilkan list sesi ujian yang dimiliki dan akan berlangsung pada halaman dashboard student	Sukses menampilkan list sesi ujian yang dimiliki dan akan berlangsung pada halaman dashboard student	Valid
5	Menampilkan tabel list user pada halaman dashboard Admin	Admin sudah dalam kondisi Logged In dan berada pada halaman dashboard	Sukses menampilkan tabel list user pada halaman dashboard Admin	Sukses menampilkan tabel list user pada halaman dashboard Admin	Valid
6	Menampilkan informasi sesi ujian pada sisi kanan halaman ketika card sesi ujian	Proctor dan Student dalam kondisi Logged In dan berada pada halaman dashboard	Sukses menampilkan informasi sesi ujian dan tombol aksi masuk ke dalam sesi ujian sesuai dengan hak akses pengguna apabila 10 menit sebelum jam dimulai.	Sukses menampilkan informasi sesi ujian dan tombol aksi masuk ke dalam sesi ujian sesuai dengan hak akses pengguna apabila 10 menit sebelum jam dimulai.	Valid
7	Menampilkan halaman live proctoring ketika Proctor menekan tombol Start Proctoring pada sesi ujian	Proctor dalam kondisi Logged In dan berada pada halaman dashboard. Kemudian menekan tombol Start Proctoring.	Sukses menampilkan halaman live proctoring dan kumpulan stream webcam Student	Sukses menampilkan halaman live proctoring dan kumpulan stream webcam Student	Valid
8	Menampilkan halaman setup device pada saat Student menekan tombol Start pada sesi ujian	Student dalam kondisi Logged In dan berada pada halaman dashboard. Kemudian menekan tombol Start pada sesi ujian	Sukses menampilkan halaman setup device dan tombol untuk melakukan setup pada webcam dan share screen	Sukses menampilkan halaman setup device dan tombol untuk melakukan setup pada webcam dan share screen	Valid
9	Melakukan setup device	Student berada pada halaman setup device dan menekan tombol setup untuk memulai menyalakan webcam dan mengaktifkan share screen	Sukses mengaktifkan webcam dan share screen layar Student. Sukses menampilkan feedback webcam dan share screen.	Sukses mengaktifkan webcam dan share screen layar Student. Sukses menampilkan feedback webcam dan share screen.	Valid
10	Memulai attempt pada exam	Student berada pada halaman setup device dan telah mengaktifkan webcam dan share screen. Kemudian menekan tombol next sesuai dengan jam untuk memulai attempt pada exam	Sukses masuk ke dalam halaman exam dan melihat soal, navigasi, serta feedback webcam Student	Sukses masuk ke dalam halaman exam dan melihat soal, navigasi, serta feedback webcam Student	Valid
11	Navigasi ke soal lain	Student berada pada halaman exam dan melakukan navigasi menuju soal-soal lainnya	Sukses menampilkan soal-soal lainnya	Sukses menampilkan soal-soal lainnya	Valid
12	Memilih jawaban dari pertanyaan	Student berada pada halaman exam dan memilih opsi jawaban dari soal	Sukses mengirimkan dan menyimpan jawaban Student dan memperbarui state jawaban soal	Sukses mengirimkan dan menyimpan jawaban Student dan memperbarui state jawaban soal	Valid
13	Menampilkan feedback webcam dan share screen Student	Proctor berada pada halaman live proctoring	Sukses menampilkan stream webcam dan share screen student	Sukses menampilkan stream webcam dan share screen student	Valid
14	Melakukan switching pada stream Student dari webcam ke share screen dan sebaliknya	Proctor berada pada halaman live proctoring dan menekan tombol switch pada stream Student	Sukses mengubah stream Student dari webcam ke share screen dan sebaliknya	Sukses mengubah stream Student dari webcam ke share screen dan sebaliknya	Valid
15	Memperbarui attention score Student	Proctor pada halaman live proctoring	Sukses memperbarui attention score Student	Sukses memperbarui attention score Student	Valid

