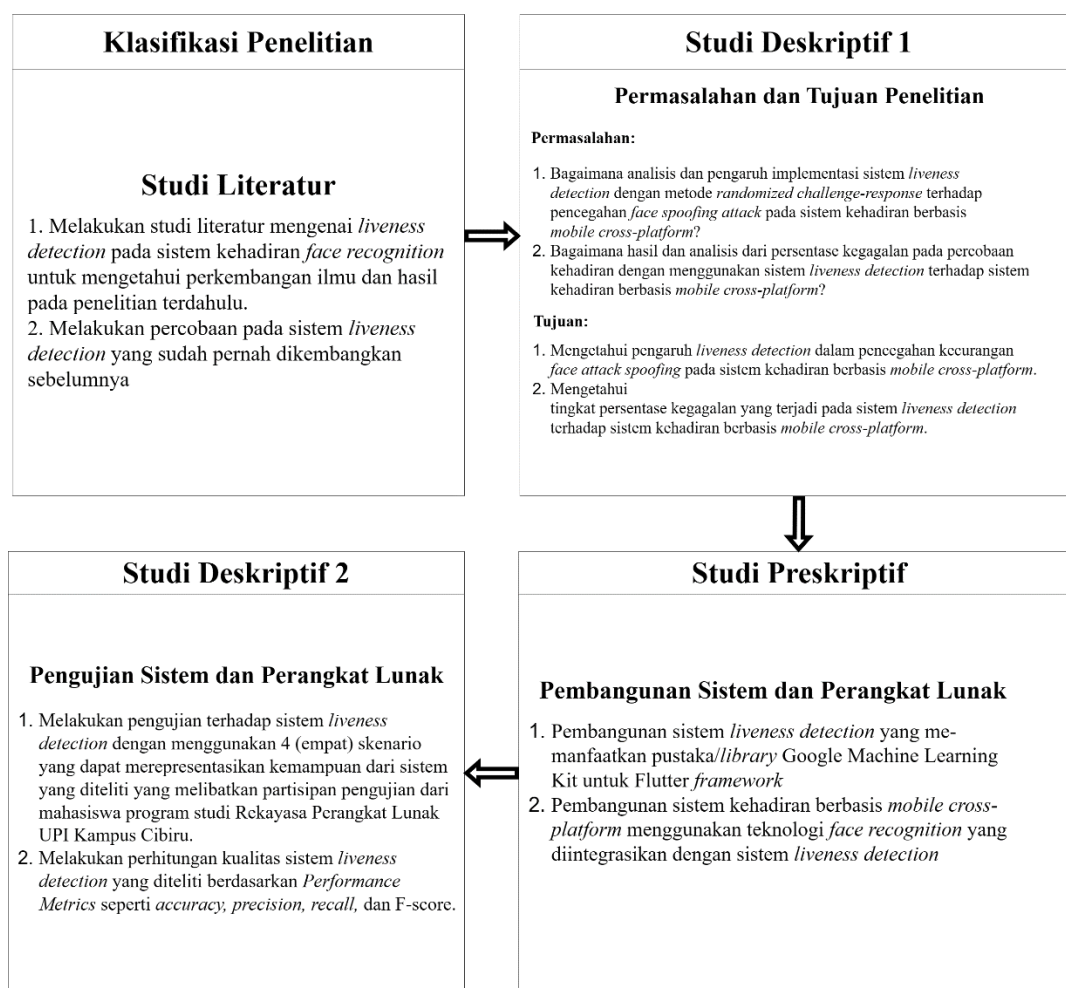


BAB III METODOLOGI PENELITIAN

3.1 Desain Penelitian

Desain penelitian yang digunakan pada penelitian adalah dengan metode *Design Research Methodology* (DRM). Pada metodologi DRM, berdasarkan penelitian (Blessing & Chakrabarti, 2009) langkah penelitian terdiri dari empat tahapan yaitu *Research Clarification* atau Klarifikasi Penelitian, *Descriptive Study 1* atau Studi deskriptif 1, *Perscriptive Study* atau studi preskriptif, dan *Descriptive Study 2* atau Studi deskriptif 2. Berikut adalah ilustrasi dari skema penelitian yang akan dilakukan berdasarkan pada kerangka kerja DRM dijelaskan pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

Berikut adalah penjelasan dari skema penelitian sebagai berikut:

3.1.1 Klarifikasi Penelitian

Tahapan pertama pada penelitian ini melakukan studi literatur yang relevan untuk memberikan informasi seputar perkembangan ilmu dan hasil pada penelitian terdahulu. Memberikan pandangan lebih terhadap tujuan penelitian dengan bidang *liveness detection* melalui kajian artikel penelitian yang relevan digunakan untuk memperluas landasan teori.

