



KERJA PRAKTIK - IF184801

**Rancang Bangun Aplikasi Pengawasan Ujian Online
(iProctor) untuk Pengawas dan Peserta Ujian**

Departemen Teknik Informatika
Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, Keputih, Kec. Sukulilo, Kota Surabaya
Periode: 14 Maret 2022 – 20 Mei 2022

Oleh:

Daniel Sugianto

05111940000075

Pembimbing Departemen

Dini Adni Navastara, S.Kom, M.Sc.

Pembimbing Lapangan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2022



KERJA PRAKTIK - IF184801

**Rancang Bangun Aplikasi Pengawasan Ujian Online
(iProctor) untuk Pengawas dan Peserta Ujian**

Departemen Teknik Informatika
Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, Keputih, Kec. Sukolilo, Kota Surabaya
Periode: 14 Maret 2022 – 20 Mei 2022

Oleh:

Daniel Sugianto

05111940000075

Pembimbing Departemen

Dini Adni Navastara, S.Kom, M.Sc.

Pembimbing Lapangan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2022

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI.....	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
LEMBAR PENGESAHAN.....	xii
KATA PENGANTAR.....	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	2
1.3. Manfaat.....	2
1.4. Rumusan Masalah	2
1.5. Lokasi dan Waktu Kerja Praktik	2
1.6. Metodologi Kerja Praktik.....	2
1.6.1. Perumusan Masalah.....	2
1.6.2. Studi Literatur.....	3
1.6.3. Analisis dan Perancangan Sistem	3
1.6.4. Implementasi Sistem	3
1.6.5. Pengujian dan Evaluasi.....	3
1.6.6. Kesimpulan dan Saran.....	3
1.7. Sistematika Laporan	3
1.7.1. Bab I Pendahuluan.....	4

1.7.2.	Bab II Profil Perusahaan.....	4
1.7.3.	Bab III Tinjauan Pustaka	4
1.7.4.	Bab IV Analisis dan Perancangan Sistem	4
1.7.5.	Bab V Implementasi Sistem	4
1.7.6.	Bab VI Pengujian dan Evaluasi.....	4
1.7.7.	Bab VII Kesimpulan dan Saran.....	4
BAB II PROFIL PERUSAHAAN		5
2.1.	Sejarah Departemen Teknik Informatika	5
2.2.	Visi dan Misi Departemen Teknik Informatika.....	6
2.3.	Tujuan Departemen Teknik Informatika	6
BAB III TINJAUAN PUSTAKA.....		9
3.1.	Python.....	9
3.2.	Face Recognition.....	9
3.3.	Voice Activity Detection.....	10
3.4.	Tkinter	10
BAB IV ANALISIS DAN PERANCANGAN SISTEM		11
4.1.	Analisis Sistem.....	11
4.2.	Definisi Umum Aplikasi.....	11
4.3.	Deskripsi Fungsional.....	12
4.4.	Perancangan Sistem.....	16
BAB V IMPLEMENTASI SISTEM.....		25
5.1.	Implementasi Source Code.....	25

5.2. Implementasi Antarmuka Pengguna.....	30
BAB VI PENGUJIAN DAN EVALUASI	37
6.1. Tujuan Pengujian.....	37
6.2. Kriteria Pengujian.....	37
6.3. Skenario Pengujian.....	38
6.4. Evaluasi Pengujian	38
BAB VII KESIMPULAN DAN SARAN	41
7.1. Kesimpulan.....	41
7.2. Saran.....	41
DAFTAR PUSTAKA.....	43
BIODATA PENULIS	45

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1 Use case diagram aplikasi iProctor.....	13
Gambar 4.2 Diagram Alir Aplikasi iProctor Student	19
Gambar 4.3 Diagram Alir Aplikasi iProctor Proctor.....	20
Gambar 4.4 Spesifikasi endpoint login pengawas.....	21
Gambar 4.5 Spesifikasi endpoint login peserta	22
Gambar 4.6 Spesifikasi endpoint unggah tingkat keyakinan peserta	23
Gambar 4.7 Spesifikasi endpoint unduh tingkat keyakinan peserta	24
Gambar 4.8 Spesifikasi endpoint ping sesi.....	25
Gambar 4.9 Spesifikasi endpoint akhiri sesi	25
Gambar 5.1 Student base controller	27
Gambar 5.2 Student login controller	28
Gambar 5.3 Student instruction controller	28
Gambar 5.4 Student calibration controller	29
Gambar 5.5 Student end controller.....	29
Gambar 5.6 Proctor base controller.....	30
Gambar 5.7 Proctor login controller.....	31
Gambar 5.8 Proctor dashboard controller	32
Gambar 5.9 Halaman Login Peserta.....	33
Gambar 5.10 Halaman Instruksi Peserta	34
Gambar 5.11 Halaman Kalibrasi Peserta.....	35
Gambar 5.12 Halaman Akhir Peserta	36
Gambar 5.13 Halaman Login Pengawas	37
Gambar 5.14 Halaman Dashboard Pengawas	38

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Spesifikasi use case melihat instruksi	14
Tabel 4.2 Spesifikasi use case kalibrasi wajah dan suara.....	14
Tabel 4.3 Spesifikasi use case melihat data peserta	15
Tabel 4.4 Spesifikasi use case melihat detail sesi	16
Tabel 4.5 Spesifikasi use case melihat grafik tingkat keyakinan	16
Tabel 4.6 Spesifikasi use case menandai peserta	17
Tabel 4.7 Spesifikasi use case mengakhiri sesi.....	18
Tabel 6.1 Hasil Evaluasi Pengujian.....	41

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Rancang Bangun Aplikasi Pengawasan Ujian Online
(iProctor) untuk Pengawas dan Peserta Ujian**

Oleh:

Daniel Sugianto

05111940000075

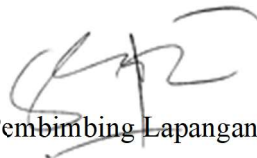
Disetujui oleh Pembimbing Kerja Praktik:

1. Dini Adni Navastara, S.Kom,
M.Sc.
NIP. 198510172015042001



(Pembimbing Departemen)

2. Shintami Chusnul Hidayati,
S.Kom., M.Sc., Ph.D.
NPP. 1987202012004



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Rancang Bangun Aplikasi Pengawasan Ujian Online (iProctor) untuk Pengawas dan Peserta Ujian

Nama Mahasiswa : Daniel Sugianto
NRP : 0511194000075
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dini Adni Navastara, S.Kom, M.Sc.
Pembimbing Lapangan : Shintami Chusnul Hidayati, S.Kom.,
M.Sc., Ph.D.

ABSTRAK

Pandemi COVID-19 telah mengubah banyak hal dalam kehidupan manusia. Aktivitas yang umumnya dilakukan secara offline telah berubah menjadi online. Pelaksanaan ujian merupakan salah satu aktivitas yang mengalami perubahan tersebut. Pelaksanaan ujian secara online menimbulkan permasalahan baru, yaitu bagaimana cara mengawasi semua siswa agar ujian dapat berjalan dengan jujur.

Aplikasi iProctor dirancang untuk menyelesaikan permasalahan tersebut. Aplikasi iProctor Student akan mengawasi siswa dengan memanfaatkan teknologi deteksi ucapan dan pengenalan wajah. Selanjutnya, aplikasi iProctor Proctor akan menampilkan hasil deteksi kecurangan setiap siswa kepada pengawas dalam bentuk probabilitas kecurangan.

Aplikasi iProctor dibuat menggunakan bahasa pemrograman Python, library Tkinter untuk antarmuka pengguna, library face-recognition untuk pengenalan wajah, dan library py-webrtcvad untuk deteksi ucapan.