16	Mengubah posisi stream Student apabila perlu perhatian lebih oleh Proctor	Proctor pada halaman live proctoring dan menekan tombol attention score yang rendah selama 3 detik	Sukses mengubah posisi stream Student ke baris paling atas dan memberikan border agar memudahkan Proctor untuk memberikan perhatian lebih	Sukses mengubah posisi stream Student ke baris paling atas dan memberikan border agar memudahkan Proctor untuk memberikan perhatian lebih	Valid
17	Memberikan alert pada Student	Proctor pada halaman live proctoring dan menekan tombol alert pada stream student	Sukses menampilkan alert pada student	Sukses menampilkan alert pada student	Valid
18	Mengunci halaman exam Student	Proctor pada halaman live proctoring dan menekan tombol lock pada stream student	Sukses mengunci exam Student dan mengubah tampilan halaman Student	Sukses mengunci exam Student dan mengubah tampilan halaman Student	Valid
19	Membuka kunci halaman exam Student	Halaman Student dalam posisi terkunci dan Student telah menuliskan alasan dan mensubmit untuk direview oleh Proctor. Proctor membaca alasan tersebut dan menekan tombol unlock	Sukses membuka kunci exam Student dan menampilkan kembali halaman ujian	Sukses membuka kunci exam Student dan menampilkan kembali halaman ujian	Valid
	Menampilkan laporan ujian selama sesi	Proctor berada pada halaman report dan memilih sesi ujian beserta Student	Sukses menampilkan laporan seperti jumlah aktivitas, jenis aktivitas, tangkapan webcam dan video rekaman	Sukses menampilkan laporan seperti jumlah aktivitas, jenis aktivitas, tangkapan webcam dan video rekaman	Valid

### 3.2.2. Pengujian Beta

Tujuan utama dari pengujian *beta* adalah untuk memvalidasi aplikasi atau perangkat lunak yang telah dibuat sebelum diluncurkan secara publik, dengan melibatkan calon pengguna sebagai pengujinya. Dalam penelitian ini, pengujian beta dilakukan dengan melibatkan 2 dosen dan 16 mahasiswa sebagai calon pengguna, yang secara khusus dilakukan dalam bentuk *closed beta testing*.

Uji coba ini merupakan *User Acceptance Testing* (UAT) untuk menjawab kebutuhan dari pengguna. Aspek yang diuji berdasarkan *Nielsen's Usability Criteria* pada *software development* yang didefinisikan oleh Nielsen Norman Group, dimana terdapat 5 komponen kualitas yang terdiri dari: *Learnability*; *Efficiency*; *Memorability*; *Errors*; *Satisfaction* yang selanjutnya dipakai sebagai variabel penilaian [17]. Penjelasan mengenai variabel penilaian dapat dilihat pada Tabel 2 di bawah ini.

Tabel 2. Variabel penilaian UAT

Variabel	Penjelasan
<i>Learnability</i>	Menilai seberapa mudah bagi pengguna untuk menyelesaikan tugas-tugas dasar saat pertama kali pengguna menggunakan aplikasi
<i>Efficiency</i>	Setelah pengguna mempelajari penggunaan aplikasi, maka selanjutnya akan dinilai seberapa cepat pengguna dapat melakukan tugas tersebut
<i>Memorability</i>	Menilai seberapa mudah pengguna mengingat letak setiap menu, cara penggunaan fitur yang ada pada aplikasi
<i>Errors</i>	Menilai berapa banyak kesalahan yang dilakukan pengguna, seberapa parah kesalahan ini, dan seberapa mudah mereka dapat pulih dari kesalahan. Sehingga, pada penilaian ini dapat diketahui tingkat kesalahan maupun kerusakan pada aplikasi
<i>Satisfaction</i>	Menilai tingkat antusiasme dari pengguna untuk terus menggunakan aplikasi

Selanjutnya, dilakukan uji coba secara langsung pada salah satu kelas yang hendak menjalankan ujian *online* berdasarkan permintaan salah satu dosen di

salah satu Universitas di Surabaya. Hasil dari pengujian *beta* yang telah dilakukan pada *student* dan *proctor* dapat dilihat pada Tabel 3 dan 4 dibawah ini.

