

**SKRIPSI**

**RANCANG BANGUN SISTEM DETEKSI KEHADIRAN  
OTOMATIS PADA MEDIA PEMBELAJARAN VIDEO  
ONLINE MENGGUNAKAN METODE PENGENALAN  
WAJAH**

**Disusun dan diajukan Oleh**

**MUHAMMAD MUFLIHUN NAIM**

**H071171007**



**PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2021**

**RANCANG BANGUN SISTEM DETEKSI KEHADIRAN  
OTOMATIS PADA MEDIA PEMBELAJARAN VIDEO  
ONLINE MENGGUNAKAN METODE PENGENALAN  
WAJAH**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains pada  
Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan  
Ilmu Pengetahuan Alam Universitas Hasanuddin**

**MUHAMMAD MUFLIHUN NAIM**

**H071171007**

**PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2021**

# PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Muflihun Naim  
NIM : H071171007  
Program Studi : Sistem Informasi  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**Rancang Bangun Sistem Deteksi Kehadiran Otomatis Pada Media Pembelajaran Video Online Menggunakan Metode Pengenalan Wajah**

adalah karya tulisan saya sendiri dan bukan merupakan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Mekasasa, 1 Desember 2021  
takan,



**Muhammad Muflihun Naim**  
NIM: H071171007



**RANCANG BANGUN SISTEM DETEKSI KEHADIRAN  
OTOMATIS PADA MEDIA PEMBELAJARAN VIDEO  
ONLINE MENGGUNAKAN METODE PENGENALAN  
WAJAH**

Disusun dan diajukan oleh

**MUHAMMAD MUFLIHUN NAIM**

**H071171007**

Telah diperhatikan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam pada 1 Desember 2021 dan dinyatakan telah memenuhi syarat kelulusan

**UNIVERSITAS HASANUDDIN**

**Menyetujui,**

**Pembimbing Utama,**

**Pembimbing Pertama,**

**Dr. Eng. Armin Lawi, S.Si., M.Eng.**  
**NIP. 197204231995121001**

**Supri Bin Hj Amir, S.Si., M.Eng.**  
**NIP. 198805042019031012**

**Ketua Program Studi,**

**Dr. Muhammad Hasbi, M.Sc.**  
**NIP. 196307201989031003**



## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Muhammad Muflihun Naim  
NIM : H071171007  
Program Studi : Sistem Informasi  
Judul Skripsi : Rancang Bangun Sistem Deteksi Kehadiran Otomatis  
Pada Media Pembelajaran Video Online Menggunakan  
Metode Pengenalan Wajah

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Tanda Tangan

Ketua : Dr. Eng. Armin Lawi, S.Si., M.Eng. (.....)  
Sekretaris : Supri Bin Hj Amir, S.Si., M.Eng. (.....)  
Anggota : Dr. Muhammad Hasbi, M.Sc. (.....)  
Anggota : Dr. Hendra, S.Si., M.Kom. (.....)

Ditetapkan di : Makassar

Tanggal : 1 Desember 2021





## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT. karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Strata 1 yang berjudul **“Rancang Bangun Sistem Deteksi Kehadiran Otomatis pada Media Pembelajaran Video Online Menggunakan Metode Pengenalan Wajah”**. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini dengan segala kerendahan hati penulis menyampaikan terima kasih yang setulus-tulusnya kepada:

1. Keluarga, ayahanda tercinta **Drs. Abd. Hafid** dan ibunda tersayang **Dr. Nurtaqwa Amin, M.Hum.** yang telah memberikan dukungan baik moril maupun materiil serta doa yang tiada henti-hentinya kepada penulis.
2. Rektor Universitas Hasanuddin beserta jajarannya, Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam beserta jajarannya, dan seluruh pihak birokrasi atas pengetahuan yang diberikan, baik dalam bidang akademik maupun bidang kemahasiswaan.
3. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku pembimbing utama dan Bapak **Supri Bin Hj Amir, S.Si., M.Eng.** selaku pembimbing pertama untuk segala ilmu dan kesabaran dalam membimbing dan mengarahkan penulis, serta bersedia meluangkan waktunya untuk mendampingi penulis sehingga skripsi ini dapat terselesaikan. Serta kepada **Bapak Dr. Muhammad Hasbi, M.Sc.** selaku Ketua Program Studi Sistem Informasi sekaligus dan Bapak **Dr. Hendra, S.Si., M.Kom.** atas kesediaannya menjadi anggota tim penguji yang telah bersedia meluangkan waktunya untuk memberikan saran dan arahan kepada penulis dalam penyusunan skripsi ini.
4. Dosen Departemen Matematika, dan terkhusus kepada ibu dan bapak dosen Program Studi Sistem Informasi Fakultas MIPA Universitas Hasanuddin untuk semua ilmu yang telah diberikan kepada penulis selama menempuh pendidikan di jenjang strata.
5. Pegawai dan staf Science Building dan Departemen Matematika Fakultas MIPA Universitas Hasanuddin yang senantiasa membantu pengurusan dokumen.
6. Teman-teman seperjuangan dari Program Studi Sistem Informasi 2017 khususnya kepada **Ilmi Kalam, S.Kom.** yang senantiasa mendukung, memberikan semangat,

pelajaran-pelajaran dan motivasi hidup, **Khawaritzmi Abdallah Ahmad, S.Kom.** yang senantiasa meluangkan waktunya untuk belajar bersama, **Muh. Arizki, S.Kom.** dan **Muhammad Fitrah** yang senantiasa membawakan kunci untuk membuka ruang belajar Sistem Informasi, **Muh. Fiqih Hamda, Kennedy, Muh. Fadhil, Rigel Rivaldo Subiyakto, Muhammad Nur, Restu Adi Akbar, Effendy, Arya Indrawan,** dan teman-teman Sistem Informasi 2017 lainnya.

7. Kakak-kakak dan adik-adik Program Studi Sistem Informasi 2014, 2015, 2016, 2018, 2019, dan 2020.
8. Komunitas Stack Overflow dan Github yang telah membantu penulis dan programmer lain di belahan dunia lainnya untuk menemukan jalan keluar dari setiap permasalahan dalam menyusun kode program.
9. Semua pihak yang telah membantu penulis dan tak sempat penulis tuliskan satu persatu.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Makassar, 1 Desember 2021

Penulis

## **PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Muflihun Naim  
NIM : H071171007  
Program Studi : Sistem Informasi  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin. **Hak Bebas Royalti Noneksklusif (Non-exclusive-Royalty-Free Right)** atas karya ilmiah saya yang berjudul:

### **Rancang Bangun Sistem Deteksi Kehadiran Otomatis pada Media Pembelajaran Video Online Menggunakan Metode Pengenalan Wajah**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 1 Desember 2021

Yang menyatakan,



(Muhammad Muflihun Naim)



## ABSTRAK

Perolehan data kehadiran peserta didik oleh pengajar dalam sebuah pembelajaran video online dengan cara tradisional hanya memperoleh data kehadiran hanya pada saat absensi dilakukan, bukan data kehadiran yang sesungguhnya dalam pembelajaran video online penuh. Ini tidak bisa menjamin apakah peserta didik / siswa benar-benar hadir sepenuhnya dalam pembelajaran video online. Sistem yang diusulkan dalam penelitian ini menggunakan teknologi *Pengenalan Wajah* untuk pendeteksian kehadiran siswa secara otomatis menggunakan *Faceapi.js*. Metode pengumpulan data yang digunakan adalah metode Studi Pustaka, sedangkan metode pengembangan sistem menggunakan metode *Waterfall*. Hasil dari penelitian ini adalah aplikasi web berbasis *Firebase* sebagai media pembelajaran video online menggunakan *Jitsi Meet IFrame API*, dengan fitur pendeteksian kehadiran otomatis menggunakan teknologi pengenalan wajah dari *library Faceapi.js*. Aplikasi diuji menggunakan perangkat Laptop dengan kapasitas RAM 8 GB, Intel Core i5 2.11 GHz. Aplikasi web memiliki performa *Throughput* yang cukup berdasarkan standar TIPHON dengan nilai 707,62 kbps, performa *latency* yang sangat baik berdasarkan standar TIPHON dengan nilai rata-ratanya yaitu 0,535 ms untuk operasi baca dan 0,962 ms untuk operasi tulis, performa Response Time yang baik berdasarkan standar Google dengan nilai rata-ratanya di semua kategori kecepatan jaringan yaitu 523,614 ms, penggunaan *memory* tertinggi yaitu ketika aplikasi mendeteksi wajah user pertama kali yaitu sebanyak yaitu sebanyak 120,844 MB, penggunaan *CPU* tertinggi yaitu ketika aplikasi mendeteksi wajah user selain dari pendeteksian pertama yaitu sebanyak 15,6%.

