

SKRIPSI

SISTEM PRESENSI MAHASISWA DENGAN *FACE RECOGNITION* DAN *LOCATION BASED SERVICE* BERBASIS *WEB* MENGGUNAKAN ALGORITMA *EUCLIDEAN DISTANCE*

**Disusun Oleh :
Muhammad Taofik
065120006**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2024**

SKRIPSI

SISTEM PRESENSI MAHASISWA DENGAN *FACE RECOGNITION* DAN *LOCATION BASED SERVICE* BERBASIS *WEB* MENGGUNAKAN ALGORITMA *EUCLIDEAN DISTANCE*

Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana Komputer Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

Disusun Oleh :
Muhammad Taofik
065120006



PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN

BOGOR

2024

HALAMAN PENGESAHAN

Judul : Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*

Nama : Muhammad Taofik

NPM : 065120006

Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA – UNPAK



Adriana Sari Aryani, S.Kom., M.Cs.

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA - UNPAK



Arie Qur'ania, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Ilmu Komputer
FMIPA – UNPAK



Arie Qur'ania, S.Kom., M.Kom.

Dekan
FMIPA - UNPAK



Asep Denih, S.Kom., M.Sc., Ph.D.

PERNYATAAN KEASLIAN KARYA TULIS

Dengan ini penulis menyatakan bahwa :

Sejauh yang penulis ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian dimana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini penulis buat dengan sebenar - benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, November 2024



Muhammad Taofik
065120006

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Saya yang bertandatangan di bawah ini :

Nama : Muhammad Taofik
NPM : 065120006
Judul Skripsi : Sistem Presensi Mahasiswa Dengan *Face Recognition*
Dan *Location Based Service* Berbasis *Web*
Menggunakan Algoritma *Euclidean Distance*.

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, November 2024



Muhammad Taofik
065120006

RIWAYAT HIDUP

Penulis dilahirkan di Kota Bogor pada tanggal 02 Mei 2001. Penulis merupakan anak dari pasangan Bapak Atang dan Ibu Neneng Listiawati sebagai anak bungsu dari dua bersaudara. Penulis memulai Pendidikan di Sekolah Dasar yang bertempat di SDN Kertamaya, kemudian lanjut masuk ke SMPN 17 Kota Bogor dan Penulis adalah Alumni dari SMKN 4 Kota Bogor.

Ketika lulus SMK pada tahun 2020 penulis meneruskan Pendidikan ke Universitas Pakuan Bogor, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam.

Semester 8 berjalan penulis menyelesaikan penelitian berjudul Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*.

RINGKASAN

Teknologi mengalami perkembangan yang begitu pesat, perkembangan tersebut berdampak besar pada hampir semua bidang, termasuk bidang pendidikan. Pendidikan sendiri merupakan kegiatan yang esensial bagi manusia yang dapat menentukan arah baik dalam hal hukum, sosial, kebudayaan dan lain-lain. Presensi sendiri bertujuan untuk mengetahui tingkat kehadiran serta kedisiplinan seseorang. Laporan presensi menjadi sangat penting pada sebuah institusi terutama pada perkuliahan karena presensi sendiri dijadikan sebagai salah satu penilaian penting terhadap kedisiplinan seseorang mahasiswa sehingga ketepatan waktu dalam mengikuti kegiatan perkuliahan atau presensi harus dicatat dengan baik. Terkhusus pada Laboratorium Ilmu Komputer FMIPA Universitas Pakuan. Sistem presensi untuk mata kuliah luring atau *offline* masih menggunakan cara manual yaitu menggunakan buku presensi. Untuk mengatasi masalah ini, penelitian ini mengimplementasikan teknologi *face recognition* yang merupakan salah satu teknik biometrik yang dapat melakukan identifikasi wajah dan identitas seseorang juga merupakan salah satu program *artificial intelligence*, tepatnya *deep learning*, yang sejak awal kemunculannya masih diminati dan terus dikembangkan hingga saat ini dan digabungkan dengan *Location Based Service* atau LBS yaitu layanan yang berfungsi untuk mencari atau memberitahu posisi dimana perangkat berada dengan menggunakan teknologi *Global Positioning Service* atau GPS.

Penelitian ini merujuk pada beberapa studi sebelumnya mengenai implementasi *face recognition* dan *Location Based Service* di berbagai institusi. Hasil penelitian ini adalah pengembangan sistem presensi mahasiswa berbasis *web*, yang dilengkapi dengan teknologi *face recognition* dan *Location Based Service*. Judul penelitian ini adalah " Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*."

Kata Kunci : *Face Recognition*, *Location Based Service*, *Euclidean Distance*, Sistem Presensi Mahasiswa, Laboratorium Ilmu Komputer FMIPA Universitas Pakuan.

KATA PENGANTAR

Puji syukur selalu dipanjatkan atas kehadiran Allah SWT yang telah memberikan rahmat dan petunjuk, sehingga skripsi yang berjudul “**Sistem Presensi Mahasiswa Dengan *Face Recognition Dan Location Based Service* Berbasis Web Menggunakan Algoritma *Euclidean Distance***” dapat terselesaikan. Penulisan tugas akhir ini merupakan salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Ilmu Komputer FMIPA UNPAK Bogor.

Dalam penulisan tugas akhir ini, penulis dengan senang hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Arie Qur'ania, S.Kom., M.Kom., selaku Pembimbing Utama yang telah memberikan dorongan moril dan motivasi.
2. Adriana Sari Aryani, S.Kom., M.Cs., selaku Pembimbing Pendamping yang telah memberikan bimbingan semangat dan motivasi.
3. Arie Qur'ania, S.Kom., M.Kom., selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Pakuan Bogor.
4. Kedua Orang Tua yang sealalu mendo'akan, memberikan semangat dan selalu menjadi motivasi.

Saran dan kritik yang membangun dalam penulisan tugas akhir ini akan diterima dengan senang hati. Mudah-mudahan Allah SWT akan membalas semua kebaikan kepada semua pihak yang membantu. Akhir kata, semoga laporan ini dapat bermanfaat bagi kita semua. Aamiin

Bogor, November 2024



Muhammad Taofik
065120006

DAFTAR ISI

HALAMAN PENGESAHAN	i
PERNYATAAN KEASLIAN KARYA TULIS	ii
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	iii
RIWAYAT HIDUP	iv
RINGKASAN	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Ruang Lingkup.....	2
1.4 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA	4
2.1 Landasan Teori.....	4
2.1.1 Sistem Presensi.....	4
2.1.2 Mahasiswa.....	4
2.1.3 <i>Face Recognition</i>	4
2.1.4 <i>Location Based Service</i>	4
2.1.5 <i>Website</i>	5
2.1.6 <i>Roboflow</i>	5
2.1.7 <i>Convolutional Neural Network</i>	5
2.1.8 <i>Tensorflow</i>	5
2.1.9 <i>Euclidean Distance</i>	6
2.1.10 <i>Face-api.js</i>	6
2.1.11 Javascript.....	8
2.1.12 Python.....	8
2.1.13 PHP.....	8
2.1.14 Bootstrap.....	9
2.1.15 MySQL.....	9
2.2 Penelitian Terdahulu.....	10
2.3 Tabel Perbandingan Penelitian.....	12
BAB III METODE PENELITIAN	13
3.1 Metodologi Penelitian.....	13
3.1.1 <i>Location Based Service (LBS)</i>	13
3.1.2 <i>Face Recognition</i>	13
3.1.3 <i>Systems Development Life Cycle (SDLC)</i>	16
3.2 Tempat Penelitian.....	17
3.3 Alat dan Bahan.....	17
3.3.1 Alat Penelitian.....	17
3.3.2 Bahan Penelitian.....	17
BAB IV RANCANGAN DAN IMPLEMENTASI	18
4.1 <i>Location Based Service (LBS)</i>	18
4.1.1 Penyimpanan Koordinat Kampus.....	18

4.1.2	Akusisi Koordinat Perangkat	18
4.1.3	Perhitungan Jarak	18
4.2	<i>Face Recognition</i>	19
4.2.1	Akusisi Data	19
4.2.2	Deteksi Wajah	20
4.2.3	<i>Feature Extraction</i>	20
4.2.4	<i>Face Descriptor</i>	21
4.2.5	Pencocokan Wajah	22
4.3	<i>Systems Development Life Cycle (SDLC)</i>	23
4.3.1	Tahap Perencanaan	23
4.3.2	Tahap Analisis Kebutuhan	24
4.3.3	Tahap Perancangan	24
4.3.4	Tahap Implementasi	27
BAB V HASIL DAN PEMBAHASAN		29
5.1	Hasil	29
5.1.1	<i>Location Based Service</i>	29
5.1.2	<i>Face Recognition</i>	29
5.1.3	Pelatihan Model	31
5.2	Uji Coba	31
5.2.1	Prosedur Pengujian	31
5.2.2	Hasil Pengujian	32
5.2.3	Evaluasi Pengujian	34
BAB VI KESIMPULAN DAN SARAN		35
6.1	Kesimpulan	35
6.2	Saran	35
DAFTAR PUSTAKA		36
LAMPIRAN		38

DAFTAR GAMBAR

Gambar 1. Arsitektur CNN.....	5
Gambar 2. Arsitektur Model <i>SSD MobileNet V1</i>	7
Gambar 3. face-api.js - <i>ssd_mobilenetv1_model</i>	7
Gambar 4. face-api.js - <i>face_landmark_68_model</i>	7
Gambar 5. Arsitektur Model ResNet 34.....	8
Gambar 6. Alur LBS.....	13
Gambar 7. Alur Face Recognition.....	14
Gambar 8. Alur Transformasi Gambar	14
Gambar 9. Tahap system development life cycle (SDLC)	16
Gambar 10. Akusisi Data Wajah.....	19
Gambar 11. Deteksi Wajah	20
Gambar 12. Face Landmark	20
Gambar 13. Alur Pembuatan Model.....	24
Gambar 14. Alur Pembuatan Dataset	25
Gambar 15. Kumpulan Data.....	25
Gambar 16. Anotasi Data	25
Gambar 17. Pemabagian Data	26
Gambar 18. Augmentasi Data	26
Gambar 19. Alur Pelatihan Model.....	27
Gambar 20. Konfigurasi Pelatihan	28
Gambar 21. Kode Validasi.....	28
Gambar 22. Proses Pelatihan Model.....	28
Gambar 23. Hasil Implementasi LBS.....	29
Gambar 24. Mahasiswa Tidak Terdaftar.....	29
Gambar 25. Mahasiswa terdaftar.....	30
Gambar 26. Presensi Berhasil.....	30
Gambar 27. Tidak Memiliki Jadwal	30
Gambar 28. Sudah Melakukan Presensi	31
Gambar 29. Hasil Pelatihan Model.....	31
Gambar 30. Gambar Referensi (Mawar)	32
Gambar 31. Gambar Validasi (Melati).....	32
Gambar 32. Deteksi Wajah Referensi.....	33
Gambar 33. Landmark Wajah Referensi.....	33
Gambar 34. Descriptor Wajah Referensi	33
Gambar 35. Hasil Pengenalan Wajah	34
Gambar 36. Use Case Diagram	56
Gambar 37. Activity Register.....	57
Gambar 38. <i>Activity</i> Presensi.....	58
Gambar 39. <i>Activity Login</i>	59
Gambar 40. <i>Activity</i> Jadwal Asprak.....	60
Gambar 41. <i>Activity</i> Laporan Asprak.....	60
Gambar 42. <i>Activity</i> Mengelola Mahasiswa	61
Gambar 43. <i>Activity</i> Mengelola Asprak.....	62
Gambar 44. <i>Activity</i> Mengelola Kelas	63
Gambar 45. <i>Activity</i> Mengelola Matkul	64
Gambar 46. <i>Activity</i> Mengelola Jadwal.....	65