Kata Kunci: Python, Tkinter, Face Recognition, Speech Detection

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Rancang Bangun Aplikasi Pengawasan Ujian Online (iProctor) untuk Pengawas dan Peserta Ujian.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Dini Adni Navastara, S.Kom, M.Sc. selaku dosen pembimbing departemen.
3. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D. selaku pembimbing lapangan.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 27 Mei 2022

Daniel Sugianto

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pandemi COVID-19 telah mengubah banyak hal dalam kehidupan manusia. Aktivitas yang dahulu umum dilakukan secara luar jaringan (offline) sekarang telah berubah menjadi dalam jaringan (online). Salah satu aktivitas yang mengalami perubahan tersebut adalah pelaksanaan ujian. Pelaksanaan ujian yang dahulu offline di mana peserta diawasi secara langsung oleh pengawas telah berevolusi menjadi online di mana peserta ujian diawasi secara tidak langsung melalui platform meeting oleh pengawas. Pelaksanaan ujian secara online tersebut membuka banyak peluang bagi peserta ujian untuk melakukan kecurangan karena pengawas tidak dapat mengawasi semua peserta dalam waktu yang bersamaan seperti saat pelaksanaan ujian secara offline. Jika pelaksanaan ujian tidak diawasi dengan baik, maka akan memunculkan banyak kasus kecurangan yang menyebabkan penurunan tingkat akurasi penilaian kemampuan peserta ujian.

Untuk menyelesaikan permasalahan tersebut, kami merancang aplikasi yang mengotomasi pengawasan ujian online dengan memanfaatkan kecerdasan buatan yang memungkinkan pengawas untuk mengawasi lebih banyak peserta ujian dalam waktu yang bersamaan. Aplikasi tersebut akan mengawasi kecocokan wajah, arah pandang, dan aktivitas suara peserta ujian selama ujian berlangsung.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar dua sks dan membantu Departemen Teknik Informatika dalam membangun aplikasi pengawasan ujian online.

1.3. Manfaat

Manfaat yang diperoleh dengan adanya aplikasi pengawasan ujian online antara lain adalah mempermudah pengawas dalam mengawasi peserta ujian online dan menjaga kejujuran peserta ujian dalam mengerjakan ujian yang dilakukan secara online.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana cara mendeteksi kecurangan dalam ujian online?
2. Bagaimana alur pengawasan ujian yang dilakukan secara online?

1.5. Lokasi dan Waktu Kerja Praktik

Sehubungan dengan adanya pandemi dan diberlakukannya *Work From Home*, pengerjaan kerja praktik ini dilakukan secara *remote*.

Adapun kerja praktik dimulai pada tanggal 14 Maret 2022 hingga 20 Mei 2022.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi:

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan-kebutuhan fungsional dan non-fungsional dari sistem, kami mengikuti rapat-rapat bersama Ibu Dini dan Ibu Shintami. Pada rapat tersebut, dijelaskan tentang

kebutuhan utama dari aplikasi iProctor. Setelah rapat selesai, tim pengembang membuat rancangan aplikasi iProctor.

1.6.2. Studi Literatur

Setelah ditentukan kebutuhan dari aplikasi iProctor, dilakukan studi literatur mengenai cara implementasinya, dalam hal ini tinjauan dilakukan terhadap bahasa pemrograman Python, metode pengenalan wajah, metode deteksi ucapan, dan library Tkinter. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi segala bentuk code yang ada.

1.6.3. Analisis dan Perancangan Sistem

Setelah melakukan tinjauan pustaka, untuk merancang sistem yang baik perlu adanya sebuah desain arsitektur sistem. Pada aplikasi ini tim *developer* setuju untuk menggunakan arsitektur desain MVC (Model - View - Controller).

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini kami membangun aplikasi iProctor dari awal.

1.6.5. Pengujian dan Evaluasi

Setelah aplikasi yang direncanakan telah jadi, perlu adanya evaluasi untuk menguji apakah aplikasi sesuai dengan harapan client. Jika masih belum sesuai atau perlu menambah fitur, maka rapat akan dilakukan lagi untuk membahas fitur - fitur apa saja yang perlu diperbaiki atau ditambah.

1.6.6. Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi kerja praktik, waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini menjelaskan secara rinci mengenai profil perusahaan tempat kerja praktik dilaksanakan, yaitu Departemen Teknik Informatika.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan Sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

BAB II

PROFIL PERUSAHAAN

2.1. Sejarah Departemen Teknik Informatika

Sebagai perkiraan untuk saat ini dan masa yang akan datang, teknologi informasi menjadi tulang punggung pertumbuhan ekonomi bangsa. Saat ini pun invasi teknologi informasi sudah terasa di berbagai bidang kehidupan manusia. Hal ini sepenuhnya disadari oleh pemerintah, sehingga sejak Repelita V yang lalu, pemerintah telah mencanangkan bahwa pengembangan pendidikan tinggi dalam bidang komputer dan informatika merupakan salah satu program prioritas, bersama-sama dengan disiplin ilmu lainnya seperti rekayasa, perilaku, manajemen, akuntansi, dan kesenian.

Pendidikan tinggi diarahkan untuk mempersiapkan bangsa Indonesia dalam menghadapi era pembangunan industri dan informasi. Untuk itu pemerintah melalui Direktorat Jendral Pendidikan Tinggi pada tahun 1985 menginstruksikan untuk membuka Program Studi S1 baru untuk bidang ilmu teknologi komputer di empat universitas atau institut di mana ITS termasuk di dalamnya. Di ITS, program ini awalnya diberi nama Program Studi Teknik Komputer. Namun sejak tahun 1993, nama Program Studi Teknik Komputer diubah menjadi Jurusan Teknik Komputer. Akhirnya, pada tahun 1996 secara resmi jurusan ini berganti nama menjadi Jurusan Teknik Informatika berdasarkan Surat Keputusan Direktur Jendral Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996, tanggal 11 Juli 1996. Pada saat ini, Jurusan Teknik Informatika memperoleh nilai akreditasi A berdasarkan Surat Keputusan Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) Nomor 003/BAN-PT/Ak-X/S1/V/2006, tanggal 18 Mei 2006.

Selain program Sarjana (S1), Jurusan Teknik Informatika juga menyelenggarakan program Pasca Sarjana (S2) yang dirintis sejak tahun 1994, dengan surat keputusan Direktur Jendral Pendidikan Tinggi No. 2851/D/T/2001, perihal ijin penyelenggaraan Program-

Program Studi Jenjang Program Strata-2 (S2) pada Institut Teknologi Sepuluh Nopember Surabaya. Pada tahun 2011, Jurusan Teknik Informatika mulai menyelenggarakan program Doktor (S3).

2.2. Visi dan Misi Departemen Teknik Informatika

2.2.1. Visi

Sejalan dengan visi ITS yaitu menjadi perguruan tinggi dengan reputasi internasional dalam ilmu pengetahuan, teknologi, dan seni, terutama yang menunjang industri dan kelautan yang berwawasan lingkungan, maka visi Departemen Informatika adalah menjadi inovator bidang informatika yang unggul di tingkat nasional dengan reputasi internasional, serta berperan aktif dalam upaya memajukan dan mensejahterakan bangsa.

2.2.2. Misi

Departemen Informatika memiliki misi sebagai berikut:

1. Menyelenggarakan proses pembelajaran yang berkualitas dan memenuhi standar nasional maupun internasional.
2. Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
3. Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
4. Menjalin kemitraan dengan berbagai lembaga.

2.3. Tujuan Departemen Teknik Informatika

Tujuan dari Departemen Informatika ITS adalah sebagai berikut:

1. Menghasilkan lulusan yang kompeten di bidang Informatika, serta memiliki daya saing dan kemandirian untuk berkompetisi di tingkat nasional dan internasional.
2. Melakukan perbaikan proses pembelajaran secara berkesinambungan.

3. Menghasilkan karya penelitian yang inovatif dan bermanfaat bagi masyarakat, serta publikasi di jurnal nasional ataupun internasional bereputasi.
4. Memberikan kontribusi bagi peningkatan mutu kehidupan masyarakat.
5. Mengambil peran aktif dalam kegiatan bidang Informatika pada tingkat nasional dan internasional.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Python

Python adalah bahasa pemrograman high-level yang interpreted, berorientasi objek, dan memiliki semantik dinamis. Python memiliki struktur data built-in, dynamic typing, dan dynamic binding yang sangat cocok untuk pengembangan aplikasi secara cepat (Rapid Application Development). Python memiliki sintaks yang mudah dibaca, sehingga dapat mengurangi biaya perawatan program. Python juga mendukung modul dan paket yang dapat meningkatkan modularitas program dan penggunaan ulang kode. Interpreter Python dan library standar dari Python dapat digunakan secara gratis dan dapat didistribusikan secara bebas.