Kata Kunci: Deteksi Kehadiran Otomatis, Pembelajaran Video Online, Pengenalan Wajah.

## ABSTRACT

The acquisition of student attendance data by the teacher in an online video learning in the traditional way only obtains attendance data only when attendance is being done, not actual attendance data in full online video learning. This can not guarantee whether the students are really fully present in online video learning. The system proposed in this study uses *Facial Recognition* Technology to detect student presence automatically using *Faceapi.js*. The data collection method used is the Literature Study method, and the system development method used is the Waterfall method. The results of this study are a *Firebase-based* web application as an online video learning medium using the *Jitsi Meet IFrame API*, with an automatic presence detection feature using *Facial Recognition* technology from the *Faceapi.js* library. The application was tested using a Laptop device with a RAM capacity of 8 GB, Intel Core i5 2.11 GHz. The web application has sufficient throughput performance based on the TIPHON standard with a value of 707.62 kbps, very good latency performance based on the TIPHON standard with an average value of 0.535 ms for read operations and 0.962 ms for write operations, good Response Time performance based on standards Google with the average value in all categories of network speed is 523,614 ms, the highest memory usage is when the application detects the user's face for the first time, which is 120.844 MB, the highest CPU usage is when the application detects the user's face other than the first detection, which is 15.6%.

Keywords: Automatic Presence Detection, Online Video Learning, Facial Recognition.

## DAFTAR ISI

SAMPUL	
HALAMAN JUDUL .....	i
PERNYATAAN KEASLIAN .....	ii
HALAMAN PERSETUJUAN PEMBIMBING .....	iii
HALAMAN PENGESAHAN .....	iv
KATA PENGANTAR.....	v
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR.....	vii
ABSTRAK .....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL .....	xvi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Pembelajaran Daring.....	5
2.2. Aplikasi Web dan Video Conference.....	5
2.3. Jitsi Meet .....	6
2.4. Face-api.js .....	6
2.5. Face Recognition.....	7
2.5.1. SSD Mobilenet v1 Model .....	9
2.5.2. Face-Recognition Model.....	9
2.6. Euclidean Distance.....	11



2.7.	Firestore .....	13
2.7.1.	Firestore Authentication .....	13
2.7.2.	<i>Realtime Database</i> .....	13
2.7.3.	<i>Cloud Firestore</i> .....	14
2.8.	Penelitian Terkait .....	15
BAB III METODE PENELITIAN .....		18
3.1.	Metode Pengumpulan Data .....	18
3.1.1.	Studi Pustaka .....	18
3.2.	Metode Pengembangan Sistem .....	18
3.2.1.	Analisis Kebutuhan Sistem .....	19
3.2.2.	Perancangan Sistem .....	19
3.2.3.	Implementasi Sistem dan Deployment .....	19
3.2.4.	Pengujian Sistem .....	20
3.3.	Alur Penelitian .....	20
3.4.	Waktu dan Tempat Penelitian .....	21
BAB IV HASIL DAN PEMBAHASAN .....		22
4.1.	Analisis Kebutuhan Sistem .....	22
4.1.1.	Analisis Sistem yang Berjalan .....	22
4.1.2.	Identifikasi Masalah .....	22
4.1.3.	Analisis Sistem yang Diusulkan .....	23
4.1.4.	Kebutuhan Perangkat Keras .....	24
4.1.5.	Kebutuhan Perangkat Lunak .....	24
4.2.	Perancangan Sistem .....	24
4.2.1.	Use Case Diagram .....	24
4.2.2.	<i>Use Case Scenario</i> .....	26
4.2.3.	Activity Diagram .....	30
4.2.4.	Deployment Diagram .....	34

4.2.5. Class Diagram.....	35
4.2.6. Perancangan Database .....	36
4.2.7. Perancangan Interface.....	37
4.3. Implementasi dan Deployment .....	40
4.3.1. Implementasi Database .....	40
4.3.2. Implementasi Interface .....	45
5.1.4. Deployment.....	71
4.4. Pengujian Sistem.....	72
4.2.1. Pengujian Fungsional.....	72
4.2.2. Pengujian Performa.....	81
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>88</b>
5.1. Kesimpulan .....	88
5.2. Saran.....	89
<b>DAFTAR PUSTAKA.....</b>	<b>90</b>
<b>LAMPIRAN .....</b>	<b>93</b>

## DAFTAR GAMBAR

Gambar 2.1. Jarak Euclid di $R^2$ .....	11
Gambar 2.2. Contoh penggunaan Jarak Euclid untuk menghitung kemiripan dua citra wajah.....	12
Gambar 2.3. Struktur data Realtime Database .....	14
Gambar 2.4. Struktur data Firestore .....	14
Gambar 3.1. Tahap-tahap pengembangan sistem Metode Waterfall .....	18
Gambar 3.2. Alur Penelitian.....	20
Gambar 4.1. Alur sistem yang sedang berjalan.....	22
Gambar 4.2. Alur sistem yang diusulkan .....	23
Gambar 4.3. Use Case Diagram .....	25
Gambar 4.4. Activity diagram Create Classroom (sisi kiri) dan Activity Diagram Join Classroom & Verify Presence Status (sisi kanan).....	30
Gambar 4.5. Activity diagram Login (sisi kiri) dan Register (sisi kanan) .....	31
Gambar 4.6. Activity diagram Face-Register.....	32
Gambar 4.7. Activity Diagram Check Classroom Attendance.....	33
Gambar 4.8. Firebase menggantikan infrastruktur sisi server tradisional .....	34
Gambar 4.9. Deployment Diagram .....	34
Gambar 4.10. Class Diagram.....	35
Gambar 4.11. Perancangan Database .....	36
Gambar 4.12. Contoh Aturan Keamanan Firestore .....	37
Gambar 4.13. Perancangan interface halaman Login.....	37
Gambar 4.14. Perancangan interface halaman Register .....	38
Gambar 4.15. Perancangan interface halaman Home for Teacher.....	38
Gambar 4.16 Perancangan interface halaman Home for Student.....	39
Gambar 4.17. Perancangan interface halaman Classroom meeting .....	39
Gambar 4.18. Perancangan interface halaman Detail Classroom .....	40
Gambar 4.19. Koleksi rooms dalam Firestore.....	41
Gambar 4.20. Koleksi users dalam Firestore.....	41
Gambar 4.21. Koleksi attendances dalam Realtime Database .....	42
Gambar 4.22. Koleksi users_identifiers dalam Realtime Database .....	42
Gambar 4.23. Implementasi Firebase Authentication .....	44
Gambar 4.24. Halaman Login .....	46