Tabel 3. Hasil wawancara *student*

Variabel	P1	P2	P3	P4	P5	Mean	Standard Deviation	
<b>Learnability</b>	4.267	4.333	4.267	3.533	3.467	3.973	0.434	
<b>Efficiency</b>	3.867	4.067				3.967	0.141	
<b>Memorability</b>	4.2667	4.5333	4.2	4.2		4.3	0.159	
<b>Errors</b>	4	3.8	3.733			3.844	0.139	
<b>Satisfaction</b>	3.6	3	4.067	3.6		3.567	0.437	
<b>Mean Total</b>	<b>3.930</b>							

Tabel 4. Hasil wawancara *proctor*

Variabel	P1	P2	P3	P4	P5	P6	Mean	Standard Deviation
<b>Learnability</b>	4	4	3.5	2.5	4.5	4	3.750	0.689
<b>Efficiency</b>	5	2					3.500	2.121
<b>Memorability</b>	4	4	4.5	4.5			4.250	0.289
<b>Errors</b>	3.5	3.5	3.5				3.500	0.000
<b>Satisfaction</b>	5	4	4	3	4.5	5	4.250	0.758
<b>Mean Total</b>	<b>3.850</b>							

Secara keseluruhan, hasil pengujian kegunaan dan umpan balik memberikan wawasan berharga tentang fitur dan area lain yang perlu diperbaiki dari aplikasi tersebut. Umpan balik positif mengenai *learnability* dan *memorability* menunjukkan nilai yang kuat, sementara penilaian yang bervariasi dalam *efficiency* menyoroti perlunya pengujian dan optimalisasi lebih lanjut. *Satisfaction* secara garis

besar memiliki nilai tinggi adalah hal yang memuaskan, namun saran-saran khusus dari pengguna pengawas harus dipertimbangkan untuk meningkatkan fungsionalitas aplikasi dan mengatasi masalah yang dilaporkan. Oleh sebab itu, dilakukan penilaian dan umpan balik terhadap fitur yang hasilnya dapat dilihat pada Tabel 5 di bawah ini.

Tabel 5. Hasil penilaian terhadap fitur sistem

Fitur	Rata-Rata
<b>Live Proctoring:</b>	4,5
- Fitur ini memungkinkan pengawas ujian untuk melihat tangkapan <i>webcam</i> dan layar peserta secara langsung, guna memudahkan pengawasan ujian secara <i>real-time</i> .	
- Fitur ini efektif memantau <i>webcam</i> dan layar peserta selama ujian.	
<b>Attention Alert:</b>	5
- Fitur ini membantu pengawas ujian dengan memberikan tanda atau notifikasi jika peserta tidak fokus pada layar ujian, sehingga membantu meningkatkan pengawasan terhadap peserta yang mungkin terdistraksi.	
- Adanya warna border sebagai Fitur <i>Attention Alert</i> membantu saya mengawasi ujian dengan lebih baik karena disesuaikan dengan skor perhatian peserta.	
<b>Exam Lock:</b>	4,5
- Fitur ini memungkinkan pengawas ujian untuk mengunci ujian peserta, sehingga mencegah peserta untuk mengakses atau keluar dari ujian sebelum waktu yang ditentukan.	
- Fitur <i>Exam Lock</i> efektif mencegah peserta melanjutkan ujian karena membutuhkan persetujuan pengawas untuk membuka kunci ujian.	
<b>Live Alert:</b>	4,5
- Fitur ini memungkinkan pengawas untuk memberikan peringatan secara <i>real-time</i> kepada peserta ujian, seperti peraturan ujian atau instruksi khusus.	
- Fitur <i>Live Alert</i> membantu dalam memberikan peringatan secara <i>real-time</i> kepada peserta ujian.	
<b>Active Tab &amp; Window Detection:</b>	4
- Fitur ini membantu pengawas dalam mendeteksi tab browser dan jendela aplikasi yang aktif pada komputer peserta ujian. Hal ini berguna untuk mencegah peserta dari membuka tab atau aplikasi lain yang tidak diizinkan selama ujian, sehingga menjaga integritas ujian.	
- Fitur <i>Active Tab &amp; Window Detection</i> berhasil mencegah peserta membuka tab atau jendela aplikasi lain selama ujian dengan mengunci ujian secara otomatis jika peserta melanggar.	

Dengan demikian, hasil penelitian ini menunjukkan bahwa aplikasi ini telah berhasil memberikan solusi yang efektif dalam memenuhi kebutuhan pengawas ujian mengacu pada hasil penilaian fitur pada tabel di atas. Umpan balik yang diberikan memberikan wawasan berharga bagi peneliti untuk terus meningkatkan kinerja, fungsionalitas, dan kepuasan pengguna aplikasi ini.