Gambar 47.	Activity Laporan Admin	66
Gambar 48.	Activity Mengelola Profil Kampus.....	66
Gambar 49.	ERD	67
Gambar 50.	Halaman Utama	71
Gambar 51.	<i>Mockup</i> Halaman <i>Register</i>	71
Gambar 52.	<i>Mockup</i> Input Wajah	72
Gambar 53.	<i>Mockup</i> Uji Coba Rekognisi.....	72
Gambar 54.	<i>Mockup</i> Halaman <i>Login</i>	73
Gambar 55.	<i>Mockup</i> Halaman <i>Dashboard</i>	73
Gambar 56.	<i>Mockup</i> Halaman Data Mahasiswa	74
Gambar 57.	<i>Mockup</i> Hapus Data Mahasiswa.....	74
Gambar 58.	<i>Mockup</i> Halaman Data Asprak	75
Gambar 59.	<i>Mockup</i> tambah Data asprak.....	75
Gambar 60.	<i>Mockup</i> Edit Data Asprak	76
Gambar 61.	<i>Mockup</i> Hapus Data Asprak	76
Gambar 62.	<i>Mockup</i> Halaman Data Kelas	77
Gambar 63.	<i>Mockup</i> Tambah Data Kelas	77
Gambar 64.	<i>Mockup</i> Edit Data Kelas	78
Gambar 65.	<i>Mockup</i> Detail Data Kelas	78
Gambar 66.	<i>Mockup</i> Hapus Data Kelas.....	79
Gambar 67.	<i>Mockup</i> Halaman Data Matkul	79
Gambar 68.	<i>Mockup</i> Tambah Data Matkul.....	80
Gambar 69.	<i>Mockup</i> Edit Data Matkul.....	80
Gambar 70.	<i>Mockup</i> Hapus Data Matkul	81
Gambar 71.	<i>Mockup</i> Halaman Data Jadwal	81
Gambar 72.	<i>Mockup</i> Tambah Data Jadwal	82
Gambar 73.	<i>Mockup</i> Edit Data Jadwal	82
Gambar 74.	<i>Mockup</i> Detail Data Jadwal	83
Gambar 75.	<i>Mockup</i> Hapus Data Jadwal.....	83
Gambar 76.	<i>Mockup</i> Halaman Profil Kampus.....	84
Gambar 77.	<i>Mockup</i> Edit Profil Kampus	84
Gambar 78.	Mockup Laporan Presensi	85
Gambar 79.	Halaman <i>Register</i>	86
Gambar 80.	Halaman Input Wajah	86
Gambar 81.	Halaman Uji Rekognisi.....	87
Gambar 82.	Halaman Login	87
Gambar 83.	Halaman <i>Dashboard</i>	88
Gambar 84.	Halaman Data Mahasiswa	88
Gambar 85.	Halaman Data Asprak	89
Gambar 86.	Halaman Data Kelas	89
Gambar 87.	Halaman Data Matkul.....	90
Gambar 88.	Halaman Data Jadwal	90
Gambar 89.	Halaman Profil Kampus	90
Gambar 90.	Laporan Presensi Mahasiswa.....	91
Gambar 91.	Mahasiswa Terdaftar Menggunakan Kacamata.....	91

DAFTAR TABEL

Tabel 1. Tabel Perbandingan	12
Tabel 2. Koordinat Kampus	18
Tabel 3. Koordinat Pengguna	18
Tabel 4. Nilai Radian	18
Tabel 5. Nilai x	18
Tabel 6. Nilai y	19
Tabel 7. Nilai R	19
Tabel 8. Nilai d	19
Tabel 9. Tabel Descriptor	22
Tabel 10. Selisih Descriptor	22
Tabel 11. Selisih Kuadrat	23
Tabel 12. Jumlah Kuadrat	23
Tabel 13. Hasil Perhitungan Descriptor	23
Tabel 14. Tabel Descriptor Lainnya	45
Tabel 15. Tabel Selisih Descriptor Lainnya	47
Tabel 16. Tabel Kuadrat Selsih Lainnya	50
Tabel 17. Jumlah Kuadrat Lainnya	53
Tabel 18. Tabel Mahasiswa	67
Tabel 19. Tabel Asprak	67
Tabel 20. Tabel Kelas	68
Tabel 21. Tabel Matkul	68
Tabel 22. Tabel Jadwal	68
Tabel 23. Tabel Anggota Kelas	69
Tabel 24. Tabel Sesi	69
Tabel 25. Tabel Presensi	69
Tabel 26. Tabel Profil Kampus	69
Tabel 27. Test Case Skenario 1	92
Tabel 28. Test Case Skenario 2	93
Tabel 29. Test Case Skenario 3	94
Tabel 30. Test Case Skenario 4	95
Tabel 31. Test Case Skenario 5	95
Tabel 32. Test Case Skenario 6	97
Tabel 33. Test Case Skenario 7	98
Tabel 34. Test Case Skenario 8	100
Tabel 35. Test Case Skenario 9	102

DAFTAR LAMPIRAN

Lampiran 1. Perhitungan Manual LBS	39
Lampiran 2. Perhitungan Manual Generasi Descriptor	41
Lampiran 3. Perhitungan Manual Face Recognition.....	45
Lampiran 4. Use Case Diagram	56
Lampiran 5. Activity Diagram	57
Lampiran 6. ERD	67
Lampiran 7. Mockup Website	71

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi mengalami perkembangan yang begitu pesat, perkembangan tersebut berdampak besar pada hampir semua bidang, termasuk bidang pendidikan. Pendidikan sendiri merupakan kegiatan yang esensial bagi manusia yang dapat menentukan arah baik dalam hal hukum, sosial, kebudayaan dan lain-lain. Dalam proses belajar mengajar di pendidikan terutama pada perkuliahan, mahasiswa dituntut untuk dapat mandiri serta disiplin dengan peraturan yang ada agar kegiatan belajar mengajar dapat terlaksana dengan baik. Salah satu kedisiplinan yang harus diterapkan oleh mahasiswa adalah dengan mengikuti perkuliahan dengan tepat waktu sesuai dengan jadwal yang sudah disusun oleh kampus (Galih dkk., 2023). Bersamaan dengan itu, pihak kampus juga harus dapat memberikan pelayanan yang baik dalam mencatat kehadiran yang dilakukan oleh mahasiswa atau melakukan presensi.

Presensi sendiri bertujuan untuk mengetahui tingkat kehadiran serta kedisiplinan seseorang. Laporan presensi menjadi sangat penting pada sebuah institusi terutama pada perkuliahan karena presensi sendiri dijadikan sebagai salah satu penilaian penting terhadap kedisiplinan seseorang mahasiswa sehingga ketepatan waktu dalam mengikuti kegiatan perkuliahan atau presensi harus dicatat dengan baik. Selain itu, catatan tersebut juga akan menjadi penilaian tersendiri yang dapat membantu dalam pemberian nilai pada suatu mata kuliah. Pada akhir semester kita dapat melihat laporan kehadiran mahasiswa tersebut yang menjadi salah satu prioritas utama untuk pemberian nilai kepada mahasiswa tersebut. Tetapi proses presensi tersebut terkadang masih banyak terjadi kesalahan didalamnya, terkhusus pada Laboratorium Ilmu Komputer FMIPA Universitas Pakuan. Sistem presensi untuk mata kuliah luring atau *offline* masih menggunakan cara manual yaitu menggunakan buku presensi, dimana setiap mahasiswa akan dipanggil satu demi satu yang mana ini akan menghabiskan banyak waktu. Selain memakan banyak waktu, seringkali terdapat perbedaan antara presensi yang tertera pada buku atau catatan dengan presensi yang terdapat pada laporan (Asvin dkk., 2021).

Agar proses presensi yang dilakukan dapat lebih cepat dan efisien maka diperlukan yang dapat membantu proses tersebut. Salah satu teknologi yang dapat digunakan adalah *face recognition* yang merupakan salah satu teknik biometrik yang dapat melakukan identifikasi wajah dan identitas seseorang juga merupakan salah satu program *artificial intelligence*, tepatnya *deep learning*, yang sejak awal kemunculannya masih diminati dan terus dikembangkan hingga saat ini. Selain menggunakan teknologi *face recognition*, aplikasi presensi ini juga menggunakan *web* sebagai *platform* untuk menjalankannya sehingga dapat memberikan beberapa kemudahan bagi mahasiswa dalam melakukan presensi seperti mahasiswa hanya perlu membuka *web* untuk dapat melakukan presensi. Selain itu, dengan adanya aplikasi yang terstruktur memudahkan admin untuk mengurus dan merekap presensi mahasiswa. Sistem presensi ini akan dikembangkan dengan bahasa pemrograman JavaScript dan PHP di mana model *face recognition* yang diimplementasikan akan dibantu dengan library *face-api.js* dan untuk model *location based service* akan dibantu dengan library *leaflet*. Model tersebut digunakan karena tercatat memiliki nilai akurasi yang tinggi. Selain itu, konsep rekognisi wajah yang diterapkan juga berbeda dari kebanyakan model *face recognition* lainnya. Pada model ini, tidak dilakukan proses klasifikasi untuk merekognisi identitas dari wajah seseorang melainkan proses regresi

untuk menghasilkan *descriptor* sebagai fitur dari wajah yang dideteksi. Kemudian, *descriptor* inilah yang dimanfaatkan untuk mengenali wajah dengan cara membandingkannya dengan *descriptor* yang telah disimpan sebelumnya. Hal ini memanfaatkan konsep *face similarity* di mana identitas seseorang akan dikenali sesuai dengan identitas wajah yang memiliki dengan nilai jarak *descriptor* paling dekat. Jarak tersebut dapat dihitung dengan algoritma *distance measure*. Algoritma yang diimplementasikan pada library *face-api.js* adalah *euclidean distance* yang dengannya diperoleh nilai akurasi yang tinggi.

Penelitian terdahulu yang sudah melakukan penelitian terkait Sistem Presensi Menggunakan *Face Recognition*, diantaranya oleh Wahyu Setiya Putra & Fadlil Adhim (2022) yang berjudul Sistem Informasi Presensi Online Menggunakan Teknologi *Face Recognition* dan *GPS*. Penelitian lain dilakukan oleh Jaini dan Asri (2021) yang berjudul Sistem Manajemen Kehadiran Menggunakan Metode *Face Recognition* Berbasis *Web*. Lalu, Fauzi Isputrawan (2023) juga meneliti terkait penggunaan *face recognition* pada sistem presensi yang memiliki judul Pengembangan Aplikasi Absensi Berbasis *Web* Menggunakan *Face Recognition*. Kemudian, Santoso & Kristianto (2020) yang juga menggunakan metode *face recognition* dalam penelitiannya yang diberi judul Implementasi Penggunaan *Opencv* Pada *Face Recognition* Untuk Sistem Presensi Perkuliahan Mahasiswa. Terakhir ada Tsani Abdurahman dan Joko Nursanto (2022) yang memiliki judul Rancang Bangun Aplikasi Kehadiran Siswa Menggunakan Pengenalan Wajah Berbasis Android Dengan Metode *Machine Learning*.