Gambar 4.25. Halaman Register .....	48
Gambar 4.26. Halaman Complete Profile untuk Student .....	51
Gambar 4.27. Halaman Complete Profile untuk Teacher .....	51
Gambar 4.28. Diagram arsitektur client pada Face Register.....	52
Gambar 4.29. Halaman Face Register .....	56
Gambar 4.30. Halaman Home untuk Role Teacher .....	59
Gambar 4.31. Halaman Home untuk Role Student .....	59
Gambar 4.32. Diagram arsitektur client-side aplikasi pada proses pertemuan Classroom .....	61
Gambar 4.33. Tampilan IFrame Jitsi Meet untuk Teacher.....	65
Gambar 4.34. Tampilan IFrame Jitsi Meet untuk Student .....	66
Gambar 4.35. Tampilan Create A Classroom .....	66
Gambar 4.36. Halaman History untuk Teacher.....	69
Gambar 4.37. Halaman History untuk Student .....	69
Gambar 4.38. Halaman Detail Classroom.....	71
Gambar 4.39. Peringatan halaman Detail Classroom hanya untuk Teacher .....	71
Gambar 4.40. Tampilan menu Hosting dalam Firebase Console.....	72
Gambar 4.41. Hasil pengujian kasus ketika aplikasi tidak menampilkan tombol Create Classroom jika role adalah Student .....	73
Gambar 4.42. Hasil pengujian kasus Create Classroom gagal karena kolom masukan tanggal/waktu tidak terisi .....	74
Gambar 4.43. Hasil pengujian kasus Create Classroom berhasil.....	74
Gambar 4.44. Hasil pengujian kasus Join Classroom gagal karena classroom tidak ditemukan .....	75
Gambar 4.45. Hasil pengujian kasus Join Classroom gagal user belum melakukan Face Register.....	75
Gambar 4.46. Hasil pengujian kasus Join Classroom berhasil.....	75
Gambar 4.47. Hasil pengujian kasus Face Register gagal karena descriptor wajah user dari kamera dan dari passport photo tidak cocok .....	76
Gambar 4.48. Hasil pengujian kasus Face Register berhasil.....	76
Gambar 4.49. Hasil pengujian kasus verifikasi kehadiran student oleh sistem jika role adalah teacher, maka jangan tampilkan indikator status .....	78
Gambar 4.50. Hasil pengujian kasus status kehadiran adalah on, dan belum masuk waktu mulai classroom.....	78

Gambar 4.51. Hasil pengujian kasus status kehadiran adalah away, dan belum masuk waktu mulai classroom.....	79
Gambar 4.52. Hasil pengujian kasus status kehadiran adalah on, dan sudah masuk waktu mulai classroom dan waktu classroom belum berakhir.....	79
Gambar 4.53. Hasil pengujian kasus status kehadiran adalah away, dan sudah masuk waktu mulai classroom dan waktu classroom belum berakhir.....	79
Gambar 4.54. Hasil pengujian kasus Check Classroom Attendance kasus 1.....	80
Gambar 4.55. Hasil pengujian kasus Check Classroom Attendance kasus 2.....	80
Gambar 4.56. Hasil Profiling database untuk operasi baca.....	81
Gambar 4.57. Hasil Profiling database untuk operasi tulis.....	82
Gambar 4.58. Hasil Profiling database Bandwidth byte yang diunduh.....	82
Gambar 4.59. Hasil Profiling database Bandwidth byte yang diunggah.....	83
Gambar 4.60. Penggunaan memory dan CPU ketika aplikasi diam.....	86
Gambar 4.61. Penggunaan memory dan cpu ketika aplikasi mendeteksi wajah user pertama kali.....	86
Gambar 4.62. Penggunaan memory dan cpu ketika aplikasi mendeteksi wajah user ....	87

## DAFTAR TABEL

Tabel 2.1. Spesifikasi <i>Realtime Database</i> dan <i>Cloud Firestore</i> .....	15
Tabel 4.1. Tabel identifikasi aktor.....	24
Tabel 4.2. Tabel <i>Use Case Scenario Create Classroom</i> .....	26
Tabel 4.3. Tabel <i>Use Case Scenario Check Classroom Attendance</i> .....	26
Tabel 4.4. Tabel <i>Use Case Scenario Join Classroom</i> .....	27
Tabel 4.5. Tabel <i>Use Case Scenario Login</i> .....	28
Tabel 4.6. Tabel <i>Use Case Scenario Register</i> .....	28
Tabel 4.7. Tabel <i>Use Case Scenario Face-Register</i> .....	29
Tabel 4.8. Contoh Fungsi <i>Kueri 1</i> .....	42
Tabel 4.9. Contoh Fungsi <i>Kueri 2</i> .....	43
Tabel 4.10. Kategori <i>Throughput</i> versi <i>TIPHON</i> .....	83
Tabel 4.11. Kategori <i>Latency</i> versi <i>TIPHON</i> .....	84
Tabel 4.12. Hasil Pengujian rata-rata <i>Response Time</i> di berbagai kecepatan jaringan ..	85



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pembelajaran daring adalah pembelajaran yang dilakukan secara online tanpa melakukan tatap muka secara langsung, tetapi melalui aplikasi yang telah tersedia seperti *Google Classroom*, *Google Meet*, *Edmodo* dan *Zoom* (Ermayulis, 2020). Sistem kehadiran, pemberian materi, penugasan, ujian, dan segala kegiatan lainnya yang diperlukan dalam pembelajaran daring dilakukan secara online.

Sistem kehadiran dalam pembelajaran merupakan sebuah rangkaian kegiatan untuk memperoleh dan merekap data kehadiran peserta didik yang dilakukan secara sistematis. Kehadiran peserta didik merupakan salah satu komponen yang perlu diperhatikan dan selalu dapat dipantau karena itu dapat mempengaruhi nilai peserta didik dalam kelas, dan menjadi syarat bagi mahasiswa dalam perkuliahan untuk mengikuti ujian akhir semester (Rustan, 2019).

Dalam pembelajaran daring, ada beragam cara dalam mengelola sistem kehadiran peserta didik, salah satunya adalah sistem kehadiran dengan pengisian formulir online. Sistem kehadiran berbasis formulir online adalah sistem kehadiran dimana pengajar membagikan link platform penyedia formulir online kepada peserta didik untuk membuka dan mengisinya dengan data peserta didik yang sesuai, kemudian peserta didik yang telah mengisi formulir online tersebut dinyatakan hadir. Namun sistem kehadiran menggunakan formulir online ini dapat menimbulkan berbagai kecurangan karena berbasis *shared-link*, dimana link tersebut dapat dibuka oleh siapa pun yang memilikinya dan tidak ada autentikasi maupun verifikasi yang dapat menjamin bahwa data kehadiran yang dimasukkan valid. Ini dapat dimanfaatkan oleh peserta didik walaupun tidak mengikuti pertemuan video conference, tetapi tetap dapat dinyatakan hadir. Kecurangan demikian yang mungkin terjadi mengurangi nilai validitas data dari sistem kehadiran berbasis formulir online ini.

Sistem kehadiran dengan cara manual berbeda dengan sistem kehadiran berbasis formulir online, yang status kehadiran seorang peserta didik diperoleh secara langsung dari pernyataan hadir oleh peserta didik yang bersangkutan baik itu melalui pesan dalam media sosial seperti *Whatsapp* atau *Line*, media pembelajaran online seperti *Google*

*classroom* atau *Edmodo*, maupun melalui video conference seperti *Zoom* atau *Google Meet*, sehingga dapat diverifikasi langsung kepada pengajar oleh peserta didik yang bersangkutan. Hal ini membuat data kehadiran peserta didik menggunakan sistem kehadiran dengan cara manual lebih valid daripada sistem kehadiran menggunakan formulir online. Namun, hal ini menguntungkan bagi pelajar yang mengikuti kelas online hanya untuk sekedar mendapatkan status hadir. Tidak ada jaminan bahwa seorang peserta didik mengikuti pembelajaran daring seutuhnya walaupun sudah terverifikasi hadir.