#### 4. DISKUSI

Pada penelitian sebelumnya yang dilakukan Samsul Fauzi dan Darmawati telah dikembangkan aplikasi ujian *online* berbasis *web* yang dapat diawasi oleh pengawas ujian secara langsung di SMA PGRI Bayah [11]. Salah satu fitur unggulannya adalah *video call* yang dapat digunakan saat siswa mengerjakan ujian, hal ini memungkinkan pengawas dapat melihat tangkapan gambar dari *webcam* siswa. Namun, fitur *video call* ini tidak begitu perlu melihat bahwa tangkapan *webcam* pengawas juga tampil di halaman siswa. Berbeda dengan penelitian ini, dimana tangkapan *webcam* pengawas juga ditampilkan di halaman siswa.

Keterbaruan pada sistem yang dihasilkan penelitian ini adalah pada penerapan fitur deteksi *attention score* dari peserta ujian. Penggunaan *border* untuk menunjukkan *attention score* yang rendah dikombinasikan dengan fitur kunci ujian dapat membantu peserta ujian dalam mendapatkan

peringatan untuk tetap fokus kepada ujian *online*. Hal ini mendorong peserta ujian untuk tetap fokus dan berhati-hati karena fitur kunci ujian mensimulasikan bahwa waktu ujian terus berjalan meskipun layar ujian terkunci, sehingga dapat menyebabkan kerugian yang signifikan bagi peserta ujian yang waktunya terbuang untuk melakukan klarifikasi kepada pengawas ujian.

#### 5. KESIMPULAN

Dalam penelitian ini, telah berhasil dikembangkan sebuah sistem pemantauan ujian *online* secara *real time* menggunakan teknologi *WebRTC* dengan fitur-fitur pencegahan. Keamanan dan integritas ujian *online* dapat terjaga melalui pemantauan aktivitas peserta secara *real time* dan peringatan *attention score* saat terdeteksi nilai *attention score* rendah. Penggunaan *server* dengan spesifikasi yang dituliskan mampu menangani lebih dari 10 partisipan dalam satu sesi tanpa degradasi kinerja. Konfigurasi sistem pada satu *droplet* memberikan manfaat seperti manajemen yang lebih mudah, penurunan latensi, dan komunikasi yang lebih efisien. Performa sistem yang baik, serta umpan balik pengguna yang positif, memperkuat potensinya sebagai alat yang dapat diandalkan untuk pengawasan ujian *online*.

Proses pengujian yang dilakukan dalam penelitian ini telah berhasil menjawab bagaimana

merancang dan mengembangkan sistem pemantauan ujian *online* secara *real-time* menggunakan teknologi WebRTC, dengan menerapkan fitur-fitur pencegahan. Berikut ini adalah kesimpulan yang dapat diambil dari pengujian tersebut:

- Pemantauan ujian *online* menjadi lebih mudah dengan adanya fitur *streaming webcam*, berbagi layar, dan *attention score*. Pengawas dapat dengan mudah mengamati *webcam* dan layar peserta. Penggunaan *border* untuk menunjukkan *attention score* yang rendah juga membantu dalam pemantauan setelah pengawas terbiasa dengan sistem ini.
- Fitur kunci dan peringatan memfasilitasi tindakan pencegahan yang dilakukan pengawas secara dini. Namun, tidak adanya fitur pemindahan otomatis video peserta yang memerlukan perhatian menjadi kekurangan pada *website* ini.
- Fitur kunci ujian juga mendorong peserta untuk lebih berhati-hati agar ujian mereka tidak terkunci. Hal ini mendorong mereka untuk tetap fokus dan berhati-hati karena ujian akan terus berjalan meskipun terkunci, yang dapat menyebabkan kerugian yang signifikan.

Kesimpulan ini menunjukkan keberhasilan implementasi sistem pemantauan ujian *online* secara *real-time*, yang efektif memfasilitasi proses pemantauan, tindakan pencegahan, dan peningkatan perhatian peserta. Performa sistem ini, didukung oleh umpan balik pengguna yang diperoleh selama pengujian, memperkuat potensinya sebagai alat yang dapat diandalkan untuk pengawasan ujian *online*, dimana berdasarkan hasil pengujian dengan pengguna fitur *Live Proctoring* mendapatkan skor 4,5; *Attention Alert* mendapatkan skor 5; *Exam Lock* mendapatkan skor 4,5; *Live Alert* mendapatkan skor 4,5; dan *Tab & Window Detection* mendapatkan skor 4.