Adapun proses dalam pencarian masalahnya adalah melakukan studi literatur yang komprehensif dengan melibatkan penelusuran artikel dari jurnal nasional maupun internasional yang mengangkat topik seputar *liveness detection* dan pengaruhnya terhadap *face spoofing attack* dalam sistem kehadiran yang terintegrasikan dengan teknologi *face recognition*.

3.1.2 Studi Deskriptif I

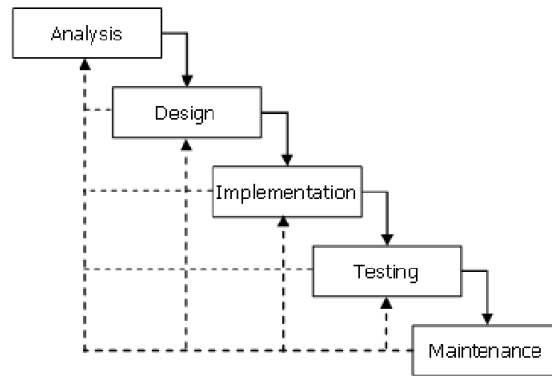
Tahapan kedua adalah mengidentifikasi permasalahan dan tujuan penelitian dengan melihat adanya penelitian terdahulu yang memiliki masalah dan tujuan penelitian yang relevan. Permasalahan yang diangkat berupa pertanyaan-pertanyaan yang akan diselesaikan dengan penelitian dengan latar belakang yang telah dibuat.

3.1.3 Studi Perskriptif

Tahapan ketiga adalah melakukan pembangunan sistem dan perangkat lunak. Setidaknya terdapat dua elemen yang dikembangkan dalam penelitian ini. Pertama, sistem *liveness detection* atau pendeteksi kehidupan yang dibangun dari pustaka/*library* Google Machine Learning Kit (ML-Kit) untuk kerangka kerja/*framework* Flutter. Kedua, sistem kehadiran berbasis *mobile cross-platform* menggunakan teknologi *face recognition* yang diintegrasikan dengan sistem *liveness detection*. Oleh karena itu, dua elemen ini dipisahkan dalam penjelasan proses pengembangan sistem.

Untuk kedua tahapan tersebut, penulis akan menggunakan model pengembangan perangkat lunak Waterfall. Model SDLC Waterfall adalah proses pengembangan perangkat lunak berurutan di mana kemajuan dianggap mengalir

semakin ke bawah (mirip dengan air terjun) melalui daftar fase yang harus dijalankan untuk berhasil membangun perangkat lunak komputer. Model Waterfall diusulkan oleh Winston W. Royce pada tahun 1970 untuk menjelaskan praktik rekayasa perangkat lunak yang ada (Bassil, 2012). Berikut adalah ilustrasi dan penjelasan mengenai model SDLC Waterfall dijelaskan pada Gambar 3.2.



Gambar 3.2 Model SDLC Waterfall

1) Analysis

Pada fase analisis, seringkali dilakukan pendefinisian dari kebutuhan sistem perangkat lunak yang mencakup perilaku sistem yang akan dikembangkan. Kemudian, dalam fase ini ditentukan juga bagaimana seharusnya kebutuhan fungsional dan non-fungsional agar spesifikasi dan kualitas dari sistem perangkat lunak dapat ditentukan dan dibangun secara kokoh dari awal tahapan pengembangannya. Kebutuhan fungsional merupakan instruksi atau fitur-fitur yang seharusnya ada di dalam suatu sistem, sedangkan kebutuhan non-fungsional adalah bagaimana suatu sistem dapat bekerja dengan baik yang dapat dinilai dari kehandalan, kinerja, dan mudahnya penggunaan dari sistem tersebut (Dabbagh & Lee, 2015).