3.2. Face Recognition

Face recognition adalah teknologi yang mampu mengenali wajah manusia dari gambar digital atau frame video dengan menggunakan data wajah yang tersimpan dalam suatu basis data. Pada umumnya, face recognition digunakan untuk mengautentikasi pengguna dengan cara menemukan dan mengukur beberapa fitur dari gambar wajah pengguna. Library Python face_recognition yang dibuat oleh Adam Geitgey memungkinkan penggunaan face recognition dalam program Python.

3.3. Voice Activity Detection

Voice activity detection (VAD) adalah deteksi keberadaan ucapan manusia dan membedakan antara bagian audio yang berupa ucapan dan bukan ucapan. VAD umum digunakan dalam speech coding dan speech recognition. Speech coding adalah proses untuk mendapatkan representasi kompak dari sinyal suara untuk transmisi yang lebih efisien melalui saluran yang memiliki band terbatas. Speech recognition adalah kemampuan program untuk mengubah ucapan manusia menjadi format tertulis. Library Python `py-webrtcvad` yang dibuat oleh John Wiseman memungkinkan penggunaan voice activity detection dalam program Python.

3.4. Tkinter

Tkinter adalah paket standar Python yang digunakan untuk membuat graphical user interface atau GUI. Setiap distribusi standar Python pasti memiliki Tkinter. Tkinter menyediakan interface ke Tk toolkit dan bekerja sebagai lapisan object-oriented di atas Tk. Tk toolkit adalah koleksi dari elemen kontrol grafik atau widget. Tkinter memiliki beragam komponen atau widget built-in yang dapat digunakan untuk membuat antarmuka pengguna, seperti komponen `Frame` untuk membuat kerangka GUI, `OptionMenu` untuk membuat menu dropdown, `Button` untuk membuat tombol, `Entry` untuk membuat kotak input, dan `Label` untuk membuat label.

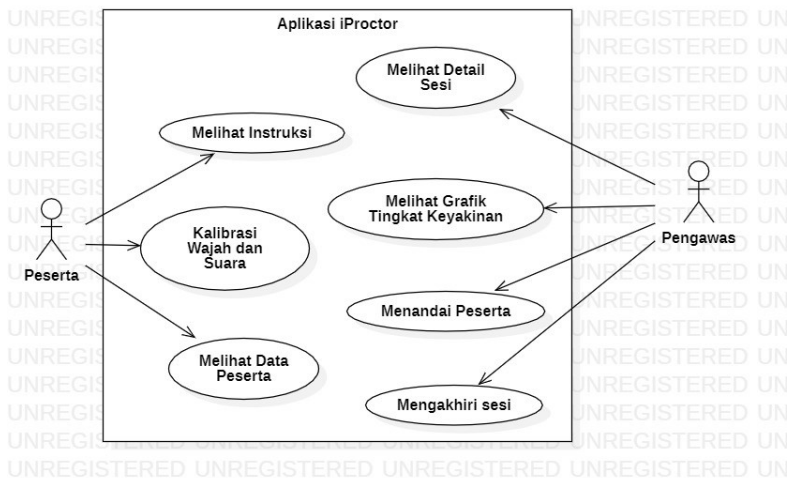
BAB IV ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam pembangunan aplikasi iProctor, yaitu analisis dari sistem yang dibangun. Penjelasan dibagi menjadi dua bagian, yaitu definisi umum aplikasi dan deskripsi fungsional.

4.2. Definisi Umum Aplikasi

Aplikasi iProctor adalah aplikasi desktop yang digunakan untuk membantu pengawas dalam mengawasi dan mendeteksi kecurangan dalam suatu ujian online. Peserta akan menjalankan aplikasi iProctor Student pada komputer masing-masing untuk mendeteksi kecurangan. Pengawas akan menjalankan aplikasi iProctor Proctor untuk melihat hasil deteksi kecurangan setiap peserta ujian.



Gambar 4.1 Use case diagram aplikasi iProctor

4.3. Deskripsi Fungsional

Pada deskripsi fungsional ini akan ditampilkan setiap spesifikasi use case dalam pembangunan aplikasi iProctor.

4.3.1. F001: Melihat instruksi

Tabel 4.1 merupakan spesifikasi use case melihat instruksi pada aplikasi iProctor.

Tabel 4.1 Spesifikasi use case melihat instruksi

Nama Use Case	Melihat instruksi
Nomor	F001
Aktor	Peserta
Deskripsi	Peserta melihat instruksi penggunaan aplikasi iProctor
Relasi	-
Kondisi Awal	Peserta berhasil login
Kondisi Akhir	-
Alur Kejadian Normal	
Aktor	Sistem
1. Peserta menekan tombol masuk pada halaman login	1.1. Sistem menampilkan instruksi penggunaan aplikasi iProctor
Alur Kejadian Alternatif	
-	

4.3.2. F002: Kalibrasi wajah dan suara

Tabel 4.2 merupakan spesifikasi use case kalibrasi wajah dan suara pada aplikasi iProctor.

Tabel 4.2 Spesifikasi use case kalibrasi wajah dan suara

Nama Use Case	Kalibrasi wajah dan suara
Nomor	F002
Aktor	Peserta

Deskripsi	Peserta melakukan kalibrasi wajah dan suara
Relasi	-
Kondisi Awal	-
Kondisi Akhir	-
Alur Kejadian Normal	
Aktor	Sistem
1. Peserta menekan tombol lanjut pada halaman instruksi	1.1. Sistem menampilkan status kalibrasi wajah dan suara
	1.2 Sistem menampilkan teks deklarasi
2. Peserta membaca teks deklarasi dengan menghadap ke kamera	2.1. Sistem memperbarui status kalibrasi wajah dan suara
	2.2 Sistem membuka tombol lanjut
Alur Kejadian Alternatif	
-	

4.3.3. F003: Melihat data peserta

Tabel 4.3 merupakan spesifikasi use case melihat data peserta pada aplikasi iProctor.

Tabel 4.3 Spesifikasi use case melihat data peserta

Nama Use Case	Melihat data peserta
Nomor	F003
Aktor	Peserta
Deskripsi	Peserta melihat data miliknya
Relasi	-
Kondisi Awal	Peserta berhasil kalibrasi wajah dan suara
Kondisi Akhir	-
Alur Kejadian Normal	

Aktor	Sistem
1. Peserta menekan tombol lanjut pada halaman kalibrasi	1.1. Sistem menampilkan data peserta
Alur Kejadian Alternatif	
-	

4.3.4. F004: Melihat detail sesi

Tabel 4.4 merupakan spesifikasi use case melihat detail sesi pada aplikasi iProctor.

Tabel 4.4 Spesifikasi use case melihat detail sesi

Nama Use Case	Melihat detail sesi
Nomor	F004
Aktor	Pengawas
Deskripsi	Pengawas melihat detail sesi
Relasi	-
Kondisi Awal	Pengawas berhasil login
Kondisi Akhir	-
Alur Kejadian Normal	
Aktor	Sistem
1. Pengawas menekan tombol masuk pada halaman login	1.1. Sistem menampilkan daftar peserta
	1.2 Sistem menampilkan id sesi dan token sesi
Alur Kejadian Alternatif	
-	

4.3.5. F005: Melihat grafik tingkat keyakinan

Tabel 4.5 merupakan spesifikasi use case melihat grafik tingkat keyakinan pada aplikasi iProctor.

Tabel 4.5 Spesifikasi use case melihat grafik tingkat keyakinan

Nama Use Case	Melihat grafik tingkat keyakinan
---------------	----------------------------------

Nomor	F005
Aktor	Pengawas
Deskripsi	Pengawas melihat grafik tingkat keyakinan seorang peserta
Relasi	-
Kondisi Awal	-
Kondisi Akhir	-
Alur Kejadian Normal	
Aktor	Sistem
1. Pengawas menekan tombol lihat pada item peserta	1.1. Sistem menampilkan grafik tingkat keyakinan dari peserta yang dipilih
Alur Kejadian Alternatif	
-	

4.3.6. F006: Menandai peserta

Tabel 4.6 merupakan spesifikasi use case menandai peserta pada aplikasi iProctor.