#### DAFTAR PUSTAKA

- [1] R. Raman, B. Sairam, G. Veena, H. Vachharajani, & P. Nedungadi, "Adoption of online proctored examinations by university students during covid-19: innovation diffusion study", *Education and Information Technologies*, vol. 26, no. 6, p. 7339-7358, 2021. <https://doi.org/10.1007/s10639-021-10581-5>.
- [2] N. Selwyn, C. O'Neill, G. Smith, M. Andrejevic, & X. Gu, "A necessary evil? the rise of online exam proctoring in australian universities", *Media International Australia*, vol. 186, no. 1, p. 149-164, 2021. <https://doi.org/10.1177/1329878x211005862>
- [3] A. Prabowo and D. Wardani, "Faktor-faktor yang mempengaruhi kecurangan akademik pada perkuliahan online", *Kajian Bisnis Sekolah Tinggi Ilmu Ekonomi Widya Wiwaha*, vol. 29, no. 1, p. 16-29, 2021. <https://doi.org/10.32477/jkb.v29i1.235>.
- [4] E. Bilen and A. Matros, "Online cheating amid COVID-19," *J Econ Behav Organ*, vol. 182, pp. 196–211, Feb. 2021, doi: 10.1016/j.jebo.2020.12.004.
- [5] J. M. Dyer, H. C. Pettyjohn, and S. Saladin, "Academic Dishonesty and Testing: How Student Beliefs and Test Academic Dishonesty and Testing: How Student Beliefs and Test Settings Impact Decisions to Cheat Settings Impact Decisions to Cheat," 2020. [Online]. Available: [https://dc.cod.edu/testing\\_pubs](https://dc.cod.edu/testing_pubs).
- [6] R. Conijn, A. Kleingeld, U. Matzat, & C. Snijders, "The fear of big brother: the potential negative side-effects of proctored exams", *Journal of Computer Assisted Learning*, vol. 38, no. 6, p. 1521-1534, 2022. <https://doi.org/10.1111/jcal.12651>.
- [7] J. Pleasants, J. M. Pleasants, and B. P. Pleasants, "Cheating on Unproctored Online Exams: Prevalence, Mitigation Measures, and Effects on Exam Performance," *Online Learning Journal*, vol. 26, no. 1, pp. 268–284, Mar. 2022, doi: 10.24059/olj.v26i1.2620.
- [8] C. Hébert, "Online Remote Proctoring Software in the Neoliberal Institution: Measurement, Accountability, and Testing Culture," 2021. [Online]. Available: <https://orcid.org/0000-0001-9775-9416>.
- [9] K. Linden and P. Gonzalez, "Zoom invigilated exams: A protocol for rapid adoption to remote examinations," *British Journal of Educational Technology*, vol. 52, no. 4, pp. 1323–1337, Jul. 2021, doi: 10.1111/bjet.13109.
- [10] L. M, L. Luo, S. Sikdar, N. Nizam, S. Gao, H. Shanet al., "Optimized collusion prevention for online exams during social distancing", *NPJ Science of Learning*, vol. 6, no. 1, 2021. <https://doi.org/10.1038/s41539-020-00083-3>.
- [11] S. Fauzi, "OKTAL : Jurnal Ilmu Komputer dan Science Pengembangan Aplikasi Ujian Online Berbasis Web Dengan Memanfaatkan Teknologi Web Real-Time Communication (Web RTC) Di SMP PGRI BAYAH," 2022, [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal>.
- [12] R. Ainanda and E. V. Haryanto, "Rancang Bangun Aplikasi E-meeting Menggunakan WebRTC (Web Real time Communication) Design and Build E-meeting Applications Using WebRTC (Web Real time Communication)," 2020.

- [13] F. N. Azzam, D. P. Kartikasari, and F. A. Bakhtiar, "Implementasi Video Conference dengan File Sharing menggunakan WebRTC," 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [14] M. A. Arshad et al., "Psychological Support Application Using The Real-Time Webrtc: A Case Study Of Undergraduate In Malaysian And Indonesian Higher Learning Institutions," 2021.
- [15] J. LEE, "Impact of proctoring environments on student performance: online vs offline proctored exams", *Journal of Asian Finance Economics and Business*, vol. 7, no. 8, p. 653-660, 2020. <https://doi.org/10.13106/jafeb.2020.vol7.no8.653>.
- [16] A. Spillner and T. Linz, *Software Testing Foundations*. dpunkt.verlag Heidelberg.
- [17] "Usability 101: Introduction to Usability." <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (accessed May 20, 2023).