Berdasarkan uraian masalah yang sudah di paparkan diatas, penggunaan teknologi *face recognition* sebagai media untuk melakukan presensi sangat berguna dan bermanfaat karena dapat membuat proses presensi menjadi lebih efektif dan efisien. Selain itu, penggunaan aplikasi berbasis *web* dapat mempermudah mahasiswa yang memiliki *smartphone* lawas serta tempat penyimpanan yang penuh agar tidak perlu menginstall atau memasang aplikasi android. Tema penelitian ini sudah banyak dilakukan tetapi belum ada penelitian yang secara khusus dilakukan pada Laboratorium Ilmu Komputer FMIPA Universitas Pakuan sehingga diharapkan dengan adanya aplikasi *web* untuk presensi mahasiswa ini dapat meningkatkan kedisiplinan mahasiswa serta diharapkan dapat membantu admin dalam proses rekapitulasi presensi pada akhir semester.

1.2 Tujuan

Tujuan dari skripsi ini adalah untuk merancang Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*.

1.3 Ruang Lingkup

Penelitian mengenai Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance* memiliki ruang lingkup sebagai berikut:

1. Penelitian ini berfokus pada implementasi model *face recognition* dan *location based service* pada sistem presensi berbasis *web*.
2. Data yang digunakan pada penelitian ini meliputi Data Mahasiswa, Data Asprak, Data Kelas, Data Matkul dan Data Jadwal yang ada di Laboratorium Ilmu Komputer FMIPA Universitas Pakuan.
3. Data latih awal yang akan digunakan pada penelitian ini berjumlah sekitar 1037 data gambar wajah yang mana 27 gambar diambil secara langsung dan 973 gambar

diambil dari data sekunder. Kemudian, data dibagi menjadi 3 bagian yaitu *train*, *valid*, dan *test* dengan proporsi 70, 20 dan 10 persen. Lalu, dilakukan proses *augmentasi* data sehingga data bertambah menjadi 2489 gambar, sehingga proporsi akhir yaitu berkisar 2178 gambar atau 87 persen untuk data *train*, 207 gambar atau 8 persen untuk *valid*, dan 104 gambar atau 4 persen untuk *test*.

4. Dataset yang digunakan adalah dataset sendiri yang dibuat menggunakan *roboflow*.
5. *Software* yang digunakan dalam pembuatan sistem ini yaitu Visual Studio Code, Python, PHP, Javascript, MYSQL, *Roboflow*, dan *Tensorflow*.

1.4 Manfaat Penelitian

Adapun manfaat penelitian mengenai Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance* adalah sebagai berikut :

1. Mahasiswa
 - a. Memudahkan Mahasiswa dalam melakukan presensi.
 - b. Membuat proses presensi menjadi efektif dan efisien.
2. Laboratorium
 - a. Memudahkan Laboratorium Ilmu Komputer FMIPA Universitas Pakuan dalam membuat laporan rekapitulasi presensi mahasiswa.
 - b. Meminimalisir kecurangan yang dapat dilakukan dengan mengharuskan mahasiswa melakukan presensi dengan pemindaian wajah sendiri.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan Teori berisi tentang teori yang digunakan memuat acuan primer diutamakan artikel jurnal, diktat maupun artikel yang penulisnya bisa divalidasi. Jika diperlukan silahkan masukkan gambar atau tabel dari sumber yang relevan. Jika memasukkan rumus tuliskan nomor persamaan contoh persamaan 1, persamaan 2 dst. Landasan teori menyesuaikan judul penelitian dan sesuaikan dengan kebutuhan penelitian yang di bagi pada sub-sub bab. Urutannya dari Objek penelitian, Metode/Algoritma/Teknik/Sensor dan Software yang digunakan.

2.1.1 Sistem Presensi

Sistem Presensi adalah suatu model pendataan kehadiran yang merupakan bagian dari aktivitas pelaporan suatu institusi atau komponen institusi. Data-data absensi disusun dan diatur sehingga dapat dengan mudah untuk dicari dan dipergunakan ketika diperlukan oleh pihak yang berkepentingan (Habie dkk., 2022).

2.1.2 Mahasiswa

Mahasiswa adalah setiap individu yang secara resmi terdaftar mengikuti pelajaran di perguruan tinggi dengan batas usia 18 – 30 tahun dan bisa disebut sebagai suatu kelompok di dalam masyarakat yang mendapatkan statusnya karena ikatan dengan perguruan tinggi (Satwikayana dkk., 2021).

2.1.3 Face Recognition

Face Recognition merupakan sebuah teknologi yang berbasis pada *Biometric Artificial Intelligence* (AI) yang dapat mengidentifikasi seseorang dengan menganalisis pola yang berdasar pada bentuk dan tekstur wajah seseorang yang sebelumnya telah terekam dan tersimpan (Isum dkk., 2019).

2.1.4 Location Based Service

Location Based Service atau LBS adalah sebuah layanan yang berfungsi untuk mencari atau memberitahu posisi dimana perangkat berada dengan menggunakan teknologi *Global Positioning Service* atau GPS. Secara umum, teknologi ini digambarkan sebagai teknologi yang digunakan untuk menemukan perangkat pengguna (Siregar dkk., 2023). Penggunaan LBS sendiri memanfaatkan kondisi dari lingkungan dengan bantuan *geolocation* (Syakti & Oktaviani, 2020). *Geolocation* sendiri adalah sebuah perangkat API yang mana menggunakan titik lokasi bumi atau bisa disederhanakan dengan posisi lintang yang biasa disebut *latitude* dan garis bujur atau yang disebut *longitude*. Pada penggunaannya, *geolocation* dapat digunakan untuk mendapatkan lokasi titik perangkat tersebut berada (Hajar dkk., 2021). Terdapat 2 bagian penting yang terdapat pada layanan lokasi atau *Location Based Service* (Desmile dkk., 2021), antara lain:

1. *API Location Manager*

API Location Manager adalah sebuah API yang berfungsi untuk memberikan sumber daya untuk layanan lokasi seperti peta. Selain itu, API ini dapat juga digunakan untuk menampilkan, mengubah atau memanipulasi serta melakukan fungsi-fungsi lain pada peta.

2. *Location Provider*

Location Provider adalah sebuah teknologi yang berfungsi untuk melakukan pencarian lokasi pada suatu perangkat yang menggunakan data dari API lokasi yang memproses data dari GPS (*Global Positioning System*).

2.1.5 Website

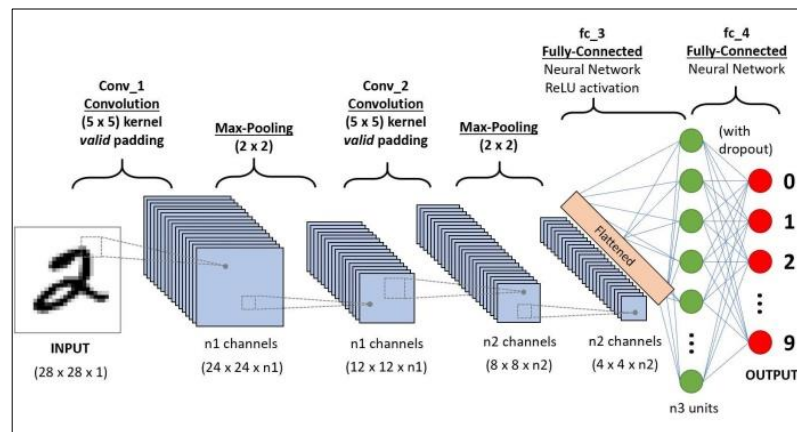
Website adalah suatu kumpulan atau gabungan halaman memiliki fungsi untuk menampilkan informasi seperti teks, tulisan, gambar, animasi, suara maupun gabungan dari semuanya dan bersifat statis maupun dinamis dan saling berkaitan serta dihubungkan dengan jaringan-jaringan pada halaman *web* (Farid Fauzi dkk., 2021).

2.1.6 Roboflow

Roboflow adalah platform penglihatan komputer atau *computer vision* yang memungkinkan pengguna untuk membangun model visi komputer dengan lebih cepat dan lebih akurat melalui penyediaan pengumpulan data, pra-pemrosesan data, dan teknik pelatihan model yang lebih baik. Salah satu fitur dari platform ini adalah pembuatan dataset *object detection* yang umumnya digunakan pada algoritma SSD MobileNetV1, *Roboflow* didirikan pada bulan Januari 2020 (Habibi, 2023).

2.1.7 Convolutional Neural Network

Convolutional neural network merupakan algoritma *deep learning*, subbidang dari machine learning, yang dirancang agar dapat melatih komputer untuk mengenali pola dari pemrosesan data menggunakan struktur yang mirip dengan jaringan saraf manusia. Algoritma ini sudah banyak dimanfaatkan di berbagai bidang seperti *image processing*, *voice recognition*, dan *natural language processing*. *Convolutional neural network* merupakan salah satu model *neural network* yang umumnya digunakan pada bidang *image processing* untuk melakukan deteksi objek, pengenalan objek, hingga pengenalan wajah. Model ini menggunakan bobot dan bias pada setiap layer. Arsitektur dalam *Convolutional neural network* dapat dilihat pada Gambar 1 (Maulida Shaliha, 2023).



Gambar 1. Arsitektur CNN

(Sumber: <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>)

2.1.8 Tensorflow

Pada 2018, Google mengeluarkan *library* baru, ML Kit, agar para pengembang dapat dengan mudah memanfaatkan machine learning di ponsel. ML kit adalah *Machine learning* yang memberi komputer kemampuan untuk belajar melalui suatu

proses yang melatih model dengan satu set input yang menghasilkan output yang diinginkan. Dengan algoritma tertentu, model yang dihasilkan mampu membuat prediksi, seperti apakah hewan yang ada dalam foto lucu atau tidak. Google menyediakan *TensorFlow* dan *TensorFlow Lite* untuk ponsel sehingga pengembang dapat membusat model pembelajaran mesin mereka sendiri dan menggunakannya di aplikasi mereka. Hal tersebut membantu jumlah yang luar biasa dalam membuat machine-learning lebih mudah diimplementasikan. Namun, hal tersebut masih dianggap cukup sulit bagi banyak pengembang. Menggunakan *TensorFlow* masih memerlukan pengetahuan tentang machine-learning, dan seringkali dibutuhkan kemampuan untuk terus berlatih dengan menggunakan model algoritma sendiri (Redhia Pratama, 2021).

2.1.9 Euclidean Distance

Algoritma ini digunakan untuk mengalkulasi jarak antara dua vektor dengan nilai *real* berupa angka, baik yang bertipe data *integer* (bilangan bulat) maupun *floating point* (bilangan desimal). Algoritma ini menghitung jumlah nilai akar dua dari selisih nilai per kolom yang dikuadratkan untuk setiap nilai pada vektor. Perhitungan jarak ini dapat dilakukan dengan menerapkan konsep *pythagoras* di mana sisi miring (*hypotenuse*) memiliki jarak terdekat antara kedua titik. Umumnya, nilai yang dihitung akan dinormalisasi terlebih dahulu sebelum kalkulasi dimulai (Maulida Shaliha, 2023).