Tabel 4.6 Spesifikasi use case menandai peserta

Nama Use Case	Menandai peserta
Nomor	F006
Aktor	Pengawas
Deskripsi	Pengawas menandai peserta yang diduga curang
Relasi	-
Kondisi Awal	-
Kondisi Akhir	-
Alur Kejadian Normal	
Aktor	Sistem
1. Pengawas menekan tombol tandai pada item peserta	1.1. Sistem mengubah warna tombol tandai
Alur Kejadian Alternatif	
-	

4.3.7. F007: Mengakhiri sesi

Tabel 4.7 merupakan spesifikasi use case mengakhiri sesi pada aplikasi iProctor.

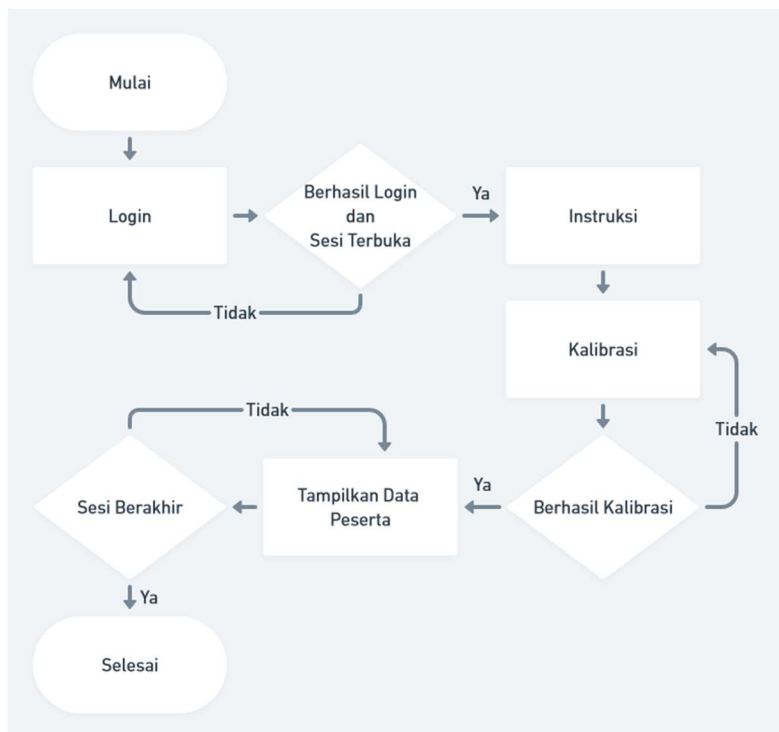
Tabel 4.7 Spesifikasi use case mengakhiri sesi

Nama Use Case	Mengakhiri sesi
Nomor	F007
Aktor	Pengawas
Deskripsi	Pengawas mengakhiri sesi
Relasi	-
Kondisi Awal	Sesi sedang berjalan
Kondisi Akhir	-
Alur Kejadian Normal	
Aktor	Sistem
1. Pengawas menekan tombol akhiri sesi	1.1. Sistem menampilkan modal untuk mengkonfirmasi akhiri sesi
2. Pengawas memilih tombol OK	2.1 Sistem membangkitkan laporan daftar peserta yang ditandai
Alur Kejadian Alternatif	
-	

4.4. Perancangan Sistem

Pada perancangan sistem ini akan dijelaskan mengenai aliran aplikasi iProctor Student, aliran aplikasi iProctor Proctor, spesifikasi API iProctor.

4.4.1. Aliran Aplikasi iProctor Student

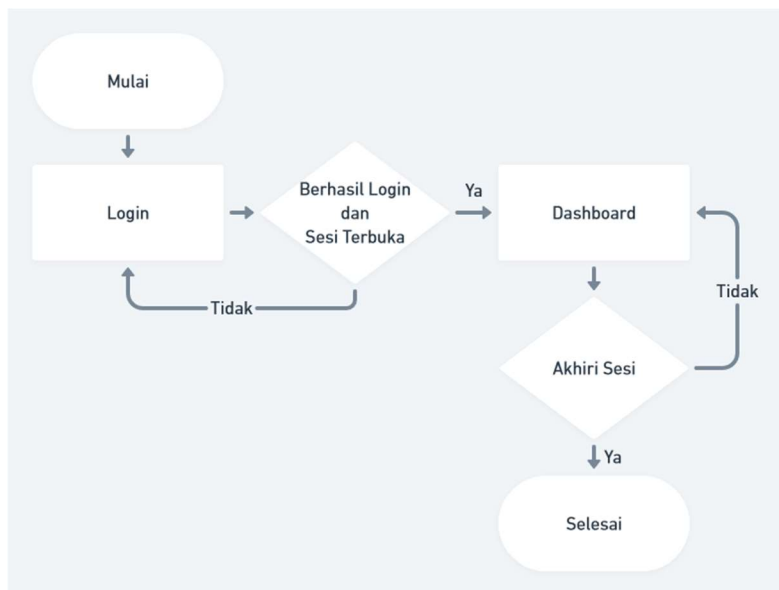


Gambar 4.2 Diagram Alir Aplikasi iProctor Student

Aplikasi iProctor Student adalah aplikasi desktop yang dijalankan pada komputer peserta ujian. Pada gambar 4.2 dapat dilihat bahwa alur aplikasi dimulai dengan peserta melakukan login. Selanjutnya, peserta diarahkan ke halaman instruksi di mana sistem menampilkan instruksi penggunaan aplikasi iProctor Student sembari menginisiasi model audio-video. Setelah peserta selesai membaca instruksi, peserta akan melakukan kalibrasi wajah dan suara di halaman kalibrasi. Kalibrasi dinyatakan berhasil jika

wajah peserta terdeteksi dan suara peserta terdeteksi. Pada halaman akhir, sistem akan menampilkan user id milik peserta, menjalankan model audio-video untuk mendeteksi kecurangan selama sesi berjalan, dan melakukan ping ke API iProctor untuk memeriksa status dari sesi. Hasil deteksi kecurangan oleh model audio-video adalah sebuah angka dalam range $[0, 1]$ yang menyatakan probabilitas kecurangan peserta. Angka tersebut akan dikirim secara berkala ke API iProctor.

4.4.2. Aliran Aplikasi iProctor Proctor



Gambar 4.3 Diagram Alir Aplikasi iProctor Proctor

Aplikasi iProctor Proctor adalah aplikasi desktop yang dijalankan pada komputer pengawas ujian. Pada gambar 4.3 dapat dilihat bahwa alur aplikasi dimulai dari pengawas melakukan login. Selanjutnya, pengawas diarahkan ke halaman dashboard di

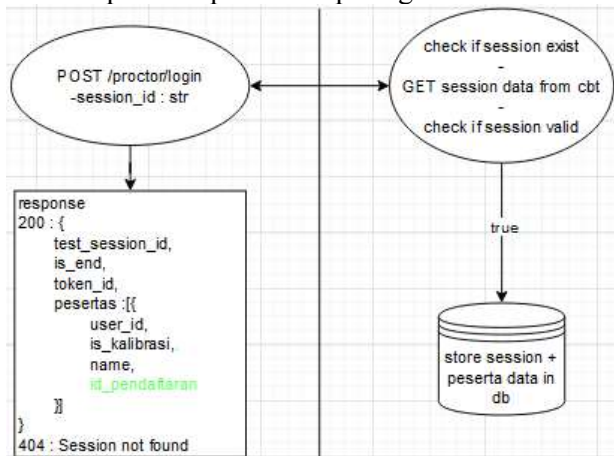
mana pengawas dapat melihat daftar peserta, detail sesi, dan grafik tingkat keyakinan seorang peserta. Tingkat keyakinan setiap peserta akan diperbarui secara berkala.