Dari permasalahan di atas, diperlukan pengembangan sistem informasi yang dapat memverifikasi kehadiran peserta didik secara otomatis dalam pembelajaran daring melalui sebuah video conference, dimana sistem dapat mengidentifikasi wajah pengguna dari kamera handphone atau laptop secara realtime, sehingga data kehadiran peserta didik yang diperoleh lebih akurat. Dengan demikian, pengajar yang mengadakan pembelajaran daring melalui video conference tidak perlu lagi memverifikasi kehadiran peserta didik satu per satu secara manual.

Teknologi yang dapat diterapkan dari mekanisme di atas adalah *Face Recognition (Pengenalan Wajah)*. *Pengenalan Wajah* adalah teknologi yang dapat mendeteksi, menemukan lokasi dengan mengabaikan latar belakang, dan mengidentifikasi wajah manusia baik melalui citra digital maupun video (Suprianto, 2013). Hasil dari identifikasi dari citra wajah menghasilkan output berupa deskripsi fitur yang merepresentasikan karakteristik dari wajah yang teridentifikasi, kemudian deskripsi fitur wajah tersebut dibandingkan dengan deskripsi fitur wajah dari database untuk pencocokan. Pencocokan dapat dilakukan dengan menghitung *kemiripan (similarity)* dari kedua deskripsi fitur wajah.

Dalam penelitian ini, penulis bermaksud untuk merancang dan membangun aplikasi pembelajaran daring video conference yang mengimplementasikan teknologi *Pengenalan Wajah* berbasis web, untuk mengembangkan sistem kehadiran melalui verifikasi wajah otomatis secara realtime guna meningkatkan validitas data kehadiran peserta didik dalam pembelajaran daring.

## 1.2. Rumusan Masalah

Dengan mengacu pada latar belakang di atas, maka rumusan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menganalisis kebutuhan dan merancang sistem deteksi kehadiran otomatis pada media pembelajaran video online.
2. Bagaimana mengimplementasikan sistem deteksi kehadiran otomatis pada media pembelajaran video online.
3. Bagaimana menguji sistem deteksi kehadiran otomatis pada media pembelajaran video online.

## 1.3. Batasan Masalah

Supaya pembahasan masalah yang dilakukan tidak menyimpang dari pokok permasalahan, maka permasalahan yang akan dibahas dibatasi sebagai berikut:

1. Aplikasi web pembelajaran video online dirancang dan dibangun dengan menggunakan webpack sebagai *module-bundler*, Firebase sebagai *Backend Service* dan *Deployment Service (Server Hosting)*, serta *Bootstrap* dan *Materialize* sebagai *CSS Framework*.
2. Aplikasi web pembelajaran video online dikembangkan menggunakan metode *Waterfall* dan menerapkan *Progressive Web App (PWA)*.
3. Sistem aplikasi mengimplementasikan *IFrame API* dari *Jitsi Meet* untuk fitur pertemuan pembelajaran *video conference*.
4. Library yang digunakan untuk model API adalah *Face-api.js* yang didasarkan pada *TensorFlow.js*, dan model yang digunakan adalah model *SSD Mobilenet v1* dan model *face-recognition* untuk pendeteksian dan pengenalan wajah.
5. Keluaran dari aplikasi web pembelajaran video online berupa data kehadiran peserta didik berisi persentase kehadiran dalam suatu pertemuan pembelajaran online.
6. Bahasa pemrograman yang digunakan dalam pembangunan aplikasi web pembelajaran video online adalah *Javascript*.
7. Pendeteksian dan pengenalan wajah serta perhitungan persentase kehadiran user dilakukan di *client-side* (sisi klien) aplikasi web pembelajaran video.
8. Aplikasi web pembelajaran video online hanya dioptimalkan untuk perangkat PC atau Laptop.



#### **1.4. Tujuan Penelitian**

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk menganalisis kebutuhan dan merancang sistem deteksi kehadiran otomatis pada media pembelajaran video online.
2. Untuk mengimplementasikan sistem deteksi kehadiran otomatis pada media pembelajaran video online.
3. Untuk menguji sistem deteksi kehadiran otomatis pada media pembelajaran video online.

#### **1.5. Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem informasi yang dapat memverifikasi kehadiran peserta pembelajaran daring secara otomatis menggunakan pendeteksian dan pengenalan wajah secara realtime melalui aplikasi Media Pembelajaran Video Online sehingga diperoleh data kehadiran yang lebih akurat.
2. Memberikan solusi kepada masyarakat khususnya yang berprofesi sebagai pengajar dan pelajar dalam melaksanakan pembelajaran daring menggunakan sistem kehadiran otomatis pada aplikasi Media Pembelajaran Video Online.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Pembelajaran Daring**

Beberapa penelitian telah menunjukkan bahwa teknologi memberikan banyak pengaruh positif terhadap pembelajaran bahasa seperti membaca menggunakan video (Gheytsi, 2015). Internet telah dipadukan sebagai alat untuk melengkapi aktivitas pembelajaran (Martins, 2015). Pembelajaran daring adalah pembelajaran yang dilakukan secara online tanpa melakukan tatap muka secara langsung, tetapi melalui aplikasi yang telah tersedia seperti *Google Classroom*, *Google Meet*, *Edmodo* dan *Zoom* (Ermayulis, 2020). Sistem kehadiran, pemberian materi, penugasan, ujian, dan segala kegiatan lainnya yang diperlukan dalam pembelajaran daring dilakukan secara online. Tujuan dari adanya pembelajaran daring ialah memberikan layanan pembelajaran bermutu dalam jaringan yang bersifat masif dan terbuka untuk menjangkau peminat ruang belajar agar lebih banyak dan lebih luas (Sofyana, 2019).

#### **2.2. Aplikasi Web dan Video Conference**

Web atau WWW (*World Wide Web*) merupakan sebuah dokumen yang berformat *HTML (Hyper Text Markup Language)* yang diakses melalui *HTTP request (Hyper Text Transfer Protocol)* lalu ditampilkan ke aplikasi browser. Halaman utama web bisa diakses melalui sebuah *Url* yang biasa disebut dengan Homepage. *Url* ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun *hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan.

Kumpulan halaman web yang saling terkait disebut Website. Website adalah sebuah kumpulan informasi atau kumpulan halaman web yang berisi informasi yang dibangun untuk tujuan tertentu yang dapat diakses melalui browser. Informasi yang dapat dimuat dalam sebuah website umumnya berisi mengenai konten gambar, ilustrasi, video, dan teks untuk berbagai macam kepentingan (Adani, 2020). Ada juga website yang merupakan sebagai media perantara pertukaran informasi antara dua pihak atau lebih.

Berdasarkan pengembangannya website terbagi menjadi tiga jenis, di antaranya website statis, website dinamis, dan website interaktif (Adani, 2020). Website Statis adalah website dimana informasi yang terkandung di dalamnya tidak bisa diperbarui

secara langsung melalui aplikasi website tersebut melainkan harus mengubah *script HTML* yang ada di dalamnya. Sedangkan Website Dinamis adalah website dimana informasi yang terkandung di dalamnya dapat di diperbarui secara langsung melalui aplikasi website tersebut. Adapun Website interaktif adalah website yang dirancang agar penggunanya dapat saling berinteraksi dan bertukar informasi. Website interaktif atau biasa disebut Web Application (disingkat Web App) mempunyai fitur tambahan yang dapat digunakan pengguna untuk berinteraksi di dalam website. Berbeda dengan website lainnya, Web App lebih responsif terhadap tindakan penggunanya. Web App dapat memberikan info yang lebih terperinci serta membantu aktivitas pengguna di dalam website. Beberapa contoh Web App adalah Trello, Slack, berbagai situs online banking, dan situs E-Commerce (NKD, 2019).