2) Design

Dalam tahapan desain, sistem perangkat lunak yang dikembangkan akan ditentukan seperti rencana solusi dari permasalahan yang ada mencakup algoritma, desain struktur perangkat lunak, desain antarmuka, hingga pendefinisian dari struktur data.

3) Implementation

Pada fase implementasi, semua kebutuhan sistem perangkat lunak yang telah didefinisikan beserta hasil dari fase desain, diubah dan dibentuk menjadi sebuah program dari kode-kode pemrograman yang dikompilasi menjadi sebuah sistem perangkat lunak yang dapat dijalankan oleh pengguna.

4) Testing

Dalam fase pengujian, terjadi verifikasi dan validasi untuk memeriksa apakah sistem perangkat lunak telah memenuhi kebutuhan dan spesifikasi yang diharapkan dan didefinisikan pada fase-fase sebelumnya. Proses verifikasi dan validasi pada penelitian ini menggunakan pengujian *blackbox* yang dilakukan secara manual untuk menguji setiap *test case* yang dimiliki sistem *liveness detection* maupun sistem kehadiran berbasis *mobile cross-platform*.

5) Maintenance

Proses terakhir dalam model pengembangan sistem perangkat lunak yang memiliki tujuan untuk melakukan berbagai pengembangan lanjutan setelah terjadi fase pengujian. Hal ini dimaksudkan untuk meningkatkan performa dan kehandalan dari sistem perangkat lunak yang dibangun.

3.1.4 Studi Deskriptif II

Tahapan terakhir adalah melakukan pengujian pada sistem dan perangkat lunak. Pada tahapan ini, ditetapkan mahasiswa program studi Rekayasa Perangkat Lunak, Universitas Pendidikan Indonesia dijadikan sebagai data uji dalam pengujian penelitian sebanyak 30 mahasiswa. Selanjutnya, pengujian sistem dan perangkat lunak ini menggunakan 4 (empat) skenario yang dapat merepresentasikan kemampuan dari sistem yang diteliti merujuk pada penelitian oleh (Widjaya & Wicaksana, 2023). Berikut adalah uraian dari skenario yang digunakan pada pengujian sistem dan perangkat lunak pada Tabel 3.1.

Tabel 3.1
Skenario Pengujian

No	Skenario	Deskripsi	Keterangan
1	Skenario I	Mahasiswa diharuskan melakukan kehadiran dengan wajah asli mereka. Hasil yang diharapkan adalah wajah mereka dapat diidentifikasi sebagai wajah asli dan berhasil melakukan kehadiran.	Seluruh mahasiswa diberikan waktu untuk kehadiran selama 20 menit.
2	Skenario II	Mahasiswa diharuskan melakukan kehadiran dengan wajah dari hasil video tiruan dengan gerakan-gerakan yang terdapat di sistem <i>liveness detection</i> yang diteliti ini. Hasil yang diharapkan adalah mereka tidak dapat melakukan kehadiran karena wajah mereka tidak dapat diverifikasi sebagai wajah asli.	Pengujian dengan wajah hasil dari perekaman video tiruan ini diberikan kesempatan sebanyak 3 kali dalam kurun waktu 20 menit.
3	Skenario III	Mahasiswa diharuskan untuk melakukan kehadiran dengan wajah asli mereka. Hasil yang diharapkan adalah mereka tidak dapat melakukan kehadiran dan tidak dapat diverifikasi sebagai wajah asli.	Seluruh mahasiswa diberikan waktu untuk kehadiran selama 20 menit.
4	Skenario IV	Mahasiswa diharuskan melakukan kehadiran dengan wajah dari hasil video tiruan dengan gerakan-gerakan yang terdapat di sistem <i>liveness detection</i> yang diteliti ini. Harapannya adalah mahasiswa	Pengujian dengan wajah hasil dari perekaman video tiruan ini diberikan kesempatan

No	Skenario	Deskripsi	Keterangan
		dapat melakukan kehadiran dan wajah mereka dapat diverifikasi.	sebanyak 3 kali dalam kurun waktu 20 menit..

Adapun hasil dari pengujian yang direpresentasikan melalui beberapa skenario seperti yang terdapat pada tabel 3.1 kemudian dianalisis *performance metrics* seperti *accuracy*, *precision*, *recall*, dan F-Score untuk mengevaluasi performa dari sistem yang diujikan (Hossin & Sulaiman, 2015).