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (1)$$

Keterangan:

d = jarak	i = indeks vektor
p, q = vektor nilai	n = total index pada vektor

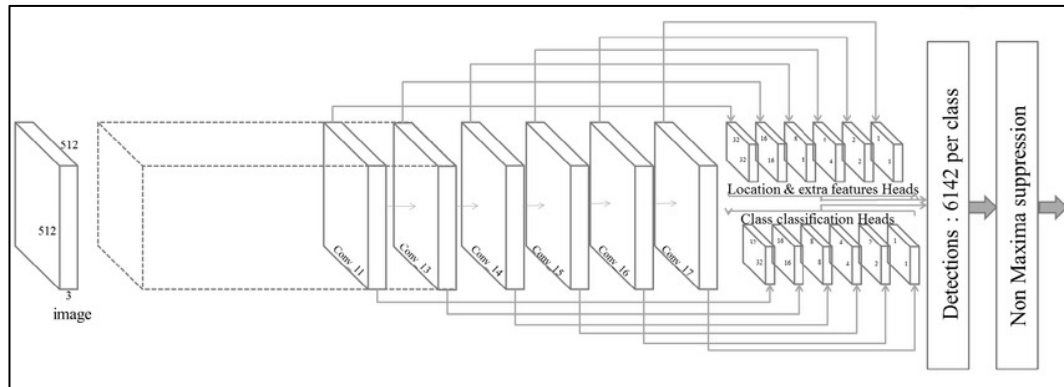
Rumus ini dapat diterapkan pada *machine learning* maupun *deep learning* seperti menentukan jarak terdekat dengan *centroid* pada k-means, mengukur jarak kedekatan fitur untuk melakukan pengenalan wajah, dan lain-lain.

2.1.10 Face-api.js

Face-api.js merupakan salah satu *library* yang menyediakan model *machine learning* untuk beberapa macam keperluan seperti *face detection*, *face landmark detection*, *face recognition*, dan lain-lain. Model-model tersebut dikembangkan menggunakan bahasa pemrograman JavaScript dengan dukungan *Tensorflow.js*, sebuah *library* bersifat *open-source* yang dapat mendukung pengembangan proyek *machine learning* berbasis *web* sehingga dapat dijalankan secara langsung pada browser (Maulida Shaliha, 2023). Berikut ini merupakan tiga di antara semua model yang disediakan oleh *library face-api.js*:

a. *ssd_mobilenetv1_model*

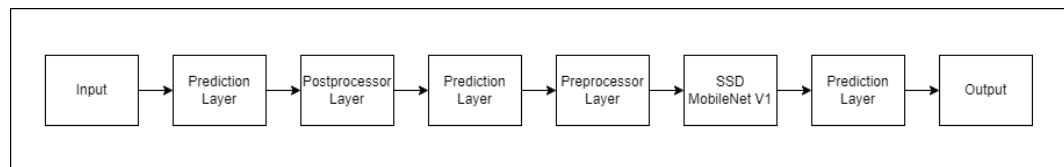
Model ini menerapkan arsitektur *SSD MobileNet V1*, yang disebut sebagai salah satu model pendeteksi objek paling baik, dengan penambahan beberapa convolutional layer pada bagian awal dan bagian akhir. Model ini digunakan untuk mendeteksi ada atau tidaknya wajah pada gambar yang diinput. Apabila terdeteksi adanya wajah, maka model akan memberikan output berupa lokasi *bounding box* dan *confidence score*. Gambar 2 merupakan arsitektur dari model *SSD MobileNet V1*.



Gambar 2. Arsitektur Model *SSD MobileNet V1*

(Sumber: <https://iq.opengenus.org/ssd-mobilenet-v1-architecture/>)

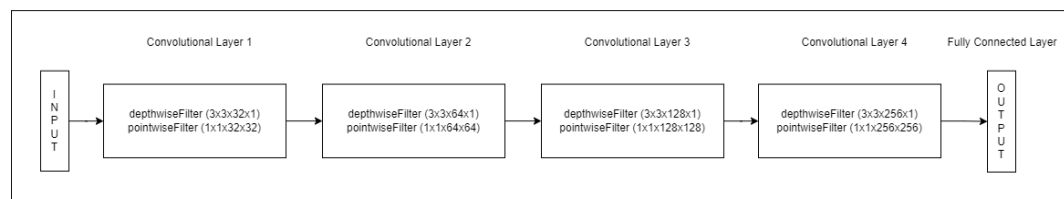
Sehingga, secara sederhana, arsitektur akhir dari model *ssd_mobilenetv1* yang telah dimodifikasi pada *library face-api.js* adalah seperti yang dapat dilihat pada Gambar 3.



Gambar 3. *face-api.js - ssd_mobilenetv1_model*

b. *face_landmark_68_model*

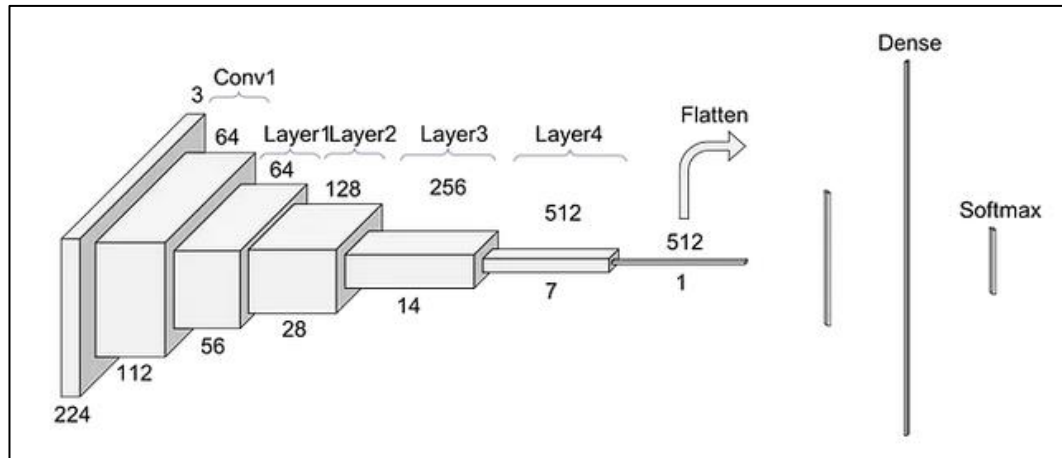
Model ini dibuat oleh pengembang *library face-api.js* dan memiliki jumlah *layer* yang tidak terlalu banyak. Fungsi dari model ini adalah memperoleh *face landmark* dari wajah yang terdeteksi oleh model *ssd_mobilenetv1*. Dalam kasus *face recognition*, *face landmark* dimanfaatkan untuk memotong (*crop*) gambar sehingga yang tersisa adalah gambar wajah dengan posisi tepat di tengah. Hal ini dilakukan untuk meningkatkan akurasi pada proses rekognisi. Secara sederhana, arsitektur dari model ini dapat dilihat pada Gambar 4 di mana untuk masing-masing *convolution layer* dilakukan perulangan sebanyak empat kali.



Gambar 4. *face-api.js - face_landmark_68_model*

c. *face_recognition_model*

Model *face_recognition* yang disediakan oleh *library face-api.js* menerapkan arsitektur *ResNet-34* seperti yang tertera pada Gambar 5. Model ini berperan untuk menghasilkan *descriptor* wajah berdasarkan *face landmark* yang dihasilkan oleh model *face_landmark_68* dimana *descriptor* merupakan sebuah baris data dengan nilai numerik yang setiap angkanya merepresentasikan fitur dari wajah seseorang. Dengan adanya *descriptor*, proses rekognisi wajah seseorang dapat dilakukan dengan melihat jarak kemiripan fiturnya menggunakan algoritma *euclidean distance*.



Gambar 5. Arsitektur Model *ResNet 34*

(Sumber: <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8>)

2.1.11 Javascript

JavaScript merupakan salah satu bahasa pemrograman tingkat tinggi yang banyak digunakan saat ini. Bahasa pemrograman ini dapat dimanfaatkan pada bagian *front-end* maupun *back-end* aplikasi. Dengan JavaScript, seorang developer dapat menanamkan logika pada skrip HTML agar hasilnya menjadi lebih dinamis serta sebagai berkas individu yang dapat dimanfaatkan untuk keperluan manipulasi data (Maulida Shaliha, 2023).

2.1.12 Python

Python adalah bahasa pemrograman *open source* yang memiliki status sebagai bahasa pemrograman dengan pertumbuhan tercepat dan mengalami peningkatan yang tajam dalam penggunaannya terutama untuk ilmu data dan bidang. Selain itu, Python juga memiliki komunitas *Stackoverflow* terbesar di dunia yang merupakan situs tanya jawab programming terbesar di dunia sehingga dapat mempermudah para pengguna untuk menyelesaikan masalah kode yang mereka alami (Redhia Pratama, 2021).

2.1.13 PHP

PHP atau *Perl Hypertext Preprocessor* adalah bahasa pemrograman *web* yang bersifat *open source* dan berjalan pada bagian *server* atau *server-side*. PHP sendiri merupakan *script* yang dapat berintegrasi dengan HTML (Hypertext Markup Language) dalam membuat halaman website menjadi lebih dinamis (Nilfaidah dkk., 2021). Penjelasan lain mengenai PHP adalah bahasa *server-side-scripting* yang menyatu atau berintegrasi dengan HTML dengan tujuan untuk membuat halaman *website* menjadi lebih dinamis. Perintah atau *syntax* pada PHP akan dieksekusi atau dijalankan pada *server* yang hasilnya akan diberikan kepada pengguna atau *client* dengan format HTML. Dengan demikian, kode atau *script* PHP yang dijalankan pada suatu *website* tidak akan terlihat oleh pengguna sehingga keamanan lebih terjamin. Selain itu, PHP dirancang agar dapat membuat sebuah *website* lebih dinamis, yang artinya informasi pada halaman web akan selalu *up to date* sesuai dengan permintaan terbaru dari *client* (Herman & Wijaya, 2021). PHP memiliki beberapa kelebihan dibandingkan bahasa pemrograman lain dalam membangun sebuah web (Rianto Sitanggang, 2022), diantaranya:

1. Bahasa pemrograman PHP bersifat *open-source* yang berarti setiap orang dapat menggunakan maupun mengubah bahasa ini secara gratis.
2. PHP sendiri merupakan bahasa multiplatform sehingga bahasa ini dapat berjalan pada berbagai mesin dan sistem informasi seperti Linux, Windows, Macintosh dan lain-lain.
3. Dalam penggunaannya, banyak *web server* yang sudah mendukung bahasa PHP seperti Apache, nginx dan lain-lain. Selain itu, konfigurasi yang dilakukan relatif mudah dan cepat.
4. Pengembangan menggunakan bahasa PHP lebih mudah dikarenakan besarnya komunitas dan banyaknya dokumentasi yang diberikan sehingga memudahkan dalam proses pengembangan *website*.
5. PHP sendiri mendukung banyak basis data atau *database* seperti MySQL, Oracle, dan lain-lain.

2.1.14 Bootstrap

Bootstrap merupakan sebuah *framework* CSS atau *Cascading Style Sheets* yang digunakan untuk mendesain suatu *website* agar halamannya menjadi responsif. Halaman yang responsif artinya halaman tersebut akan menyesuaikan lebar maupun tinggi secara otomatis dengan *device* yang digunakan sehingga halaman *website* akan tetap bagus walaupun dibuka melalui *smartphone* ataupun *tablet*. Bootstrap sendiri adalah jenis *framework* yang dibuat dengan menggabungkan CSS dan JavaScript oleh Mark Otto dan Jacob Thornton dengan tujuan agar dapat menghadirkan konsistensi pada tahap *interface development* atau tahap pengembangan antarmuka dalam sebuah *website* (Anis dkk., 2022). Penjelasan lain mengenai *framework* Bootstrap adalah sebuah *library framework* CSS yang dibuat secara khusus untuk pengembangan *website* bagian *front-end* atau bagian tampilan. *Framework* Bootstrap cukup populer dan banyak digunakan oleh kalangan *developer* atau pengembang *web* dalam mendesain tampilan *website* mereka agar menjadi lebih responsif (Kadek dkk., 2020). Beberapa keunggulan Bootstrap dalam pengembangan *website* bagian front-end (Widayati dkk., 2021), diantaranya:

1. Membuat tampilan pada halaman *website* yang dikembangkan menjadi lebih responsif pada berbagai *device*.
2. Dapat mempercepat proses pengembangan *website* terkhusus pada pengembangan antarmuka sebuah *website*.
3. *Website* yang menggunakan Bootstrap cenderung lebih ringan dan cepat.
4. Tampilan *website* menjadi lebih modern berkat kelas-kelas serta *icon* dan *font* yang disediakan.