Video Conference merupakan salah satu jenis aplikasi interaktif yang mentransmisikan data berupa audio ataupun video melalui *stream*, sehingga memungkinkan penggunanya mengadakan pertemuan tatap muka jarak jauh dan saling berkomunikasi melalui jaringan komputer dan internet.

### 2.3. Jitsi Meet

*Jitsi Meet* adalah salah satu proyek dengan sumber terbuka yang menawarkan solusi Video Conference tanpa memerlukan akun. *Jitsi Meet* ini hanya didukung oleh beberapa browser saja, yaitu browser berbasis Chromium, Firefox, dan Safari. *Jitsi Meet* juga menyediakan *IFrame API* untuk developer yang memungkinkan developer untuk menanamkan fungsionalitas *Jitsi Meet* ke aplikasi mereka. Developer juga dapat mengintegrasikan penerapan *IFrame API* yang didistribusikan secara global dan sangat tersedia di platform *meet.jit.si* itu sendiri.

### 2.4. Face-api.js

TensorFlow.js adalah library untuk melatih, mengembangkan, dan menggunakan model Deep Learning menggunakan bahasa pemrograman Javascript. Saat berjalan di dalam browser, TensorFlow.js menggunakan GPU perangkat melalui WebGL. Di Node.js, TensorFlow.js mengikat ke library TensorFlow C yang memungkinkan akses penuh ke TensorFlow. TensorFlow.js juga menyediakan implementasi CPU yang lebih lambat sebagai fallback, yang diimplementasikan dalam JS biasa. Fallback ini dapat berjalan di lingkungan eksekusi apa pun dan secara otomatis digunakan saat lingkungan tidak memiliki akses ke WebGL atau biner TensorFlow.

Face-api.js adalah modul Javascript, dibangun di atas core tensorflow.js, dan menerapkan beberapa *Convolutional Neural Network* (CNN) untuk melakukan *Pendeteksian Wajah* dan *Pengenalan Wajah*, dan telah dioptimalkan untuk bekerja di web dan perangkat seluler.

Berikut adalah contoh penggunaan face-api.js untuk mendeteksi wajah dari elemen HTML, bisa berupa *HTML img*, *canvas*, maupun *video* element.

```
const detections = await faceapi.detectAllFaces(input)
```

Berbeda dengan fungsi *detectAllFaces*, fungsi *detectSingleFace* hanya mendeteksi satu wajah dalam sebuah gambar dengan nilai *confidence score* tertinggi seperti yang ditunjukkan dalam baris kode berikut.

```
const detection = await faceapi.detectSingleFace(input)
```

## 2.5. Face Recognition

*Face Recognition* (*Pengenalan wajah*) adalah masalah identifikasi dan verifikasi wajah orang dalam citra. Ini adalah tugas yang biasa dilakukan oleh manusia, bahkan di bawah cahaya yang berbeda-beda, dan ketika wajah diubah oleh usia atau dihalangi dengan aksesoris ataupun rambut. Namun demikian, ini tetap menjadi masalah yang menantang bagi *Computer Vision* selama beberapa dekade hingga saat ini (Brownlee, 2019).

Machine Learning dikenal sebagai bagian dari kecerdasan buatan. Algoritma Machine Learning membangun model matematika berdasarkan data sampel, yang dikenal sebagai data training, untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit (Pratama, 2020). Deep Learning merupakan bagian dari Machine Learning yang terdiri dari banyak lapisan (*hidden layer*) dan membentuk tumpukan, lapisan tersebut adalah sebuah algoritma atau metode yang melakukan klasifikasi perintah yang di-input hingga menghasilkan output. Perkembangan bidang Deep Learning saat ini telah dipermudah oleh banyaknya library dan Application Program Interface (API) (Nurfita, 2018).

*Convolutional Neural Network* (CNN) bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah citra. CNN adalah sebuah teknik yang terinspirasi dari cara mamalia atau manusia, menghasilkan persepsi visual. CNN memanfaatkan proses

konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah citra, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian citra tersebut dengan filter yang digunakan (Lina, 2019).

*Pengenalan Wajah* adalah teknologi yang dapat mendeteksi, menemukan lokasi dengan mengabaikan latar belakang, dan mengidentifikasi wajah manusia baik melalui citra digital maupun video. Prinsip dasar *Pengenalan Wajah* adalah mengutip informasi unik wajah, kemudian dilakukan *encode* dan dibandingkan dengan hasil *decode* yang sebelumnya dilakukan. Pada metode *Eigen Face*, *decoding* dilakukan dengan menghitung vektor eigen kemudian direpresentasikan dalam sebuah matriks. Vektor eigen juga dinyatakan sebagai karakteristik wajah oleh karena itu metode ini disebut dengan *Eigen Face*. Setiap wajah direpresentasikan dalam kombinasi linear eigenface. Metode *Eigen Face* adalah bagaimana cara menguraikan informasi yang relevan dari sebuah citra wajah, kemudian mengubahnya ke dalam satu set kode yang paling efisien dan membandingkan kode wajah tersebut dengan database berisi beragam wajah yang telah dikodekan secara serupa (Suprianto, 2013).

Teknologi *Pengenalan Wajah* terus berkembang dan semakin canggih. Namun, ada beberapa faktor yang dapat mengurangi tingkat akurasi *Pengenalan Wajah* menurut Arfienda (2018), di antaranya:

- Foto dengan resolusi rendah dan minim pencahayaan
- Sudut foto yang berbeda
- Ekspresi wajah, bahkan senyum kecil saja bisa menjadi tantangan tersendiri bagi sistem pencocokan wajah
- Mata memiliki peranan kunci dalam menentukan akurasi. Maka, benda seperti kacamata hitam yang menghalangi mata dapat mengganggu kerja software
- Benda lainnya yang menutupi bentuk asli wajah seperti topi, syal, atau model rambut yang menghalangi wajah, terutama area mata
- Dalam kasus tertentu, jambang dan riasan dapat mengganggu program deteksi wajah



### 2.5.1. SSD Mobilenet v1 Model

Salah satu pendeteksi wajah utama yang akan digunakan sebagai model adalah SSD (Single Shot Multibox Detector) yang terdiri dari CNN berbasis *MobileNet v1*. Namun, model ini menyertakan lapisan prediksi kotak tambahan yang ditumpuk di atas jaringan. Jaringan saraf dapat menghitung lokasi setiap wajah dalam sebuah gambar dan kemudian akan mengembalikan kotak pembatas dan probabilitasnya untuk setiap wajah yang ditemukan. Detektor ini difokuskan untuk memperoleh akurasi tinggi dalam mendeteksi kotak pembatas wajah, bukan waktu inferensi yang rendah. *SSD Mobilenet* adalah model deteksi wajah yang telah dilatih sebelumnya pada dataset WIDERFACE yang tersedia untuk umum.

Model *SSD Mobilenet v1* dapat dimuat dan diterapkan melalui library *face-api.js* seperti yang ditunjukkan dalam baris kode di bawah ini dengan asumsi “/models” adalah lokasi direktori model tersimpan.

```
await faceapi.loadSsdMobilenetv1Model('/models')
```

Baris kode berikut ekuivalen dengan fungsi *loadSsdMobilenetv1Model*.

```
await faceapi.nets.ssdMobilenetv1.loadFromUri('/models')
```

### 2.5.2. Face-Recognition Model

Model *Face-Recognition* didasarkan pada library *face-recognition* dari *Python Package Index (PyPI)*. Untuk *Pengenalan Wajah*, arsitektur seperti *ResNet-34* diimplementasikan untuk menghitung *descriptor* wajah (vektor fitur dengan dengan panjang 128) dari gambar wajah yang diberikan, yang digunakan untuk mendeskripsikan karakteristik wajah seseorang. Model ini tidak hanya terbatas pada dataset citra wajah pada saat *training*, tapi dapat digunakan untuk pengenalan citra wajah baru dengan menghitung *descriptor* dari citra tersebut, lalu *descriptor* tersebut dibandingkan dengan *descriptor* dari citra wajah dari database untuk pencocokan. Lebih lanjut lagi, pencocokan dapat dilakukan dengan menghitung jarak Euclid dari kedua *descriptor* wajah lalu algoritma akan menentukan apakah kedua wajah serupa berdasarkan nilai *Threshold Distance*.