3.2 Perancangan Sistem Liveness Detection

Sistem *Liveness Detection* yang akan dibangun dengan kerangka kerja Flutter dengan bahasa pemrograman Dart yang akan menjadi sebuah *plugin* / modul mandiri yang dapat digunakan dalam sistem kehadiran digital yang akan dikembangkan dan diteliti. Sistem ini mendeteksi tanda kehidupan / *life sign* sebagai pencegahan *face spoofing attack* yang merujuk pada penelitian (S. Chakraborty & Das, 2014) yang menyatakan bahwa klasifikasi pengenalan wajah dapat dilihat dari kemampuan berkedip, menggerakkan bibir, dan ekspresi wajah sebagai cara identifikasi suatu makhluk hidup dinyatakan memiliki tanda kehidupan serta merujuk pada penelitian (Widjaya & Wicaksana, 2023) untuk metode *liveness detection* yang digunakan, yaitu metode *randomized challenge response*.

Sistem *Liveness Detecion* ini akan memiliki enam tantangan yang akan memvalidasi apakah suatu wajah yang sedang dideteksi merupakan wajah asli atau tipuan. Berikut adalah enam tantangan yang terdapat pada sistem *liveness detection*:

Tabel 3.2
Tantangan *Liveness Detection* untuk Menandakan Tanda Kehidupan

Tantangan	Deskripsi
Senyum	Pengguna ditantang untuk tersenyum dihadapan kamera ketika proses <i>liveness detection</i> .
Berkedip	Pengguna diharuskan untuk berkedip sebanyak 1-2 kali untuk menandai bahwa wajah yang sedang dideteksi merupakan objek hidup dan bukan tiruan

Tantangan	Deskripsi
Menoleh ke atas (<i>Nodding up</i>)	Pengguna ditantang untuk menolehkan kepala mereka ke atas dengan sudut tertentu agar dapat dideteksi sebagai objek yang dapat bergerak dan terhindar dari potensi <i>photo</i> dan <i>video attack</i>
Menoleh ke bawah (<i>Nodding down</i>)	Pengguna ditantang untuk menolehkan kepala mereka ke bawah dengan sudut tertentu agar dapat dideteksi sebagai objek yang dapat bergerak dan terhindar dari potensi <i>photo</i> dan <i>video attack</i>
Menoleh ke kanan (<i>Shaking right</i>)	Pengguna ditantang untuk menolehkan kepala ke kanan dengan sudut tertentu agar dapat dideteksi sebagai objek yang dapat bergerak dan terhindar dari potensi <i>photo</i> dan <i>video attack</i>
Menoleh ke kiri (<i>Shaking left</i>)	Pengguna ditantang untuk menolehkan kepala ke kiri dengan sudut tertentu agar dapat dideteksi sebagai objek yang dapat bergerak dan terhindar dari potensi <i>photo</i> dan <i>video attack</i>

Sistem ini akan memberikan tantangan kepada objek wajah yang dideteksi untuk melakukan tantangan yang memiliki urutan acak (*randomized challenge-response*). Jika berhasil, maka sistem akan mengembalikan suatu model berupa foto wajah yang sudah teridentifikasi sebagai sebuah gambar wajah manusia asli. Hal ini didasari oleh keberhasilan pengguna melewati semua tantangan yang ada. Jika dalam proses verifikasi sistem kehilangan deteksi wajah karena objek menghilang pada daerah area cangkupan penangkapan objek wajah melalui kamera, maka proses *liveness detection* akan kembali ke kondisi awal dimana tantangan pertama dimulai. Dalam proses verifikasinya, sistem *liveness detection* diberikan batasan waktu selama 45 detik. Hal ini bertujuan agar mencegah aktivitas mencurigakan dalam proses verifikasi wajah pada sistem *liveness detection*.