2.1.15 MySQL

MySQL merupakan sebuah *database engine* atau basis data atau *server database* yang mendukung bahasa SQL atau *Structured Query Language* sebagai bahasa untuk mengelola data (Rianto Sitanggang, 2022). Pengertian lain, MySQL sendiri merupakan sebuah RDBMS atau *Relational Database Management System* yang sudah dilisensikan dibawah GPL (*General Public License*). MySQL sendiri merupakan sebuah turunan dari konsep utama pada *database* yaitu SQL (Nilfaidah dkk., 2021).

2.2 Penelitian Terdahulu

1. Nama : Yusuf Wahyu Setiya Putra dan Muhammad Fadlil Adhim
Judul : Sistem Informasi Presensi Online Menggunakan Teknologi *Face Recognition* dan GPS
Tahun : 2022
Isi : Sistem presensi di SMK Muhammadiyah 1 Weleri masih menggunakan cara konvensional sehingga membutuhkan waktu yang lama dan rentan adanya manipulasi data maupun data ganda. Perancangan system informasi presensi online ini dibangun dengan teknologi GPS dan *Face Recognition* yang berbasis android dan *web*. Sistem ini dirancang dengan menggunakan PHP sebagai bahasa pemrograman, dan MySQL sebagai basis data. Dalam Pembuatan system pada penelitian ini penulis menggunakan metode *Software Development Life Cycle (SDLC) Waterfall*. Kemudian untuk pengujian menggunakan *User Acceptance Testing (UAT)* untuk menguji sistem terhadap spesifikasi aplikasi dengan membagikan kuisioner untuk responden. Pengujian tersebut akan dilakukan oleh guru dan karyawan di SMK Muhammadiyah 1 Weleri. Penelitian ini bertujuan untuk memudahkan pengelolaan presensi guru dan karyawan, sehingga dapat memberikan informasi secara lengkap dan akurat, serta dapat melakukan monitoring presensi tersebut dengan efektif dan efisien. Kemudian hasil dari uji sistem menggunakan *User Acceptance Testing (UAT)* sebesar 93,8% yang menandakan sistem ini dapat diterima oleh penggunanya.
2. Nama : Nur Jaini, Ervan Asri, dan Fitri Nova
Judul : Sistem Manajemen Kehadiran Menggunakan Metode *Face Recognition* Berbasis *Web*
Tahun : 2021
Isi : Sistem absensi dengan pemanfaatan *face recognition* adalah sistem yang memanfaatkan biometrik berupa pola wajah dan membandingkan wajah yang terdeteksi (*face detect*) dengan wajah yang tersimpan, Untuk menangkap pola wajah dari user yang akan absen menggunakan sebuah modul camera esp32-cam yang ditanamkan *library face recognition* sehingga mampu membandingkan wajah yang terdeteksi dengan yang tersimpan. Sistem ini terintegrasi dengan *web server* yang di rancang untuk mengatur dan menyimpan rakap absen dari mahasiswa sehingga satu alat *face detect* dapat di gunakan oleh lebih dari satu kelas secara bersamaan. *Web server* di kontrol oleh admin untuk mengatur jadwal absen mulai dari jadwal mata kuliah, jadwal kelas serta jam masuk dan jam keluar, sehingga dengan pemanfaatan *web server* tidak memungkinkan ada nya data ganda atau absensi lebih dari satu kali.
3. Nama : M. Fauzi Isputrawan dan Suriyanti
Judul : Pengembangan Aplikasi Absensi Berbasis *Web* Menggunakan *Face Recognition*
Tahun : 2023
Isi : Penelitian ini bertujuan untuk mengembangkan Aplikasi Absensi Berbasis *Web* Menggunakan *Face Recognition* untuk kemudian digunakan dalam *work from home*. Penggunaan aplikasi absensi berbasis website ini dapat membantu dalam pengajuan cuti, pembuatan laporan, penyimpanan, pengolahan data-data kepegawaian dan absensi menggunakan pindai wajah. Aplikasi absensi ini menggunakan bahasa pemograman PHP dengan metode pengembangan SDLC dan model *waterfall*. Dari hasil penelitian yang dilakukan telah disimpulkan bahwa

aplikasi absensi ini membantu mengatasi masalah pada perusahaan dalam kegiatan manajemen kepegawaian.

4. Nama : Banu Santoso dan Ryan Putranda Kristianto
Judul : Implementasi Penggunaan *Opencv* Pada *Face Recognition* Untuk Sistem Presensi Perkuliahan Mahasiswa
Tahun : 2020
Isi : Data presensi perkuliahan pada perguruan tinggi menjadi suatu acuan dalam menunjukkan kredibilitas setiap mahasiswa yang digunakan oleh para dosen sebagai data untuk pemberian nilai mahasiswa sekaligus sebagai bahan evaluasi keberhasilan kegiatan belajar mengajar dalam perkuliahan, namun ada beberapa contoh kasus, terkait dengan data presensi mahasiswa yang saat ini marak terjadi pada dunia pendidikan atau perkuliahan adalah fenomena “Titip Absen”. Selain itu, masalah lainnya juga muncul dari pihak dosen dan pegawai tata usaha yakni kesulitan dalam memonitor kehadiran mahasiswa serta upaya memvalidasi data presensi karena jumlah data mahasiswa yang begitu banyak. Oleh karena itu dalam penelitian ini diajukan suatu sistem untuk mengurangi tingkat kecurangan dalam pengisian daftar presensi dan efektivitas pengolahan data mahasiswa dengan menggunakan sistem penerapan metode *Face Recognition* berbasis *Open CV* dengan metode *Haar Cascade Classifier* dan *Local Binary Patterns Histograms* (LBPH). Hasil penelitian *Face Recognition* ini berhasil mendeteksi apabila semua pengguna yang dipresensikan telah terdaftar ke sistem, dengan jarak jangkauan optimal *Face Recognition* agar terdeteksi sampai 150 cm. Sedangkan *Face Recognition* tidak berhasil terdeteksi apabila ada obstacle menutupi objek wajah dan jarak melebihi dari 150 cm.
5. Nama : Iskandar, Umar Tsani Abdurahman, dan Joko Nursanto
Judul : Rancang Bangun Aplikasi Kehadiran Siswa Menggunakan Pengenalan Wajah Berbasis Android Dengan Metode *Machine Learning*
Tahun : 2022
Isi : *Machine learning* merupakan suatu pendekatan dalam *Artificial Intelligence* yang banyak diaplikasikan dalam menirukan perilaku manusia untuk penyelesaian masalah secara otomatisasi. Pemanfaatan machine learning salah satunya *face recognition* atau pengenalan wajah dalam proses kehadiran siswa yang bisa mengenali kehadiran secara realtime menggunakan pengolahan data wajah dari siswa. Sistem kehadiran siswa yang berjalan selama ini masih dilakukan secara manual oleh guru dengan mencatatkan pada daftar hadir siswa, dimana dalam pelaksanaannya kadangkala terjadi kesalahan tulis serta membutuhkan waktu relatif lama dalam proses pencatatannya. Seiring dengan kemajuan jaman pengelolaan sistem kehadiran siswa, manajemen membutuhkan sistem yang *flexibel*, tepat guna dan aman terhadap perubahan kebutuhan manajemen sekolah. Dalam penelitian ini dirancang sistem kehadiran siswa berbasis mobile android dengan menggunakan *machine learning* untuk sistem kehadiran dengan menggunakan pengenalan wajah menggunakan metode algoritma K-Nearest Neighbor dengan library Python 3.8, TensorFlow Lite dan sistem manajemen database MySQL. Dari hasil implementasi dan pengujian model menghasilkan nilai akurasi 94.66%, presisi 94.41%, recall 94.20%, 1 k-fold score 94.19, untuk pengujian dari sisi aplikasi *BackEnd* dan *FrontEnd* menunjukkan kecepatan dan keakuratan sudah memadai untuk dapat digunakan pada satu institusi kelas atau

sekolah. Kestabilan *server face_api* juga cukup memadai selama testing dapat berjalan tanpa gangguan lebih dari 7x24 jam.

2.3 Tabel Perbandingan Penelitian

Tabel penelitian terdahulu ini digunakan untuk melihat perbandingan dengan penelitian yang telah dilakukan pada sebelumnya untuk bahan referensi.

Tabel 1. Tabel Perbandingan

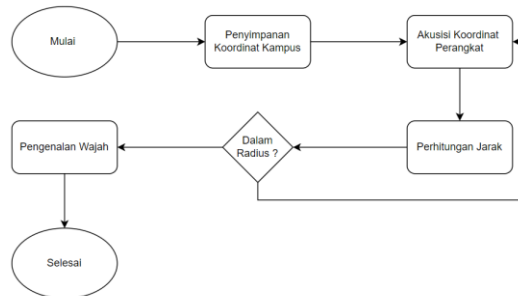
No	Nama Peneliti	Metode Penelitian			Pengembangan		
		SDLC	Euclidean Distance	LBS	Android	Website	Prototype
1	Yusuf Wahyu Setiya Putra, Muhammad Fadlil Adhim (2022)	✓	-	✓	-	✓	-
2	Nur Jaini, Ervan Asri, Fitri Nova (2021)	✓	-	-	-	✓	-
3	M. Fauzi Isputrawan, Suriyanti (2023)	✓	-	-	-	✓	-
4	Banu Santoso, Ryan Putranda Kristianto (2020)	✓	-	-	-	-	✓
5	Iskandar, Umar Tsani Abdurahman, Joko Nursanto (2022)	✓	-	-	✓	-	-
6	Muhammad Taofik (Sekarang)	✓	✓	✓	-	✓	-

BAB III METODE PENELITIAN

3.1 Metodologi Penelitian

3.1.1 Location Based Service (LBS)

Location Based Service atau LBS adalah sebuah layanan yang berfungsi untuk mencari atau memberitahu posisi dimana perangkat berada dengan menggunakan teknologi *Global Positioning Service* atau GPS. Gambar 6 merupakan alur LBS.



Gambar 6. Alur LBS

3.1.1.1 Penyimpanan Koordinat Kampus

Tahap penyimpanan koordinat kampus dilakukan dengan memasukan koordinat kampus yang berupa *latitude* dan *longitude*, dan radius toleransi jarak presensi.

3.1.1.2 Akusisi Koordinat Perangkat

Tahap akuisisi koordinat perangkat dilakukan dengan mendapatkan data koordinat *longitude* dan *latitude* dari sensor GPS pada perangkat.