4.4.3. Spesifikasi API iProctor

Berikut adalah spesifikasi dari Application Programming Interface (API) iProctor. Format dari request dan response yang digunakan adalah JavaScript Object Notation (JSON).

4.4.3.1. Login Pengawas

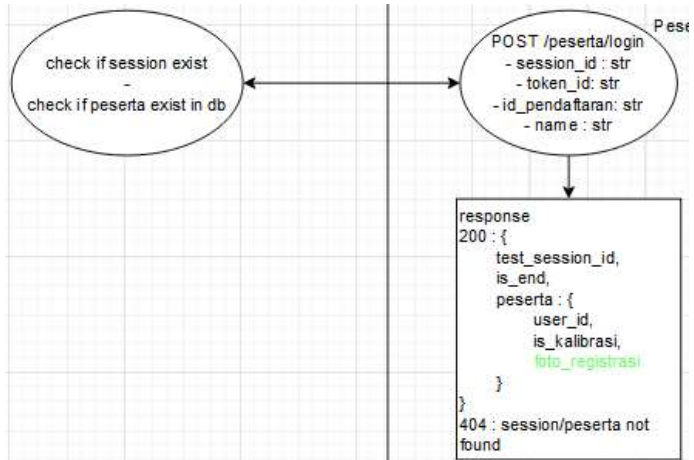
Login pengawas dilakukan dengan mengirimkan request POST ke endpoint “/proctor/login” dengan payload `session_id` yang valid. Response dapat dilihat pada gambar 4.4.



Gambar 4.4 Spesifikasi endpoint login pengawas

4.4.3.2. Login Peserta

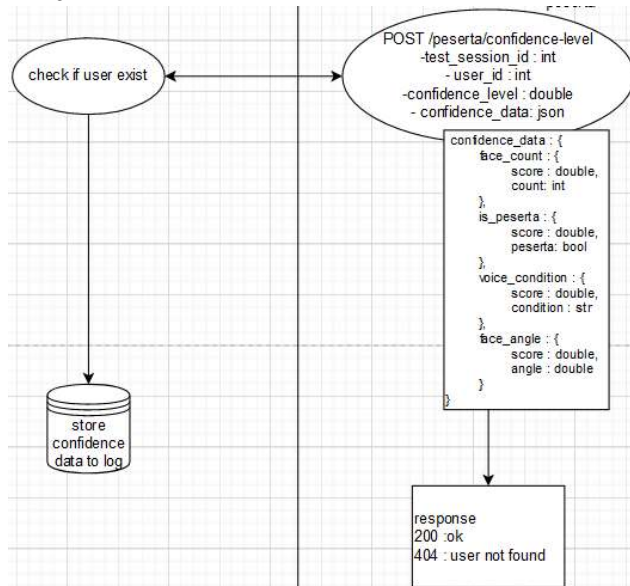
Login peserta dilakukan dengan mengirim request POST ke endpoint “/peserta/login” dengan payload session_id, token_id, id_pendaftaran, dan name yang valid. Response dapat dilihat pada gambar 4.5.



Gambar 4.5 Spesifikasi endpoint login peserta

4.4.3.3. Unggah Tingkat Keyakinan Peserta

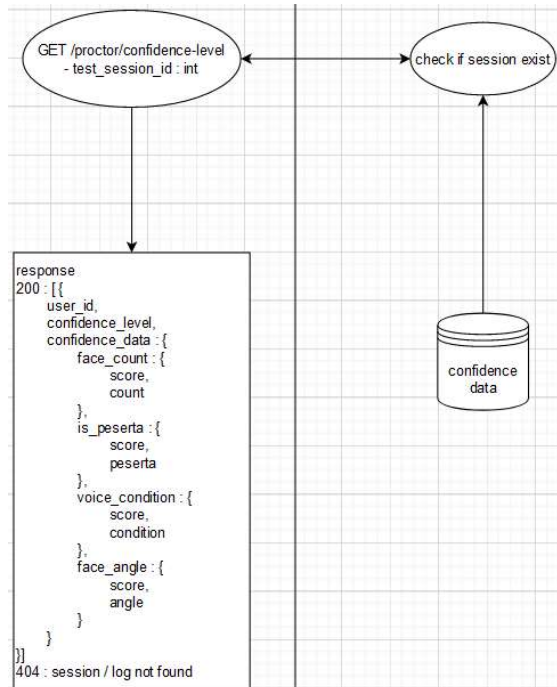
Unggah tingkat keyakinan peserta (hasil deteksi kecurangan peserta) dilakukan dengan mengirim request POST ke endpoint “/peserta/confidence-level” dengan payload `test_session_id`, `user_id`, `confidence_level`, dan `confidence_data` yang valid. Detail dari atribut `confidence_data` dan response dapat dilihat pada gambar 4.6.



Gambar 4.6 Spesifikasi endpoint unggah tingkat keyakinan peserta

4.4.3.4. Unduh Tingkat Keyakinan Peserta

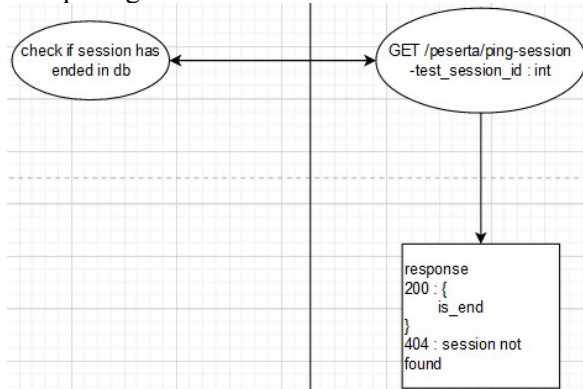
Unduh tingkat keyakinan peserta dapat dilakukan dengan mengirim request GET ke endpoint “/proctor/confidence-level” dengan payload test_session_id yang valid. Response dapat dilihat pada gambar 4.7.



Gambar 4.7 Spesifikasi endpoint unduh tingkat keyakinan peserta

4.4.3.5. Ping Sesi

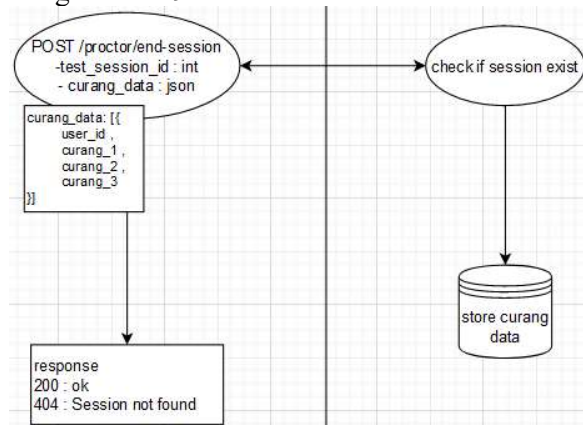
Ping sesi dilakukan dengan mengirim request GET ke endpoint “/peserta/ping-session” dengan payload test_session_id. Response dapat dilihat pada gambar 4.8.



Gambar 4.8 Spesifikasi endpoint ping sesi

4.4.3.6. Akhiri Sesi

Akhiri sesi dapat dilakukan dengan mengirim request POST ke endpoint “/proctor/end-session” dengan payload test_session_id dan curang_data. Detail atribut curang_data dan response dapat dilihat pada gambar 4.9.



Gambar 4.9 Spesifikasi endpoint akhiri sesi

[Halaman ini sengaja dikosongkan]

BAB V

IMPLEMENTASI SISTEM

5.1. Implementasi Source Code

Bagian ini menjelaskan mengenai kode sumber dalam pembangunan aplikasi iProctor Student dan iProctor Proctor.

5.1.1. iProctor Student

Berikut adalah penjelasan dari source code aplikasi iProctor Student.