Setiap tantangan, akan diberikan ambang batas/*threshold* sebagai penanda batas minimum suatu objek bergerak sebagai penanda kehidupan. Jika hasil yang dideteksi melebihi *threshold* yang ditetapkan, maka objek yang dideteksi dapat

diidentifikasi sebagai wajah asli yang hidup, bukan wajah tiruan (S. Chakraborty & Das, 2014). Penetapan ambang batas / *threshold* akan berdasarkan ketentuan dari *library* Google Machine Learning Kit (Bernal & Biradar, 2022) sebagai berikut pada Gambar 3.3:

```
final double? rotX = face.headEulerAngleX; // Head is tilted up and down rotX degrees
final double? rotY = face.headEulerAngleY; // Head is rotated to the right rotY degrees
final double? rotZ = face.headEulerAngleZ; // Head is tilted sideways rotZ degrees
```

Gambar 3.3 Ketentuan Sumbu Rotasi Gerakan Wajah

Berdasarkan dokumentasi tersebut, sumbu X berfungsi untuk mengetahui berapa derajat kepala dihadapkan ke atas atau kebawah. Nilai sudut derajat positif pada sumbu X menandakan posisi kepala sedang menoleh keatas, sedangkan nilai negatif sudut derajat pada sumbu X menandakan posisi kepala sedang menoleh kebawah. Kemudian, sumbu Y menandakan posisi kepala sedang menghadap ke kanan atau kiri. Jika nilai derajat sumbu Y positif, maka kepala yang terdeteksi diidentifikasi menoleh ke kanan. Sebaliknya, jika nilai derajat sumbu Y negatif, kepala yang terdeteksi dianggap menoleh ke kiri. Sehingga, penulis menentukan *threshold* pada sistem *liveness detection* :

Tabel 3.3
Tingkat *threshold* untuk setiap tantangan *liveness detection*

Tantangan	Threshold
Senyum	0.75 (75%) dari skala 0 s.d 1
Berkedip	0.25 (0.25%) dari skala 0 s.d 1
Menoleh ke atas (<i>Nodding up</i>)	20° dari sumbu X
Menoleh ke bawah (<i>Nodding down</i>)	-20° dari sumbu X
Menoleh ke kanan (<i>Shaking right</i>)	-30° dari sumbu Y
Menoleh ke kiri (<i>Shaking left</i>)	30° dari sumbu Y

3.3 Perancangan Sistem Kehadiran Berbasis Mobile Cross-Platform

Sistem kehadiran berbasis *mobile cross-platform* yang dibangun dan diteliti pada penelitian ini bernama My Attendance. Sistem ini menggunakan kerangka kerja Flutter dengan bahasa Dart sebagai fondasi dasar pembangunan perangkat lunak dengan arsitektur MVC (Model-View-Controller) serta *state management* Provider. Sistem ini akan melibatkan dua peran utama dalam suatu proses kehadiran pada kelas dalam perkuliahan, yakni dosen dan mahasiswa. Sehingga, sistem ini dapat digunakan oleh dua jenis peran sekaligus dalam 1 (satu) sistem yang sama.

My Attendance adalah sebuah sistem perangkat lunak yang dapat digunakan pada perangkat *mobile* dengan berbagai sistem operasi seperti Android dan iOS. Sistem ini memiliki fungsi utama sebagai media kehadiran berbasis digital yang dapat digunakan oleh mahasiswa dan dosen dalam suatu kelas perkuliahan yang dilakukan setiap pekan perkuliahan berlangsung. Bagi peran mahasiswa, sistem ini memiliki fungsi utama sebagai media kehadiran digital dengan memanfaatkan teknologi *face recognition* yang terintegrasi dengan sistem *liveness detection* yang dapat meminimalisir adanya kecurangan *face spoofing attack* pada proses kehadiran. Sedangkan bagi peran dosen, sistem ini memiliki fungsi utama sebagai media pembukaan kelas mata kuliah untuk perkuliahan yang akan dilaksanakan. Mahasiswa hanya bisa melakukan kehadiran jika dosen telah membuka pertemuan kelas serta di waktu tertentu yang masih diperkenankan melakukan kehadiran seperti halnya ketika proses dilakukan secara manual. Proses kehadiran tidak dapat dilakukan apabila diluar waktu yang ditentukan oleh dosen saat pembukaan kelas.