Untuk menghitung *descriptor* sebuah citra wajah menggunakan *face-api.js*, fungsi *withDescriptors* dipanggil setelah mendeteksi wajah menggunakan fungsi

*detectAllFaces* atau *detectSingleFace*. Berikut adalah contoh penggunaan fungsi *withDescriptors* untuk menghitung *descriptor* dari elemen gambar html yang digabungkan dengan pendeteksian semua wajah dalam citra.

```
await faceapi.detectAllFaces(input)
  .withFaceLandmarks()
  .withFaceDescriptors()
```

Sedangkan perhitungan *descriptor* dengan pendeteksian wajah tunggal ditunjukkan dalam contoh baris kode berikut.

```
await faceapi.detectSingleFace(input)
  .withFaceLandmarks()
  .withFaceDescriptor()
```

Untuk melakukan *Pengenalan Wajah*, kelas *FaceMatcher* dari *face-api.js* dapat digunakan untuk membandingkan *descriptor* citra wajah referensi dengan *descriptor* wajah yang lain yang ingin dikenali.

Contoh inisialisasi *FaceMatcher* dari citra wajah referensi ditunjukkan dalam baris kode berikut.

```
const faceMatcher = new faceapi.FaceMatcher(results)
```

Selanjutnya pengenalan wajah dapat dilakukan dengan menggunakan fungsi *findBestMatch* yang dimana parameternya adalah *descriptor* dari citra wajah yang ingin dikenali.

```
results.forEach(fd => {
  const bestMatch = faceMatcher.findBestMatch(fd.descriptor)
  console.log(bestMatch.toString())
})
```

Dalam *face-api.js* juga terdapat kelas *LabeledFaceDescriptors* untuk membuat *descriptor* dengan label yang dapat ditentukan. Kelas ini dapat digunakan dan menjadi parameter dalam konstruktor *FaceMatcher* saat inisialisasi.

```

const labeledDescriptors = [
  new faceapi.LabeledFaceDescriptors(
    'Belki',
    [descriptorBelki1, descriptorBelki2]
  ),
  new faceapi.LabeledFaceDescriptors(
    'Larry',
    [descriptorLarry]
  )
]
const faceMatcher = new faceapi.FaceMatcher(labeledDescriptors)

```

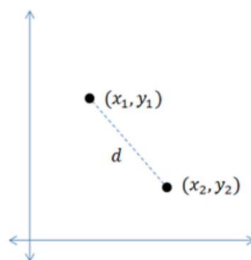
## 2.6. Euclidean Distance

*Euclidean Distance* atau Jarak Euclid adalah perhitungan jarak dari 2 buah titik dalam *Euclidean Space*. *Euclidean Space* diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Jarak Euclid biasa diterapkan untuk membantu proses klasifikasi pada *Data Mining*. Contoh algoritma yang menggunakan jarak ini adalah *K-Nearest Neighbor*. Jarak Euclid juga bisa diterapkan untuk algoritma image retrieval (temu kembali citra) (Fauzan, 2019). Dengan jarak ini, ruang Euclid menjadi ruang metrik. Secara umum, rumus Jarak Euclid ditunjukkan oleh persamaan 2.1.

$$d = \sqrt{\sum_{i=0}^n (p_i - q_i)^2} \quad (2.1)$$

Jarak Euclid untuk ruang dua dimensi didefinisikan pada persamaan 2.2. dan diilustrasikan pada Gambar 2.1.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.2)$$



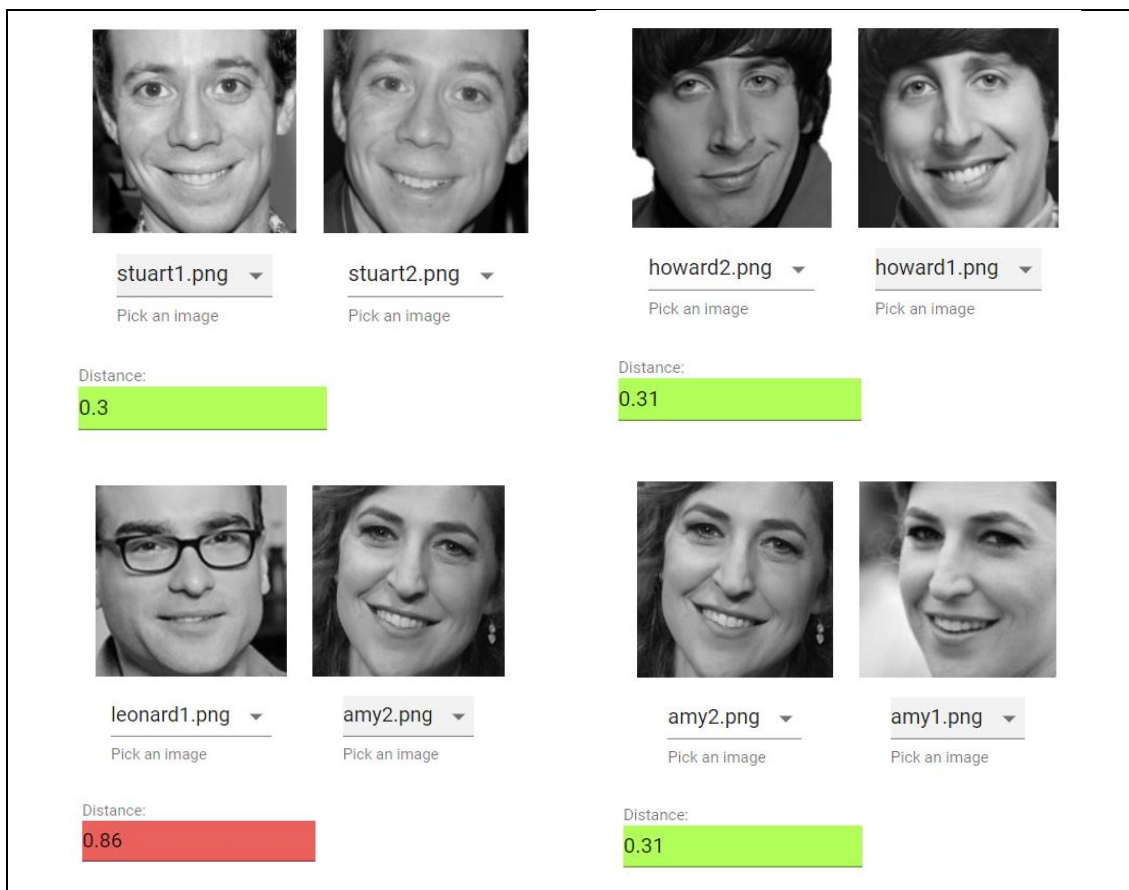
Gambar 2.1. Jarak Euclid di  $\mathbb{R}^2$

Jarak Euclid tidak mempertimbangkan hubungan masing-masing dimensi vektor, dan masing-masing dimensi sama pentingnya, yang mempengaruhi keefektifan dan ruang lingkup penggunaannya. Jarak Euclid tertimbang digunakan untuk memastikan keakuratan identifikasi wajah (Sandoval, 2019).