3.1.1.3 Perhitungan Jarak

Tahap ini merupakan tahap inti dari metode *Location Based Service* Dari kedua koordinat yang telah disimpan, diterapkan algoritma *equiangular approximation* yang akan menghasilkan jarak kedua koordinat. Dalam metode ini, digunakan rumus jarak *euclidean* pada sistem koordinat kartesian untuk menghitung jarak linear antara dua titik.

$$x = \Delta\lambda \times \cos(\varphi m) \quad (2)$$

$$y = \Delta\varphi \quad (3)$$

$$d = R \times \sqrt{x^2 + y^2} \quad (4)$$

Keterangan:

φ = latitude

λ = logitude

y = posisi vertikal pada peta

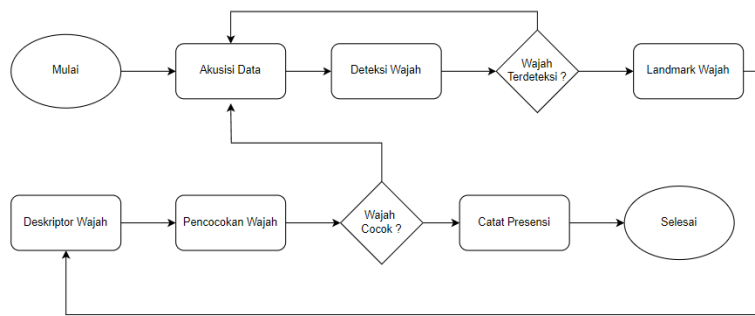
R = radius bumi

x = posisi horizontal pada peta

d = jarak antara kedua posisi

3.1.2 Face Recognition

Pengenalan wajah menggunakan *face-api.js* dilakukan melalui beberapa proses. Proses proses tersebut yaitu akuisisi data, pengenalan wajah, *landmark* wajah, *descriptor* wajah, dan pencocokan wajah seperti pada Gambar 7.



Gambar 7. Alur *Face Recognition*

3.1.2.1 Akusisi Data

Tahap ini merupakan tahap pengambilan data wajah untuk proses pengenalan wajah yaitu akuisisi data dengan mendaftarkan diri dan akuisisi data *webcam* yang akan digunakan sebagai pembandingan pencocokan wajah.

3.1.2.2 Deteksi Wajah

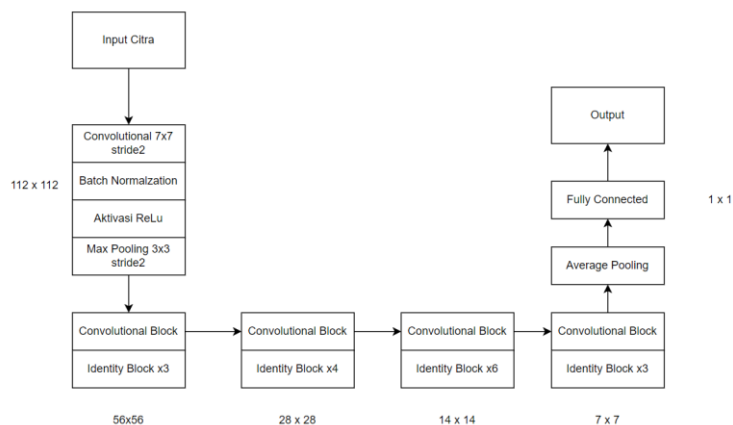
Tahap ini merupakan proses dilakukan perubahan ukuran citra yang diambil melalui *frame* dalam *webcam* menjadi maksimal 512px untuk salah satu nilai dimensi citra tertinggi dan melakukan penyesuaian skala untuk dimensi citra terendah.

3.1.2.3 Feature Extraction

Tahap ini merupakan proses kalkulasi ekstraksi fitur 68 titik *landmark* wajah menggunakan model *face_landmark_68_model* yang merupakan sebuah *pre-trained* model yang berjalan diatas *tensorflow* dimana model tersebut menghasilkan array 68 titik koordinat x dan y dari masing-masing landmark wajah yang meliputi alis kanan, alis kiri, mata kanan, mata kiri, hidung, mulut, dan *outline* wajah serta mengubah ukuran citra menjadi 224 piksel.

3.1.2.4 Face Descriptor

Tahap ini merupakan proses ekstraksi *face descriptor*, yang dilakukan menggunakan model *face_recognition_model*. Model tersebut menggunakan arsitektur ResNet-34 yang diimplementasikan untuk mengkomputasi *face descriptor* (sebuah *feature vector* dengan 128 nilai) dari sebuah gambar wajah, yang digunakan untuk mendeskripsikan karakteristik wajah seseorang. Proses alur ekstraksi citra pada arsitektur ResNet-34 ditunjukkan pada gambar 8.



Gambar 8. Alur Transformasi Gambar

Proses Secara detail dijelaskan sebagai berikut :

1. Data citra berukuran 224 x 224 piksel sebagai data masukan.
2. Data Masukan diproses dengan menambahkan zero padding berukuran 3.

Tahap 1

1. Melakukan operasi convolutional dengan filter 7 x 7 dengan 64 channel dengan stride = 2.
2. Melakukan operasi fungsi batch normalization.
3. Menambahkan fungsi aktivasi menggunakan ReLu.
4. Melakukan operasi fungsi maxpooling.
5. 3 x 3 dengan stride = 2.
6. Output dari proses tahapan ini menghasilkan ukuran citra 112 x 112 piksel.

Tahap 2

1. Melakukan operasi convolusi block dengan filter 3 x 3 dan 3 x 3 dengan channel 64 dan 64.
2. Melakukan operasi *identity block* dengan filter 3 x 3 dan 3 x 3 dengan channel 64 dan 64 serta stride = 1 yang dilakukan perulangan sebanyak 3 kali.
3. Output dari proses tahapan ini menghasilkan ukuran citra 56 x 56 piksel.

Tahap 3

1. Melakukan convolusi block dengan filter 3 x 3 dan 3 x 3 dengan channel 128 dan 128.
2. Melakukan operasi *identity block* dengan filter 3 x 3 dan 3 x 3 dengan channel 128 dan 128 yang dilakukan perulangan sebanyak 4 kali.
3. Output dari proses tahapan ini menghasilkan ukuran citra 28 x 28 piksel.

Tahap 4

1. Melakukan convolusi block dengan filter 3 x 3 dan 3 x 3 dengan channel 256 dan 256.
2. Melakukan operasi *identity block* dengan filter 3 x 3 dan 3 x 3 dengan channel 256 dan 256 yang dilakukan perulangan sebanyak 6 kali.
3. Output dari proses tahapan ini menghasilkan ukuran citra 14 x 14 piksel.

Tahap 5

1. Melakukan *convolusi block* dengan filter 3 x 3 dan 3 x 3 dengan channel 512 dan 512.
2. Melakukan operasi *identity block* dengan filter 3 x 3 dan 3 x 3 dengan channel 512 dan 512 yang dilakukan perulangan sebanyak 3 kali.
3. Output dari proses tahapan ini menghasilkan ukuran citra 7 x 7 piksel.

Setelah proses ekstraksi selesai, kemudian hasil output *convolusi* tersebut diproses dengan operasi *average pooling*. Selanjutnya, melakukan proses *flatten* sehingga mendapatkan *output array* satu dimensi. Output dari operasi *flatten* akan diproses menggunakan *fully connected layer* sebagai layer prediksi dan menghasilkan hasil akhir.

3.1.2.5 Pencocokan Wajah

Tahap ini merupakan tahap inti dari proses *face recognition* menggunakan algoritma *euclidean distance*, langkah-langkahnya adalah sebagai berikut:

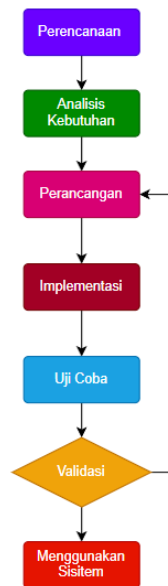
1. Siapkan data wajah yang akan diuji dan data wajah referensi yang sudah tersimpan.
2. Untuk setiap data wajah, ekstrak fitur wajah menggunakan teknik *Convolutional Neural Network (CNN)* dan ResNet-34.
3. Hitung vektor fitur wajah untuk setiap data wajah yang akan diuji dan data wajah referensi menggunakan teknik ekstraksi fitur yang dipilih.

4. Hitung jarak *euclidean* antara vektor fitur wajah untuk setiap data wajah yang akan diuji dan data wajah referensi.
5. Ambil jarak terkecil sebagai hasil pencocokan wajah.

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (5)$$

3.1.3 Systems Development Life Cycle (SDLC)

SDLC (*Systems Development Life Cycle*) *Waterfall* adalah proses pengembangan software yang berurutan dimana prosesnya mengalir ke bawah seperti air terjun (Nagara dkk., 2023). Tahap-tahap dalam metode *System Development Life Cycle* (SDLC) dapat dilihat pada Gambar 9.



Gambar 9. Tahap *system development life cycle* (SDLC)

3.1.3.1 Perencanaan

Pada tahap ini, diadakan pertemuan dengan *stakeholder* untuk mengidentifikasi produk yang akan dikembangkan. Selain itu, dilakukan juga perencanaan jadwal dan biaya, serta *tech stack* atau *tools* apa saja yang akan digunakan selama proses pengembangan nantinya.

3.1.3.2 Analisis Kebutuhan

Pada tahapan pertama ini, berfokus pada menganalisa kebutuhan sistem yang diperlukan agar sistem yang dibuat tepat dapat menyelesaikan permasalahan yang ada. Pada tahapan ini juga, akan didefinisikan kebutuhan fitur serta fungsi dari *software* atau aplikasi yang akan dibuat.

3.1.3.3 Perancangan

Pada tahap ini, berfokus pada perancangan struktur data pada database, arsitektur dari *software* yang akan dibuat, tampilan atau antarmuka aplikasi serta algoritma yang akan digunakan, dengan tujuan agar memahami aplikasi yang akan dikerjakan.

3.1.3.4 Implementasi

Pada tahap ini, dilakukan pengembangan aktual produk berdasarkan desain dan arsitektur yang telah dirancang. *Tools* maupun bahasa pemrograman yang digunakan tergantung pada jenis perangkat lunak yang dikembangkan.

3.1.3.5 Uji Coba

Pada tahapan ini, akan dilakukan uji coba atau testing pada aplikasi yang sudah dibuat. Hal ini bertujuan untuk mengetahui apakah terdapat kesalahan baik kesalahan yang disebabkan oleh *logical error* atau kesalahan logika maupun *syntax error* atau kesalahan pengkodean.

3.2 Tempat Penelitian

Penelitian ini dilaksanakan di Laboratorium Ilmu Komputer FMIPA Universitas Pakuan.

3.3 Alat dan Bahan

Berikut ini adalah alat dan bahan yang digunakan dalam penelitian mengenai Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*.

3.3.1 Alat Penelitian

Alat yang dibutuhkan pada penelitian ini berupa perangkat lunak (*software*) dan perangkat keras (*hardware*) yaitu :

a. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan adalah:

1. Sistem Operasi Windows 11 64bit, Google Chrome
2. Lucidchart
3. Figma
4. Microsoft Office
5. XAMPP
6. Visual Studio Code
7. Roboflow
8. Draw.io

b. Perangkat Keras (*Hardware*)

Perangkat Keras yang digunakan adalah:

1. Laptop AMD Ryzen 5 4600HS with Radeon Graphics 3.00 GHz
2. RAM 16.00 (15,4 GB usable)
3. SSD 512 GB

3.3.2 Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian ini sebagai sumber informasi yaitu sebagai berikut.

1. Data Mahasiswa, Data Asprak, Data Kelas, Data Matkul dan Data Jadwal yang ada di Laboratorium Ilmu Komputer FMIPA Universitas Pakuan.
2. Jurnal dan artikel pada *website* sebagai referensi yang terkait dengan judul penelitian.
3. Buku panduan penulisan skripsi dan tugas akhir Ilmu Komputer FMIPA Universitas Pakuan.

BAB IV RANCANGAN DAN IMPLEMENTASI

4.1 *Location Based Service (LBS)*

4.1.1 Penyimpanan Koordinat Kampus

Penyimpanan koordinat kampus dilakukan oleh bagian admin dengan memasukan koordinat kantor yang berupa *latitude* dan *longitude*, dan toleransi jarak presensi.