3.4 Instrumen Penelitian

Adapun instrumen penelitian yang akan digunakan penulis untuk mendukung penelitian yang akan dilakukan antara lain:

3.4.1 Perangkat Seluler

Pada penelitian ini, perangkat seluler dimanfaatkan untuk tahapan pengembangan dan pengujian sistem dan perangkat lunak yang dibangun. Penulis menggunakan dua perangkat seluler dengan tingkatan versi sistem operasi berbeda yang dijelaskan secara detail pada tabel 3.4 dibawah:

Tabel 3.4
Perangkat Seluler Pertama yang Akan Digunakan Dalam Fase Pengembangan

Spesifikasi Perangkat	Deskripsi
Nama Perangkat	Xiaomi Redmi Note 8
Chipset & CPU	Qualcomm Snapdragon 665 11nm & Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)
RAM	4 GB
Sensor Kamera Depan	13 MP, f/2.0, (wide), 1/3.1", 1.12µm
Sistem Operasi	Android 11 (Red Velvet Cake)

3.4.2 Alat Pendukung

Pada penelitian ini, alat pendukung berupa perangkat lunak dan *library* (pustaka) yang bertujuan untuk pengembangan sistem *liveness detection* serta sistem kehadiran berbasis *mobile cross-platform*.

Tabel 3.5
Daftar Perangkat Lunak yang Akan Digunakan Dalam Penelitian

No.	Nama PL	Deskripsi
1	Android Studio Flamingo 2022.22.1	Perangkat lunak IDE ini digunakan untuk pengembangan sistem <i>liveness detection</i> serta sistem kehadiran
2	Flutter Framework 3.7.12	Kerangka kerja Flutter digunakan untuk <i>environment</i> pengembangan sistem kehadiran dan sistem <i>liveness detection</i>
3	Dart 2.19.6	Bahasa pemrograman yang digunakan dalam pengembangan sistem <i>liveness detection</i> dan sistem kehadiran dengan kerangka kerja Flutter
4	Express JS	Kerangka kerja yang digunakan untuk merancang dan membangun sistem <i>face</i>

recognition yang berbentuk arsitektur REST API

Tabel 3.6
Daftar Pustaka yang Akan Digunakan Dalam Pengembangan dan Pengujian Sistem dan Aplikasi

No.	Nama Pustaka	Deskripsi
1	Google ML-Kit Face Detection ^0.5.0	Digunakan untuk sistem <i>liveness detection</i> pada sistem kehadiran untuk pengenalan wajah
2	Image ^4.0.15	Digunakan dalam <i>image processing</i> pada sistem <i>liveness detection</i>
3	Face API JS	Digunakan sebagai pengenalan wajah saat proses kehadiran melalui sistem dalam bentuk REST API

Tabel 3.7
Daftar Layanan yang Akan Digunakan Dalam Pengembangan Sistem dan Aplikasi

No.	Nama Layanan	Deskripsi
1	Supabase	Merupakan layanan penyedia <i>database</i> berbasis <i>online</i> dengan sistem manajemen PostgreSQL. Layanan ini juga menyediakan layanan REST API agar pengguna dapat mengakses data dengan <i>http request</i> dengan kredensial tertentu yang membuatnya menjadi aman. Layanan ini digunakan untuk menyimpan data aktivitas sistem kehadiran berbasis <i>mobile cross-platform</i> seperti data kehadiran dan log data kegagalan percobaan kehadiran. Terakhir, layanan ini digunakan untuk keperluan <i>authentication</i> pengguna.
2	Firebase	Merupakan layanan basis data dengan manajemen NO-SQL. Layanan ini digunakan

Bagus Subagja, 2024

PENGEMBANGAN LIVENESS DETECTION UNTUK PENCEGAHAN FACE SPOOFING ATTACK PADA SISTEM KEHADIRAN BERBASIS MOBILE CROSS-PLATFORM: STUDI KASUS MAHASISWA RPL UPI CIBIRU

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

		untuk keperluan debugging saat pengembangan sistem kehadiran
2	MongoDB	Merupakan layanan penyedia <i>database</i> berbasis <i>online</i> dengan sistem manajemen NoSQL. Layanan ini digunakan untuk menyimpan data model wajah yang telah dilatih pada sistem <i>face recognition</i>

3.4.3 Skenario Pengujian

Untuk mendapatkan kualitas performa dari sistem *liveness detection* yang digunakan pada sistem kehadiran, maka diperlukan data yang diperoleh langsung dari partisipan penelitian. Skenario pengujian diambil dari poin studi deskriptif II (lihat subbab 3.1.4) dengan rujukan dari penelitian yang dilakukan oleh (Widjaya & Wicaksana, 2023) sebagai berikut.