Berikut adalah contoh penggunaan fungsi untuk menghitung Jarak Euclid dari dua vektor menggunakan face-api.js.

```
const dist = faceapi.euclideanDistance([0, 0], [0, 10])
console.log(dist) // menghasilkan 10
```

Contoh perbandingan dua citra wajah menggunakan Jarak Euclid untuk menghitung kemiripan diilustrasikan dalam Gambar 2.2.



Gambar 2.2. Contoh penggunaan Jarak Euclid untuk menghitung kemiripan dua citra wajah

Sumber: [justadudewhohacks.github.io/face-api.js](https://justadudewhohacks.github.io/face-api.js)

## 2.7. Firebase

Firebase merupakan platform untuk aplikasi realtime. Ketika data berubah, maka aplikasi yang terhubung dengan firebase akan memperbarui secara langsung melalui setiap device (perangkat) baik website ataupun mobile (Sanadi, 2018). Firebase dikenal sebagai platform aplikasi web. Firebase membantu pengembang membangun aplikasi berkualitas tinggi. Firebase menyimpan data dalam format *Javascript Object Notation* (JSON) yang tidak menggunakan query untuk memasukkan, memperbarui, menghapus atau menambahkan data ke dalam database (Khawas, 2018). Firebase menyediakan berbagai jenis layanan, tiga di antaranya yang digunakan dalam penelitian ini adalah Authentication, *Realtime Database*, dan *Cloud Firestore* (Firebase Team).

### 2.7.1. Firebase Authentication

Firebase Authentication mendukung penyedia login sosial seperti Facebook, Google GitHub, dan Twitter. Ini adalah layanan yang dapat mengautentikasi pengguna hanya dengan menggunakan kode *client-side* dan merupakan layanan berbayar. Ini juga mencakup sistem manajemen pengguna di mana pengembang dapat mengaktifkan autentikasi pengguna dengan email dan login kata sandi yang disimpan dengan Firebase. Firebase Authentication terintegrasi erat dengan layanan Firebase lainnya dan sistem ini memanfaatkan berbagai standar industri, seperti OAuth 2.0 dan OpenID Connect, sehingga dapat dengan mudah diintegrasikan dengan kustom backend.

### 2.7.2. *Realtime Database*

*Realtime Database* adalah database NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database relasional. Firebase *Realtime Database* adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika membangun aplikasi lintas platform dengan SDK iOS, Android, dan Javascript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima pembaruan data terbaru secara otomatis.

Struktur data *Realtime Database* ditunjukkan pada Gambar 2.3.

```

{
  "collection_1" : {
    "id_1" : {
      "attr_1" : "value 1",
      "attr_2" : 2,
      "attr_3" : true
    }
  }
}

```

Gambar 2.3. Struktur data Realtime Database

*Realtime Database* menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan Keamanan Firebase *Realtime Database*, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firebase Authentication, developer dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya.

### 2.7.3. *Cloud Firestore*

*Cloud Firestore* adalah database terbaru dari Firebase untuk pengembangan aplikasi. Database ini melanjutkan keberhasilan *Realtime Database* dengan model data baru yang lebih intuitif. *Cloud Firestore* adalah database NoSQL berorientasi dokumen. Tidak seperti database SQL, database tersebut tidak memiliki tabel atau baris. Sebagai gantinya, data disimpan dalam bentuk dokumen, yang disusun menjadi koleksi, seperti contoh dokumen yang tersimpan yang ditunjukkan pada Gambar 2.4.

```

users
├── alovelace
│   ├── first : "Ada"
│   ├── last  : "Lovelace"
│   └── born  : 1815
└── aturing
    ├── first : "Alan"
    ├── last  : "Turing"
    └── born  : 1912

```

Gambar 2.4. Struktur data Firestore

Sumber: [firebase.google.com](https://firebase.google.com)

*Cloud Firestore* memiliki fitur kueri yang lebih lengkap dan lebih cepat, serta penskalaan yang lebih mendalam dibandingkan dengan *Realtime Database*. Walaupun *Realtime Database* dan *Cloud Firestore* merupakan database yang sama-sama



menawarkan sinkronisasi realtime, namun kedua database dari Firebase ini mempunyai perbedaan yang signifikan. *Realtime Database* memiliki struktur data yang lebih fleksibel dan memiliki latensi yang lebih rendah. Jika suatu data memiliki struktur yang bertingkat seperti *nested array*, dan memungkinkan sering terjadinya operasi tulis daripada operasi baca, maka *Realtime Database* adalah database yang sesuai untuk data tersebut. Sedangkan *Cloud Firestore* memiliki struktur data yang lebih sederhana tetapi memiliki fitur query yang lebih lengkap, sehingga jenis data yang sesuai untuk jenis database ini adalah jenis data dengan struktur data yang sederhana dan memungkinkan lebih sering terjadi operasi baca dibandingkan operasi tulis.

Spesifikasi dasar dari *Realtime Database* dan *Cloud Firestore* ditunjukkan pada Tabel 2.1 berikut.

Tabel 2.1. Spesifikasi *Realtime Database* dan *Cloud Firestore*

Jenis Database	Penyimpanan	Koneksi Bersamaan
<b>Realtime Database</b>	1 GB (Spark), \$5/GB (Blaze)	100 (Spark), 200.000/Database (Blaze)
<b>Cloud Firestore</b>	1 GiB (Spark), \$0,1/GiB	1.000.000 (Blaze & Spark)

## 2.8. Penelitian Terkait

Penelitian berjudul “*A Web-Based Mobile Attendance System with Facial Recognition Feature*” yang dilakukan oleh Noradila Nordin dan Nurul Husna Mohd Fauzi pada tahun 2020 bertujuan untuk merancang sistem absensi seluler berbasis web dengan fitur pengenalan wajah. Ini menggabungkan perangkat seluler yang ada dengan kamera dan sistem pengenalan wajah untuk memungkinkan sistem kehadiran digunakan di ruang kelas secara otomatis dan efisien dengan persyaratan implementasi kecil. Prototipe sistem mendapat respon positif dari para relawan yang menguji sistem untuk menggantikan penandaan absensi konvensional (Nordin, 2020).

Penelitian berjudul “*Metode Face Recognition untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang*” yang dilakukan oleh Luthfi Maslichul Kurniawan pada November 2014 bertujuan untuk membangun sebuah sistem *face recognition* yang dirancang-bangun menggunakan bahasa pemrograman Python dan pustaka *OpenCV* dimana sistem ini mampu mengolah

data-data foto yang ditangkap dari proses presensi online untuk dianalisis apakah ada wajah manusia yang terdeteksi. Hasil dari rancang bangun ini adalah sistem *face recognition* yang mampu berjalan secara otomatis di komputer server untuk membaca basis data presensi, mengolah foto-foto yang tersimpan pada basis data tersebut, mendeteksi wajah pada foto-foto yang diolah kemudian menampilkan hasilnya pada tabel basis data presensi untuk diolah dalam bentuk skor deteksi wajah yang tampil di rekapitulasi presensi online pegawai (Kurniawan, 2014).