5.1.1.1. Student Base Controller

BaseController merupakan base class atau template untuk membuat controller lainnya. Inisiasi BaseController membutuhkan root (objek tkinter.Tk), thread_manager (utility class untuk mengatur thread), dictionary models, dictionary views, target_view (nama view yang diatur), dan proctor (objek model audio-video). Binding antara view yang diatur dengan controller akan dilakukan saat BaseController diinisiasi (baris 8).

```
controllers > base.py > BaseController > _init_
1 class BaseController:
2     def __init__(self, root, thread_manager, models, views, target_view, proctor):
3         self.root = root
4         self.thread_manager = thread_manager
5         self.models = models
6         self.views = views
7         self.target_view = target_view
8         self.views[target_view].set_controller(self)
9         self.views[target_view].render()
10        self.views[target_view].style()
11        self.proctor = proctor
```

Gambar 5.1 Student base controller

5.1.1.2. Student Login Controller

StudentLoginController adalah controller yang menghubungkan view login dengan model-model yang berkaitan dengan login peserta. Method “authenticate” (baris 12) berfungsi untuk membuat thread yang memanggil method “__attempt” dengan argumen berupa credential yang telah dimasukkan oleh peserta ujian di halaman login. Method “__attempt” (baris 35) berfungsi untuk mengirim request login ke API iProctor. Method

“__init_proctor” (baris 66) berfungsi untuk menginisialisasi model audio-video.

```
controllers > login.py > StudentLoginController > authenticate
1 import threading
2 import socket
3 from controllers.base import BaseController
4 from models.auth import Auth, Credential, StudentLoginRequest
5 from models.session import Session
6 from models.student import Student
7
8 class StudentLoginController(BaseController):
9 > def __init__(self, root, thread_manager, models, views, target_view, proctor):
11
12 > def authenticate(self): ...
34
35 > def __attempt(self, credential, camera_index, image_path): ...
65
66 > def __init_proctor(self, camera_index): ...
```

Gambar 5.2 Student login controller

5.1.1.3. Student Instruction Controller

StudentInstructionController adalah controller yang mengatur halaman instruction. Method “__check_proctor” (baris 9) berfungsi untuk memeriksa apakah model audio-video sudah selesai diload. Method “next” (baris 15) berfungsi untuk lanjut ke halaman kalibrasi.

```
controllers > instruction.py > StudentInstructionController
1 import threading
2 from controllers.base import BaseController
3
4 class StudentInstructionController(BaseController):
5 > def __init__(self, root, thread_manager, models, views, target_view, proctor):
8
9 > def __check_proctor(self): ...
14
15 > def next(self): ...
22
```

Gambar 5.3 Student instruction controller

5.1.1.4. Student Calibration Controller

StudentCalibrationController adalah controller yang menghubungkan halaman kalibrasi dengan model yang berkaitan dengan kalibrasi peserta. Method “__show_face_detections” (baris 18) berfungsi untuk memperbarui frame hasil deteksi wajah peserta di halaman kalibrasi. Method “__calibrate_video” (baris 24) berfungsi untuk memeriksa kecocokan wajah peserta dengan foto wajah peserta. Method “__calibrate_audio” (baris 31) berfungsi

untuk mendeteksi keberadaan ucapan peserta. Method “__update_status” (baris 37) berfungsi untuk memperbarui status kalibrasi video dan audio di halaman kalibrasi. Method “next” (baris 52) berfungsi untuk lanjut ke halaman akhir.

```
controllers > calibration.py > ...
1 import threading
2 from controllers.base import BaseController
3 from models.session import PostPhotoRequest
4
5 class StudentCalibrationController(BaseController):
6 > def __init__(self, root, thread_manager, models, views, target_view, proctor):
17
18 > def __show_face_detections(self): ...
23
24 > def __calibrate_video(self): ...
30
31 > def __calibrate_audio(self): ...
36
37 > def __update_status(self): ...
51
52 > def next(self): ...
60
```

Gambar 5.4 Student calibration controller

5.1.1.5. Student End Controller

StudentEndController berfungsi untuk mengatur halaman akhir. Method “__update_info” (baris 17) digunakan untuk memperbarui data peserta di halaman akhir. Method “__check_session” (baris 20) berfungsi untuk memeriksa status sesi secara berkala. Method “__post_proctor_result” (baris 28) berfungsi untuk mengunggah hasil deteksi kecurangan model audio-video ke API iProctor. Method “minimize” (baris 44) berfungsi untuk minimize aplikasi.

```
controllers > end.py > ...
1 import threading
2 from controllers.base import BaseController
3 from models.session import PostConfidenceLevelRequest
4
5 class StudentEndController(BaseController):
6 > def __init__(self, root, thread_manager, models, views, target_view, proctor):
16
17 > def __update_info(self): ...
19
20 > def __check_session(self): ...
27
28 > def __post_proctor_result(self): ...
43
44 > def minimize(self): ...
46
```

Gambar 5.5 Student end controller

5.1.2. iProctor Proctor

Berikut adalah penjelasan dari source code aplikasi iProctor Proctor.

5.1.2.1. Proctor Base Controller

BaseController merupakan base class atau template untuk membuat controller lainnya. Inisiasi BaseController membutuhkan root (objek tkinter.Tk), thread_manager (utility class untuk mengatur thread), dictionary models, dictionary views, dan target_view (nama view yang diatur). Binding antara view yang diatur dengan controller akan dilakukan saat BaseController diinisiasi (baris 8).

```
controllers > base.py > ...
1 class BaseController:
2     def __init__(self, root, thread_manager, models, views, target_view):
3         self.root = root
4         self.thread_manager = thread_manager
5         self.models = models
6         self.views = views
7         self.target_view = target_view
8         self.views[target_view].set_controller(self)
9         self.views[target_view].render()
10        self.views[target_view].style()
```

Gambar 5.6 Proctor base controller

5.1.2.2. Proctor Login Controller

ProctorLoginController adalah controller yang menghubungkan view login dengan model-model yang berkaitan dengan login pengawas. Method “authenticate” (baris 14) berfungsi untuk membuat thread yang memanggil method “__attempt” dengan argumen berupa credential yang telah dimasukkan oleh pengawas ujian di halaman login. Method “__attempt” (baris 26) berfungsi untuk mengirim request login ke API iProctor.

```
controllers > login.py > ProctorLoginController > _attempt
1  √ import threading
2
3  from sympy import arg
4  from controllers.base import BaseController
5  from models.auth import Auth, Credential, ProctorLoginRequest
6  from models.session import Session
7  from models.student_map import Student
8
9
10 √ class ProctorLoginController(BaseController):
11 >   def __init__(self, root, thread_manager, models, views, target_view):...
13
14 >   def authenticate(self):...
25
26 >   def __attempt(self, session_id):...
```

Gambar 5.7 Proctor login controller

5.1.2.3. Proctor Dashboard Controller

ProctorDashboardController berfungsi untuk mengatur halaman dashboard pengawas ujian. Method “__update_session_info” (baris 16) berfungsi untuk memperbarui informasi sesi di halaman dashboard. Method “set_student_flag” (baris 22) berfungsi untuk memperbarui flag dari seorang peserta. Method “show_graph” (baris 29) berfungsi untuk memperbarui grafik tingkat keyakinan sesuai dengan peserta yang dipilih. Method “__update_students” (baris 37) berfungsi untuk memperbarui daftar peserta ujian secara berkala. Method “end_session” (baris 58) berfungsi untuk mengakhiri sesi dan membangkitkan daftar peserta ujian yang ditandai.

```
controllers > dashboard.py > ProctorDashboardController > end_session
1  from datetime import datetime
2  import threading
3  import pandas as pd
4  from pathlib import Path
5  from controllers.base import BaseController
6  from models.session import PostEndSessionRequestCurangData
7  from views.dashboard import StudentTableRowDto
8
9
10 class ProctorDashboardController(BaseController):
11 >     def __init__(self, root, thread_manager, models, views, target_view):
15
16 >     def __update_session_info(self): ...
21
22 >     def set_student_flag(self, student_id, flag): ...
28
29 >     def show_graph(self, student_id): ...
36
37 >     def __update_students(self): ...
57
58 >     def end_session(self): ...
```