Tabel 2. Koordinat Kampus

Koordinat Kampus		Batas Radius
Latitude [0]	Longitude [1]	
-6.599206553911968	106.81238825180455	200 m

4.1.2 Akuisisi Koordinat Perangkat

Akuisisi koordinat perangkat dilakukan dengan menggunakan fungsi javascript `navigator.geolocation.getCurrentPosition()`. Dengan menjalankan fungsi tersebut akan didapatkan data koordinat *longitude* dan *latitude* dari sensor GPS pada perangkat. Koordinat kemudian disimpan dalam variabel javascript yang kemudian dikalkulasikan dengan data koordinat kampus yang telah disimpan dalam *database*. Gambar 25 merupakan proses akuisisi koordinat perangkat.

Tabel 3. Koordinat Pengguna

Koordinat Perangkat	
Latitude [0]	Longitude [1]
-6.648272229626856	106.82668406899835

4.1.3 Perhitungan Jarak

Dari kedua koordinat yang telah disimpan, diterapkan formula *equirectangular approximation* yang akan menghasilkan jarak kedua koordinat. Sebagai contoh, digunakan koordinat perangkat yang berada di luar radius kampus untuk perhitungan jarak koordinat menggunakan algoritma *equirectangular approximation*. Perhitungan manual LBS secara lengkap dapat dilihat pada Lampiran 1.

1. Konversi Nilai Koordinat

Lakukan konversi pada koordinat k1-k2 dari bilangan *degree* ke bilangan radian.

Tabel 4. Nilai Radian

Koordinat Perangkat (k1)		Koordinat Kampus (k2)	
Latitude [0]	Longitude [1]	Latitude [0]	Longitude [1]
-0.11603423998	1.864477366	-0.11517788238	1.8642278569

2. Tentukan Nilai x

Setelah dilakukan konversi, tahap selanjutnya adalah mencari nilai x (posisi horizontal pada peta) dari k1-k2.

Tabel 5. Nilai x

Nilai x	- 0.00024794
---------	--------------

3. Tentukan Nilai y

Tahap selanjutnya adalah mencari nilai y (posisi vertikal pada peta) dari k1-k2.

Tabel 6. Nilai y

Nilai y	0.0008563576
---------	--------------

4. Tentukan Nilai R

Nilai R merupakan nilai mutlak pada metode *equirectangular approximation* dengan nilai sebagai berikut.

Tabel 7. Nilai R

Nilai R	6371009 m
---------	-----------

5. Tentukan Nilai d

Setelah semua nilai didapatkan, tahap terakhir adalah mengukur jaraknya dengan mencari nilai d (jarak antara kedua posisi) dari k1-k2 menggunakan algoritma *euclidean* untuk menghitung jarak linear antara dua titik.

Tabel 8. Nilai d

Nilai d	5679.51 m
---------	-----------

Jadi, total jarak perangkat pengguna ke Universitas Pakuan adalah 5679 meter. Berdasarkan batas radius yang ditentukan sekitar 200 meter maka dapat disimpulkan bahwa perangkat pengguna berada diluar radius kampus.

4.2 Face Recognition

4.2.1 Akuisisi Data

Terdapat dua tahap akuisisi data untuk proses pengenalan wajah yaitu akuisisi data dengan mendaftarkan diri dan akuisisi data *webcam* yang akan digunakan sebagai pembandingan pencocokan wajah.

Akuisisi data dengan mendaftarkan diri dilakukan dengan cara mahasiswa melakukan *register* dan melakukan pendaftaran wajah agar *descriptor* bisa didapatkan kemudian *descriptor* tersebut tersimpan dalam *database*.

Akuisisi data kedua dilakukan menggunakan fitur *webcam* yang terintegrasi dengan HTML dan JavaScript. Dalam *frame-frame* yang diambil melalui *webcam* akan dilakukan proses pengenalan wajah dengan menggunakan *frame* secara bergiliran tiap perulangan setelah wajah terdeteksi dan telah dilakukannya pencocokan wajah atau setelah proses pendeteksian selesai dan wajah tidak ditemukan. Hal ini dilakukan agar sistem tidak terbebani apabila harus melakukan proses-proses pencocokan wajah dari tiap *frame* yang digenerasi oleh *webcam*, melainkan hanya melakukan proses pencocokan pada *frame* pertama dan *frame* selanjutnya setelah pencocokan wajah selesai dijalankan pada langkah sebelumnya. Dengan begitu sistem tetap mampu berjalan secara optimal menggunakan *resource* seminimal mungkin. Gambar 10 merupakan akuisisi data wajah.



Gambar 10. Akuisisi Data Wajah

4.2.2 Deteksi Wajah

Pada proses deteksi wajah, dilakukan perubahan ukuran citra yang diambil melalui *frame* dalam *webcam* menjadi maksimal 512px untuk salah satu nilai dimensi citra tertinggi dan melakukan penyesuaian skala untuk dimensi citra terendah. Setelah dilakukan perubahan ukuran, citra diproses menjadi nilai *pixel tensor* dengan menggunakan fungsi `tf.browser.fromPixels()`. Fungsi tersebut mengkonversi gambar menjadi *array matrix* dengan nilai 0-255 untuk tiap *pixel* yang merepresentasikan gambar dari suatu elemen HTML (dalam hal ini, *frame* dari *webcam*) yang kemudian akan diolah untuk deteksi wajah.

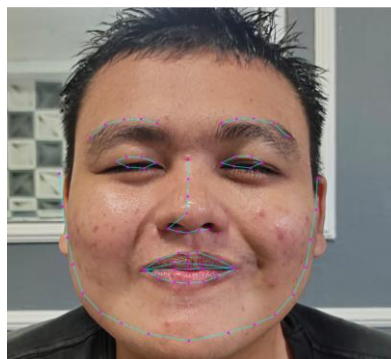
Dari gambar yang telah diolah menjadi *matrix*, dilakukan pendeteksian wajah dengan menggunakan model *tensorflow SSDMobileNetV1* yang merupakan sebuah pre-trained model yang menggabungkan *Single Shot MultiBox Detector (SSD) framework* dengan *MobileNetV1 convolutional neural network*. Telah dilakukan *pre-training* terhadap model *SSDMobileNetV1*. Hasil keluaran dari model *SSDMobileNetV1* adalah titik x dan y serta tinggi dan lebar dari objek yang terdeteksi pada gambar. Berikut merupakan penjelasan untuk melakukan *training* pada model *SSDMobileNetV1*.



Gambar 11. Deteksi Wajah

4.2.3 Feature Extraction

Dari hasil deteksi wajah pada langkah sebelumnya, akan dilakukan proses kalkulasi ekstraksi fitur 68 titik *landmark* wajah menggunakan model *face_landmark_68_model* yang merupakan sebuah *pre-trained* model yang berjalan diatas *tensorflow*. Model tersebut sudah dilatih menggunakan 35 ribu *dataset* gambar wajah yang telah diberi label untuk 68 titik *landmark* wajah. Sintaks yang digunakan untuk kalkulasi adalah `withFaceLandmarks()` dimana sintaks tersebut akan menghasilkan array 68 titik koordinat x dan y dari masing-masing landmark wajah yang meliputi alis kanan, alis kiri, mata kanan, mata kiri, hidung, mulut, dan *outline* wajah. Gambar 27 merupakan *face landmark* dari wajah yang terdeteksi.



Gambar 12. Face Landmark

4.2.4 Face Descriptor

Proses selanjutnya adalah generasi *face descriptor*, yang dilakukan menggunakan model *face_recognition_model*. Model tersebut menggunakan arsitektur seperti ResNet-34 yang diimplementasikan untuk mengkomputasi *face descriptor* (sebuah *feature vector* dengan 128 nilai) dari sebuah gambar wajah, yang digunakan untuk mendeskripsikan karakteristik wajah seseorang. Fungsi yang digunakan untuk melakukan kalkulasi *face descriptor* adalah `withFaceDescriptor()` yang akan menghasilkan array berisi nilai-nilai bertipe *float32* yang digunakan untuk mendeskripsikan wajah. Perhitungan Manual generasi descriptor dapat dilihat pada Lampiran 2.

1. Input Citra

Melalui tahap ekstraksi fitur 68 titik *landmark* menggunakan model *face_landmark_68_model*. Kemudian citra diubah menjadi ukuran 224x224x3 atau dengan channel RGB.

```
[[ (150,160,170), (150,160,170), (150,160,170)..(224)],  
 [ (150,160,170), (150,160,170), (150,160,170)..(224)],  
 [ (150,160,170), (150,160,170), (150,160,170)..(224)], [224]]
```

2. Normalisasi Citra

a. Mengubah Skala Piksel ke Rentang [0,1]

```
[[ (150/255, 160/255, 170/255), (150/255, 160/255, 170/255), (150/255, 160/255,  
 170/255)..(224)],  
 [ (150/255, 160/255, 170/255), (150/255, 160/255, 170/255), ((150/255, 160/255,  
 170/255)..(224)],  
 [ (150/255, 160/255, 170/255), (150/255, 160/255, 170/255), (150/255, 160/255,  
 170/255)..(224)], [224]]
```

b. Normalisasi dengan mean dan standard deviation (Resnet34)

Mean (0.485,0.456, 0.406)

Standard Deviation (0.229, 0.224, 0.225)

```
[[ (-1.1000001, -1.0000002, -1.545043), (-1.1000001, -1.0000002, -1.545043),  
 (-1.1000001, -1.0000002, -1.545043)..(224)],  
 (-1.1000001, -1.0000002, -1.545043), (-1.1000001, -1.0000002, -1.545043),  
 (-1.1000001, -1.0000002, -1.545043)..(224)],  
 [ (-1.1000001, -1.0000002, -1.545043), (-1.1000001, -1.0000002, -1.545043),  
 (-1.1000001, -1.0000002, -1.545043)..(224)], [224]]
```

3. Konvolusi Awal

Input: Citra berukuran 224x224x3

a. Convolutional Layer

Ukuran filter (F) = 7 x 7, Channel input C_{in} =3, Jumlah filter = 64, Stride S = 2, Padding P = 3

Output: 112x112x64

b. Max Pooling

Filter size F = 3 x 3, Stride S = 2

Output: 56x56x64

4. Residual Block

a. Tahap 1

Input: 56x56x64

Convolutional Block

Filter size F=3 x 3, Channel input C_{in} =64, Channel output C_{out} =64, Stride S=1

Output: $56 \times 56 \times 64$

b. Tahap 2

Input: $56 \times 56 \times 64$

Convolutional Block

Filter size $F=3 \times 3$, Channel input $C_{in}=64$, Channel output $C_{out}=128$, Stride $S=2$

Output: $28 \times 28 \times 128$

c. Tahap 3

Input: $28 \times 28 \times 128$

Convolutional Block

Filter size $F=3 \times 3$, Channel input $C_{in}=128$, Channel output $C_{out}=256$, Stride $S=2$

Output: $14 \times 14 \times 256$

d. Tahap 4

Input: $14 \times 14 \times 256$

Convolutional Block

Filter size $F=3 \times 3$, Channel input $C_{in}=256$, Channel output $C_{out}=512$, Stride $S=2$

Output: $7 \times 7 \times 512$

5. Average Pooling

Output : $1 \times 1 \times 512 = 512$

6. Fully Connected Layer

$F_c = 512 \times 128$

7. Softmax

Output : 128 nilai

4.2.5 Pencocokan Wajah

Pada penelitian ini pencocokan wajah dilakukan dengan mengaplikasikan algoritma *euclidean distance* terhadap dua nilai *descriptor* dari *descriptor* yang disimpan pada *database* dan *descriptor* yang dihasilkan *webcam* secara *realtime*. Berikut merupakan perhitungan pencocokan wajah dengan membandingkan *descriptor* yang disimpan pada *database* dan yang dihasilkan *webcam* secara *realtime*. *Descriptor* 1 dan 2 dapat dilihat pada tabel 9. Tabel *descriptor* 1 dan 2 lainnya dapat dilihat pada Lampiran 3.