Penelitian berjudul “*Real Time Face Recognition using Eigenface and Viola-Jones Face Detector*” yang dilakukan oleh Jacky Efendi, Muhammad Zul, dan Wawan Yunanto pada tahun 2017 bertujuan untuk mengimplementasikan pengenalan wajah berbasis *eigenface* secara realtime dalam sebuah aplikasi untuk melakukan absensi. Gambar wajah input ditangkap menggunakan webcam. Aplikasi dibangun menggunakan bahasa pemrograman C#, memanfaatkan library *EmguCV*. Aplikasi dikembangkan dengan menggunakan Visual Studio 2015. Deteksi wajah dilakukan dengan algoritma Viola-Jones. Metode *eigenface* digunakan untuk melakukan pengenalan wajah pada citra wajah yang dideteksi. Dalam penelitian ini dilakukan total 8 pengujian dalam kondisi yang berbeda. Dari pengujian didapatkan bahwa aplikasi ini mampu mengenali citra wajah dengan akurasi sebesar 90% dan terendah 6,67%. Solusi ini dapat digunakan sebagai alternatif pengambilan absensi secara realtime di lingkungan dengan intensitas cahaya 170 lux, resolusi webcam 320 x 240 piksel, dan subjek berdiri sejauh 1 meter saat tidak menggunakan kacamata. Waktu pengenalan rata-rata adalah 0,18125 ms (Efendi, 2017).

Penelitian berjudul “*Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, Eigenface PCA & MySQL*” yang dilakukan oleh Dodit Suprianto, Rini Nur Hasanah, dan Purnomo Budi Santosa pada Desember 2013 bertujuan untuk menerapkan pengenalan wajah pada Aplikasi sistem pengenalan wajah secara realtime pada sistem pengawasan, identifikasi dan keamanan melalui tahap *face detection, feature extraction* dan *face recognition*, selanjutnya dicocokkan dengan data profil yang tersimpan di dalam database. Pendeteksian wajah menggunakan metode Adaboost, pengenalan wajah menggunakan metode Eigenface PCA dan database MySQL untuk menyimpan informasi profile. Penggunaan metode tersebut untuk pengenalan wajah pada kondisi realtime dengan perbedaan jarak antara sensor dan wajah, posisi wajah, intensitas cahaya yang

mengenai wajah, mimik muka dan atribut wajah dalam penelitian ini memberikan tingkat keberhasilan sebesar 80% dalam mengidentifikasi wajah (Suprianto, 2013).

Penelitian berjudul “*Rancang Bangun Sistem Presensi Online Berbasis Granted Validitas Data*” yang dilakukan oleh Nurilla Lailatul Khoiriyah, Fitri Marisa, dan Indra Dharma Wijaya pada Maret 2018 bertujuan untuk merancang-bangun sistem presensi online yang dapat mevalidasi kehadiran mahasiswa secara online, sehingga mempermudah sistem rekapitulasi dan data yang di peroleh lebih akurat menggunakan metode *Granted Validity*. Penelitian ini menghasilkan aplikasi sistem presensi online berbasis *Granted Validitas* yang memudahkan jurusan dalam rekapan presensi, memudahkan mahasiswa dan dosen maupun ketua program studi dalam mengecek dan mencetak rekapan presensi, menghasilkan presensi ujian sesuai ketentuan, mampu memberikan prasyarat untuk mahasiswa yang kehadirannya kurang dari 80%, dan hasil uji coba oleh pengguna (ketua program studi, dosen & mahasiswa) menghasilkan 32% untuk sistem lama dan 68% untuk sistem baru yang membuktikan bahwa sistem baru lebih baik dari pada sistem lama (Khoiriyah, 2018).

Penelitian berjudul “*Penerapan Absensi Online Berbasis Android pada Peningkatan Kedisiplinan Dan Kinerja Guru Pegawai Negeri Sipil Pada Bidang PAI*” yang dilakukan oleh Yani Mulyani pada Juni 2020 bertujuan untuk (1) Menjelaskan bagaimana penerapan absensi online berbasis android di Gugus 3 Ahmad Yani Kabupaten; (2) Menggambarkan bagaimana kedisiplinan dan kinerja guru PNS pada bidang PAI di Gugus 3 Ahmad Yani Kecamatan Sindangwangi Kabupaten Majalengka; (3) Mendeskripsikan penerapan absensi online berbasis android dalam peningkatan kedisiplinan dan kinerja guru PNS bidang PAI di Gugus 3 Ahmad Yani Kecamatan Sindangwangi Kabupate Majalengka. Penelitian ini menggunakan pendekatan kualitatif dengan menggunakan model penelitian studi kasus fenomenologi. Kesimpulan dari penelitian ini adalah absensi online berbasis android merupakan daftar kehadiran dengan menggunakan ponsel pintar android yang berbasis linux sebagai bagian dari kemajuan teknologi yang mempermudah sekaligus mendisiplinkan guru Pegawai Negeri Sipil dalam bidang PAI. Walaupun masih menjadi pro kontra karena aplikasi yang belum sempurna, Sumber Daya Manusia yang belum siap dengan kemajuan teknologi, masalah ekonomi dan pribadi pegawai negeri sipil lainnya (Mulyani, 2020).

## BAB III

### METODE PENELITIAN

#### 3.1. Metode Pengumpulan Data

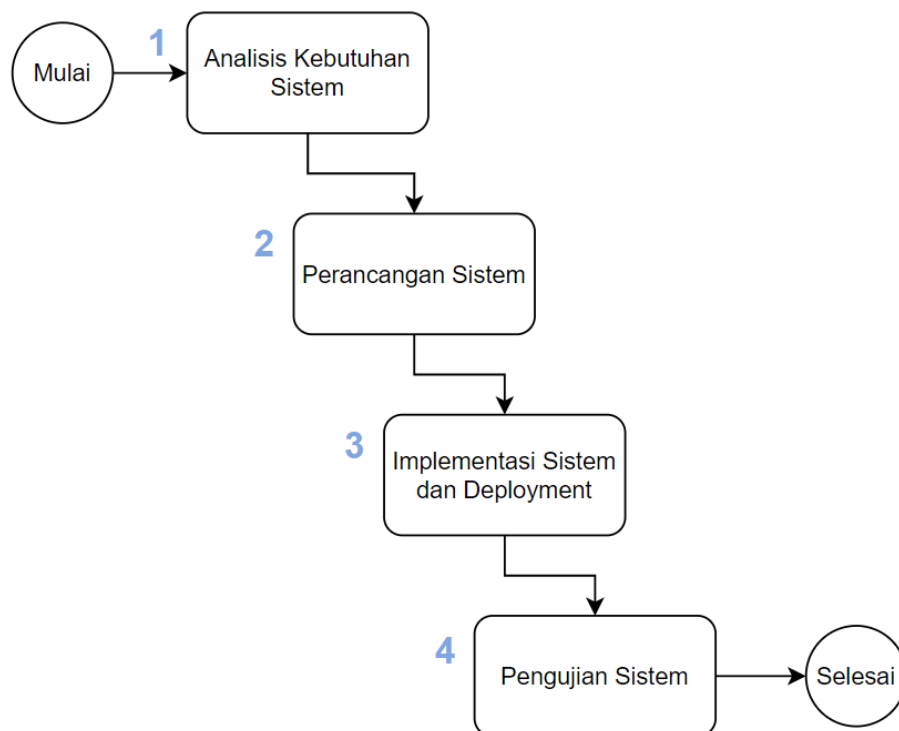
Pengumpulan data dimaksudkan untuk mencari dan mengumpulkan data yang terkait dengan penelitian seperti landasan teori, metode penelitian, metode proses, dan acuan penelitian sejenis. Metode yang digunakan dalam pengumpulan data adalah studi pustaka.

##### 3.1.1. Studi Pustaka

Peneliti menggunakan studi pustaka untuk mengumpulkan teori, fakta, kasus, dan masalah yang berkaitan dengan penelitian ini. Studi pustaka dilakukan dari sumber seperti jurnal, buku, dan *website*.

#### 3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode *Waterfall*. Metode ini terdiri dari empat tahap, antara lain analisis kebutuhan sistem, perancangan sistem, implementasi sistem, dan pengujian sistem, seperti yang terlihat pada Gambar 3.1 berikut.



Gambar 3.1. Tahap-tahap pengembangan sistem Metode *Waterfall*