Gambar 5.8 Proctor dashboard controller

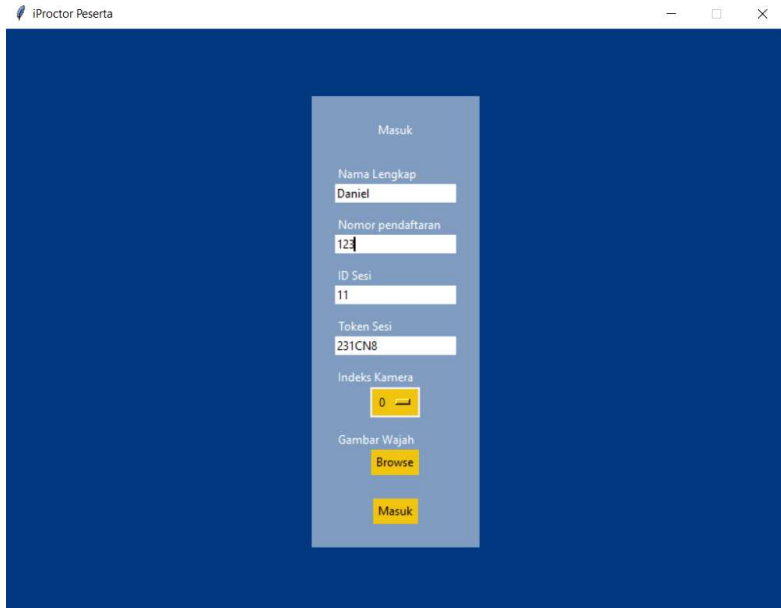
5.2. Implementasi Antarmuka Pengguna

Pada bagian ini akan ditampilkan antarmuka pengguna aplikasi iProctor Student dan iProctor Proctor.

5.2.1. iProctor Student

5.2.1.1. Halaman Login Peserta

Pada gambar 5.9 ditampilkan halaman login peserta ujian. Pada halaman tersebut, peserta harus memasukkan nama lengkap, nomor pendaftaran, id sesi, token sesi, indeks kamera, dan foto wajah peserta untuk login.



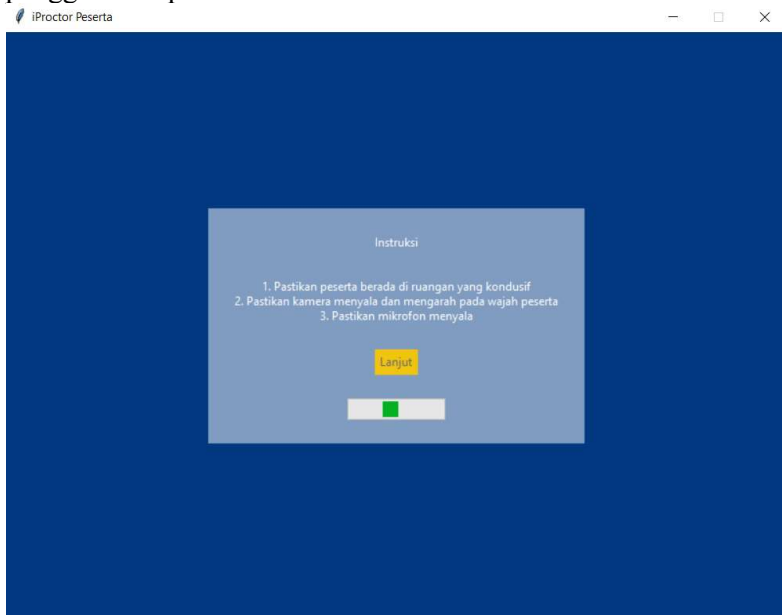
The screenshot shows a web browser window titled "iProctor Peserta". The main content is a login form on a dark blue background. The form is titled "Masuk" and contains the following fields and buttons:

- Nama Lengkap:** Input field containing "Daniel".
- Nomor pendaftaran:** Input field containing "123".
- ID Sesi:** Input field containing "11".
- Token Sesi:** Input field containing "231CN8".
- Indeks Kamera:** A dropdown menu showing "0" and a "Browse" button.
- Gambar Wajah:** A "Browse" button.
- Masuk:** A yellow button at the bottom of the form.

Gambar 5.9 Halaman Login Peserta

5.2.1.2. Halaman Instruksi Peserta

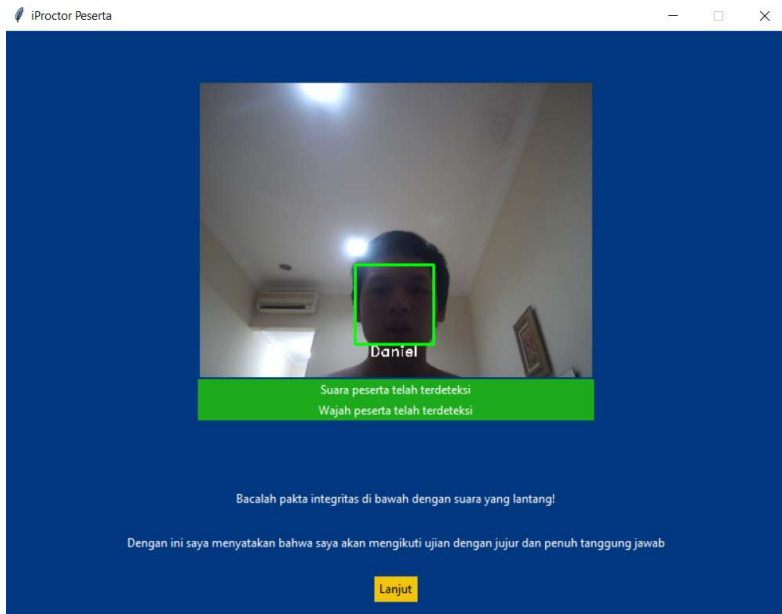
Pada gambar 5.10 ditampilkan halaman instruksi peserta ujian. Pada halaman tersebut peserta dapat membaca instruksi penggunaan aplikasi iProctor Student.



Gambar 5.10 Halaman Instruksi Peserta

5.2.1.3. Halaman Kalibrasi Peserta

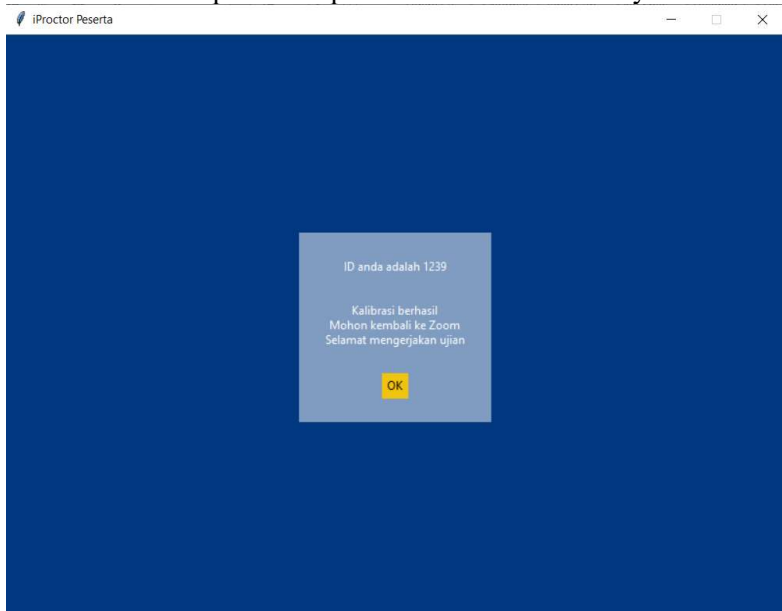
Pada gambar 5.11 ditampilkan halaman kalibrasi peserta. Pada halaman tersebut peserta ujian dapat melihat hasil deteksi wajah, status kalibrasi audio dan video, dan teks deklarasi.



Gambar 5.11 Halaman Kalibrasi Peserta

5.2.1.4. Halaman Akhir Peserta

Pada gambar 5.12 ditampilkan halaman akhir peserta. Pada halaman tersebut peserta dapat melihat user id miliknya.