Tabel 3.8
Skenario Pengujian

No	Label	Skenario Pengujian
1	Skenario I	Saya berhasil melakukan kehadiran yang melibatkan wajah asli
2	Skenario II	Saya tidak berhasil melakukan kehadiran dengan memberikan video tiruan gerakan wajah
3	Skenario III	Saya tidak berhasil melakukan kehadiran dengan melibatkan wajah asli
4	Skenario IV	Saya berhasil melakukan kehadiran dengan memberikan video tiruan gerakan wajah

Setiap poin pertanyaan dari skenario pengujian merepresentasikan metrik untuk mengevaluasi kualitas performa yang dihasilkan oleh sistem *liveness detection* dengan evaluasi *performance metrics* seperti *True Positive (TP)*, *True Negative (TN)*, *False Negative (FN)*, dan *False Positive (FP)*.

3.5 Teknik Analisis Data

Data yang akan dianalisis pada penelitian ini didapat dari skenario pengujian yang dilakukan berdasarkan desain penelitian studi deskriptif II (lihat subbab 3.1.4) untuk mendapatkan nilai *performance metrics* untuk mengukur seberapa baik sistem *liveness detection* bekerja. Berikut adalah representasi nilai dari *performance metrics* berdasarkan skenario pengujian yang dilakukan pada tabel 3.9.

Tabel 3.9 Nilai *Performance Metrics* Dengan Skenario Pengujian

No	Skenario	Nilai
1	Skenario I	<i>True Positive (TP)</i>
2	Skenario II	<i>True Negative (TN)</i>
3	Skenario III	<i>False Negative (FN)</i>
4	Skenario IV	<i>False Positive (FP)</i>

Analisis data pada penelitian ini akan menggunakan evaluasi *performance metrics* yakni *accuracy*, *precision*, *recall*, dan F-score yang dijabarkan dalam persamaan berikut (Dalianis, 2018) pada subbab 3.5.1 – subbab 3.5.4.

3.5.1 Accuracy

Accuracy merupakan salah satu penilaian dari *performance metrics* yang dapat memberikan informasi seberapa akurat rasio prediksi positif dan negatif dengan keseluruhan data atau tingkat kedekatan antara nilai prediksi dengan nilai aktual. Dengan kata lain, akurasi menjadi *metrics* yang dapat menyatakan tingkat akurasi dari sistem *liveness detection* yang dapat membedakan antara wajah asli dan wajah tiruan dengan persamaan berikut.

$$\frac{TP+TN}{TP+FP+TN+FN} \dots\dots\dots (3.1)$$

3.5.2 Precision

Precision adalah salah satu elemen dari *performance metrics* yang digunakan dalam menilai rasio prediksi benar positif dengan keseluruhan hasil yang diprediksi positif. Dengan kata lain, *precision* adalah metrik yang menjelaskan seberapa akurat sistem mendeteksi wajah asli. Nilai metrik ini dapat diperoleh melalui persamaan berikut.

Bagus Subagja, 2024

PENGEMBANGAN LIVENESS DETECTION UNTUK PENCEGAHAN FACE SPOOFING ATTACK PADA SISTEM KEHADIRAN BERBASIS MOBILE CROSS-PLATFORM: STUDI KASUS MAHASISWA RPL UPI CIBIRU

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

$$\frac{TP}{TP+FP} \dots\dots\dots (3.2)$$

3.5.3 Recall

Recall merupakan salah satu elemen dari *performance metrics* yang digunakan dalam menilai metrik prediksi benar positif dengan keseluruhan nilai data yang benar positif. Dengan kata lain, *recall* merupakan metrik yang menjelaskan seberapa lengkap sistem dapat mendeteksi wajah asli. Nilai metrik ini dapat diperoleh melalui persamaan berikut.

$$\frac{TP}{TP+FN} \dots\dots\dots (3.3)$$

3.5.4 F-score

F-Score merupakan nilai harmonisasi atau nilai tengah dari nilai metrik *precision* dan *recall*. Nilai metrik ini dapat diperoleh dari persamaan berikut.

$$\frac{2*recall*precision}{recall+precision} \dots\dots\dots (3.4)$$