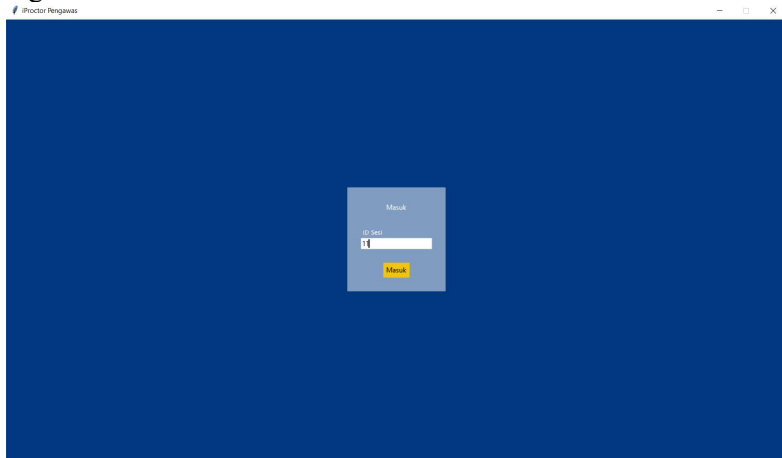


Gambar 5.12 Halaman Akhir Peserta

5.2.2. iProctor Proctor

5.2.2.1. Halaman Login Pengawas

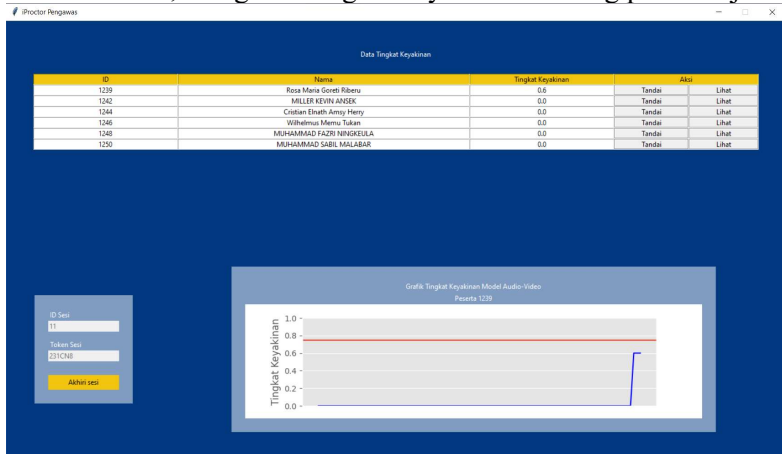
Pada gambar 5.13 ditampilkan halaman login pengawas. Pada halaman tersebut, pengawas wajib memasukkan id sesi untuk login.



Gambar 5.13 Halaman Login Pengawas

5.2.2.2. Halaman Dashboard Pengawas

Pada gambar 5.14 ditampilkan halaman dashboard pengawas. Pada halaman tersebut pengawas dapat melihat daftar peserta, informasi sesi, dan grafik tingkat keyakinan seorang peserta ujian.



Gambar 5.14 Halaman Dashboard Pengawas

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap aplikasi iProctor Student dan aplikasi iProctor Proctor. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi sistem dengan analisis dan perancangan sistem.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap aplikasi iProctor Student dan aplikasi iProctor Proctor guna melihat kinerja aplikasi yang telah dibuat, apakah berjalan sesuai dengan yang dibutuhkan dan diharapkan.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut:

- a. Kemampuan aplikasi iProctor Student untuk mengirim data ke API iProctor.
- b. Kemampuan aplikasi iProctor Student untuk mendapatkan data dari API iProctor.
- c. Kemampuan aplikasi iProctor Student untuk menjalankan kalibrasi audio dan video.
- d. Kemampuan aplikasi iProctor Student untuk mendeteksi kecurangan menggunakan model audio-video.
- e. Kemampuan aplikasi iProctor Proctor untuk mengirim data ke API iProctor.
- f. Kemampuan aplikasi iProctor Proctor untuk mendapatkan data dari API iProctor.
- g. Kemampuan aplikasi iProctor Proctor untuk membangkitkan daftar peserta ujian yang ditandai.

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai pengguna yang akan menjalankan fitur-fitur. Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut :

1. Peserta dapat melakukan kalibrasi wajah dan suara.
2. Peserta dapat melihat data miliknya.
3. Pengawas dapat melihat detail sesi.
4. Pengawas dapat melihat grafik tingkat keyakinan seorang peserta.
5. Pengawas dapat menandai seorang peserta.
6. Pengawas dapat mengakhiri sesi.

6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi iProctor Student dan aplikasi iProctor Proctor terhadap kasus skenario uji coba. Tabel 6.1 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

Tabel 6.1 Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Aplikasi iProctor Student dapat mengirim data ke API iProctor.	Terpenuhi
Aplikasi iProctor Student dapat menerima data dari API iProctor.	Terpenuhi
Aplikasi iProctor Student dapat menjalankan kalibrasi audio dan video.	Terpenuhi
Aplikasi iProctor Student dapat	Terpenuhi

Kriteria Pengujian	Hasil Pengujian
mendeteksi kecurangan menggunakan model audio-video.	
Aplikasi iProctor Proctor dapat mengirim data ke API iProctor.	Terpenuhi
Aplikasi iProctor Proctor dapat menerima data dari API iProctor.	Terpenuhi
Aplikasi iProctor Proctor dapat membangkitkan daftar peserta ujian yang ditandai.	Terpenuhi

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan perancangan aplikasi iProctor pada kegiatan kerja praktik di Departemen Teknik Informatika adalah sebagai berikut :

- a. Kecurangan dalam ujian online dapat dideteksi melalui pemeriksaan kesesuaian wajah, arah pandang, dan deteksi ucapan peserta saat ujian berlangsung yang mana dapat diotomasi menggunakan library face_recognition dan py-webrtcvad.
- b. Dengan adanya aplikasi iProctor, pengawasan ujian online yang dahulu hanya melalui visual saja, sekarang dapat ditingkatkan dengan mempertimbangkan faktor audio, yaitu deteksi ucapan.

7.2. Saran

Saran untuk perancangan sistem aplikasi iProctor adalah sebagai berikut :

- a. Deteksi wajah dapat menggunakan library mediapipe untuk meningkatkan performa model audio-video.
- b. Model audio-video untuk mendeteksi kecurangan lebih baik dijalankan pada server agar tidak memberatkan mesin client.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Geitgey, Adam. 2021. *Face Recognition's documentation*. <https://face-recognition.readthedocs.io/en/latest/>. Diakses tanggal 2 Mei 2022.
- [2] Minichino, Joe dan Joseph Howse. 2015. *Learning OpenCV 3 Computer Vision with Python*. <https://repository.unikom.ac.id/67052/1/Learning%20OpenCV%203%20Computer%20Vision%20with%20Python%20%28%20PDFDrive.com%20%29.pdf>. Diakses tanggal 2 Mei 2022.
- [3] Amos, David. 2020. *Python GUI Programming With Tkinter*. <https://realpython.com/python-gui-tkinter/>. Diakses tanggal 2 Mei 2022.
- [4] Matt. 2021. *How To Add A GUI To Your Machine Learning Model With Python*. <https://medium.com/swlh/python-oop-mvc-data-science-tkinter-23c3e8dab70f>. Diakses tanggal 2 Mei 2022.
- [5] Mueller, John Paul. 2014. *Beginning Programming with Python For Dummies*. <https://we.riseup.net/assets/345912/Beginning+Programmin+g+with+Python+For+Dummies+Mueller%2C+John+Paul+%5BBSRG%5D.pdf>. Diakses tanggal 2 Mei 2022.
- [6] Chandra, Andreas. 2018. *Voice Activity Detection Sederhana Menggunakan Python*. <https://medium.com/warung-pintar/membuat-voice-activity-detection-menggunakan-python-d13763ea277f>. Diakses tanggal 2 Mei 2022.
- [7] Moore, Alan. 2018. *Python GUI programming with Tkinter*. Birmingham: Packt Publishing.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Daniel Sugianto
Tempat, Tanggal Lahir : Denpasar, 08 Februari 2001
Jenis Kelamin : Laki-laki
Telepon : +62895336653433
Email : danielsugianto9@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2019
Semester : 6 (Enam)