Tabel 9. Tabel *Descriptor*

Descriptor 1	Descriptor 2
-0.08952424675226212	-0.1387988030910492
0.040943440049886703	0.06213417649269104
0.02458115667104721	0.1461406648159027
-0.01436180341988802	0.033213257789611816
-0.08662403374910355	-0.12445299327373505

Kemudian, ambil selisih *descriptor* pertama dan kedua. Selisih *descriptor* 1 dan 2 dapat dilihat pada tabel 10. Tabel selisih *descriptor* 1 dan 2 lainnya dapat dilihat pada Lampiran 3.

Tabel 10. Selisih *Descriptor*

Descriptor 1	Descriptor 2	Selisih
-0.08952424675226212	-0.1387988030910492	0.04927455634
0.040943440049886703	0.06213417649269104	-0.02119073644
0.02458115667104721	0.1461406648159027	-0.1215595081

-0.01436180341988802	0.033213257789611816	-0.04757506121
-0.08662403374910355	-0.12445299327373505	0.03782895952

Lalu, kuadratkan setiap selisih. Kuadrat Selisih *descriptor* 1 dan 2 dapat dilihat pada tabel 11. Tabel kuadrat selisih *descriptor* 1 dan 2 lainnya dapat dilihat pada Lampiran 3.

Tabel 11. Selisih Kuadrat

Kuadrat Selisih	Hasil
0.04927455634^2	0.002427981903
-0.02119073644^2	0.00044812863
-0.1215595081^2	0.01477671401
-0.04757506121^2	0.002263386449
0.03782895952^2	0.001431030178

Kemudian, jumlahkan semua kuadratnya. Penjumlahan kuadrat dapat dilihat pada tabel 12. Tabel kuadrat penjumlahan kuadrat lainnya dapat dilihat pada Lampiran 3.

Tabel 12. Jumlah Kuadrat

Jumlah Kuadrat
0.002427981903
0.00044812863
0.01477671401
0.002263386449
0.001431030178

Tabel 13. Hasil Perhitungan *Descriptor*

Akar kuadrat	Hasil
0.4990866697	0.7

Pada penelitian ini, penggunaan model *face recognition* dihitung berdasarkan *descriptor* yang mana pada penelitian ini mengambil 10 *descriptor* untuk mendapatkan akurasi yang baik. Lalu, 1 *descriptor* berjumlah 128 nilai. Maka, pada contoh diatas hasil perhitungan memiliki nilai yang kurang dari ari ambang batas yang diatur yaitu kurang dari atau sama dengan 0.8. Jadi *descriptor* 1 dan *descriptor* 2 akan dianggap cocok karena Jarak Euclidean-nya (sekitar 0.7) lebih kecil dari ambang batas 0.8.

4.3 *Systems Development Life Cycle* (SDLC)

4.3.1 Tahap Perencanaan

Tahap ini merupakan tahap awal yang ditujukan untuk mengetahui apa saja yang dibutuhkan untuk perancangan Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*, dalam mengumpulkan data-data yang akan digunakan pada proses perencanaan ini maka dilakukan konsultasi secara terjadwal dengan salah satu asprak untuk meminta data, saran dan masukan terhadap sistem yang akan dikembangkan. Semua data tersebut dikumpulkan dan diolah sebagai sebuah perencanaan sistem yang akan dikembangkan.

4.3.2 Tahap Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk menentukan kebutuhan fungsional dan non-fungsional *user* atau pengguna. Berikut ini adalah kebutuhan fungsional dan non-fungsional dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*.

a. Fungsional

Adapun kebutuhan fungsional antara lain:

1. Mahasiswa dapat melakukan *register* untuk mendaftarkan diri.
2. Mahasiswa dapat melakukan presensi menggunakan fitur *face recognition* dan *location based service*.

b. Non-fungsional

Adapun kebutuhan non-fungsional antara lain:

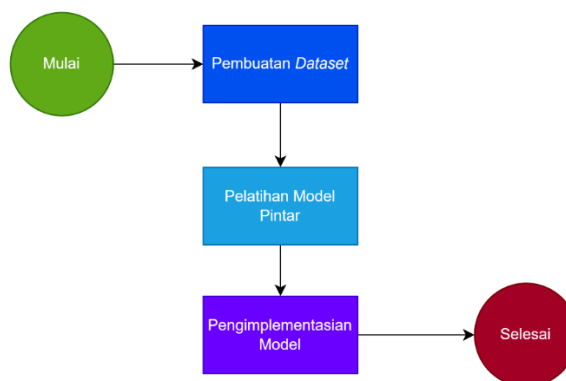
1. Sistem dapat memastikan data yang ditampilkan sesuai dengan data yang berada pada *database*.
2. Data latih berjumlah sekitar 1037 data gambar wajah yang mana 27 gambar diambil secara langsung dan 973 gambar diambil dari data sekunder.

4.3.3 Tahap Perancangan

Tahap perancangan dimulai dengan perancangan model deteksi wajah. Kemudian, ditambahkan dengan mendesain *use case diagram*, *activity diagram*, *entity relationship diagram*, hingga *mockup*.

a. Perancangan Model

Perancangan model akan menggunakan *Machine Learning Life Cycle* yaitu sebuah workflow atau alur kerja pengembangan model ML dimulai dari proses pengumpulan data sampai model tersebut siap digunakan. *Machine learning lifecycle* bersifat *forward* (maju), dan dapat berulang (iteratif) karena setiap perulangan yang dilakukan bertujuan untuk terus meningkatkan keakuratan dan performa model. Tahap-tahap dalam Pembuatan Model Deteksi Wajah dapat dilihat pada Gambar 13.



Gambar 13. Alur Pembuatan Model

1. Pembuatan Dataset

Pembuatan dataset merupakan tahapan yang penting pada model pengembangan *machine learning life cycle*. Dataset yang dibuat merupakan dataset yang cocok dengan format yang sesuai dengan model *tensorflow SSDMobileNetV1*. Dalam proses pembuatan *dataset* terdapat beberapa tahapan yang perlu dilakukan. Gambar 14 merupakan tahapan pembuatan dataset pada penelitian ini yang diproses pada *platform roboflow*:



Gambar 14. Alur Pembuatan Dataset

a) Mengumpulkan Data

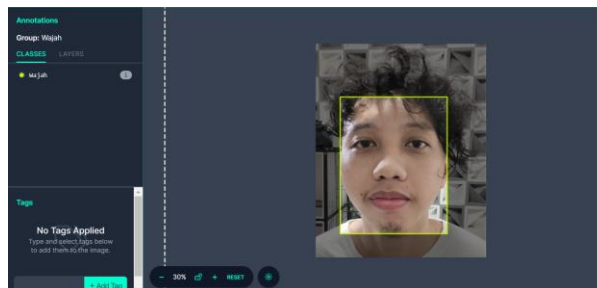
Tahapan pertama yang dilakukan yaitu mengumpulkan data, Terdapat 1037 gambar yang dikumpulkan untuk untuk proses pelatihan yang nantinya akan menghasilkan model yang akan digunakan pada penelitian ini. Dalam proses pengumpulan data yang dikumpulkan sejumlah 27 gambar yang diambil secara langsung dan 973 berasal dari data sekunder atau *dataset* publik. Proses mengumpulkan data dapat dilihat pada Gambar 15.



Gambar 15. Kumpulan Data

b) Melakukan *Pre-processing*

Tahapan kedua yaitu melakukan *pre-processing*. Pada tahapan ini dilakukan proses anotasi yaitu pemberian boks pembatas atau *bounding box* dan label sehingga model *SSDMobileNetV1* dapat mempelajari setiap *frame* yang telah dikumpulkan. Contoh gambar yang sudah melalui proses anotasi terdapat pada Gambar 16.

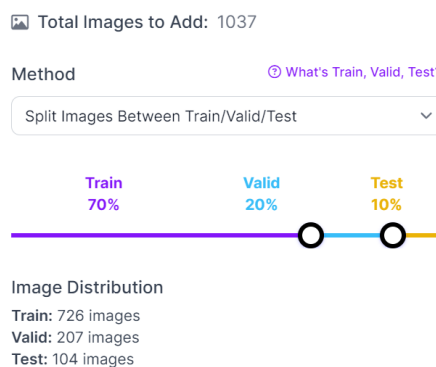


Gambar 16. Anotasi Data

Hasil dari anotasi data yang dilakukan akan menghasilkan berkas anotasi berekstensi .csv yang isinya adalah koordinat titik X dan Y label yang telah disematkan pada gambar. Setelah melalui anotasi frame, maka gambar melalui proses resize atau perubahan ukuran menjadi 640 x 640 piksel.

c) Membagikan *Dataset*

Tahapan ketiga yaitu pembagian dataset atau *splitting dataset*. Kurang lebih 1037 data dibagi menjadi 3 bagian yaitu data *train* atau data pelatihan, data *valid* atau data validasi, dan data *test* atau data tes. Hal ini dilakukan agar *dataset* terbagi sesuai dengan 3 tahapan pada alur penelitian selanjutnya yaitu pelatihan model pintar. Pada tahap ini *dataset* dibagi menjadi 726 data *training*, 207 data *valid* dan 104 data *test*. Gambar 17 merupakan proses pembagian data.



Gambar 17. Pemabagian Data

d) Melakukan *Augmentasi* Data

Setelah membagikan *dataset*, maka dilakukan *augmentasi* data untuk meningkatkan akurasi dan kualitas model akhir. Proses *augmentasi* data dilakukan pada 726 data *train* berupa penambahan 726 gambar dengan efek *Flip*, dan penambahan 726 gambar dengan efek *Grayscale* sehingga total data train sejumlah 2178 gambar. Contoh *frame* yang telah melalui proses *augmentasi* digambarkan pada Gambar 18.



Gambar 18. Augmentasi Data

Maka setelah proses *augmentasi* frame selesai lah proses pembuatan dataset sehingga total frame video pada dataset bertambah menjadi 2489 gambar dengan pembagian 2178 gambar menjadi data *train*, 207 data *valid*, dan 104 data *test*. Dataset yang telah dibuat pada platform *Roboflow* akan dapat digunakan pada pelatihan.

b. *Use Case Diagram*

Use case diagram merupakan sebuah diagram yang digunakan untuk merepresentasikan skenario apa saja yang dapat dilakukan oleh suatu sistem dan aktor/user yang mana saja yang dapat melakukan suatu skenario. Pada sistem ini melibatkan 3 (tiga) aktor pengguna antara lain Mahasiswa, Asprak, dan Admin. *Use case diagram* dapat dilihat pada Lampiran 4.

c. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan bagaimana alur berjalannya fitur dalam suatu aplikasi yang dapat dilakukan oleh pengguna dan pemrosesan apa yang dijalankan oleh sistem secara sederhana. Berikut ini merupakan *activity diagram* dari fitur pada Sistem Presensi Mahasiswa Dengan *Face Recognition Dan Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*. *Activity diagram* dapat dilihat pada Lampiran 5.

d. *Entity Relationship Diagram*

Desain database aplikasi digambarkan dalam bentuk ERD (*Entity Relationship Diagram*) dengan menunjukkan masing-masing entitas tabel beserta atribut - atributnya dan hubungannya dengan entitas lain. *Entity Relationship Diagram* dapat dilihat pada Lampiran 6.

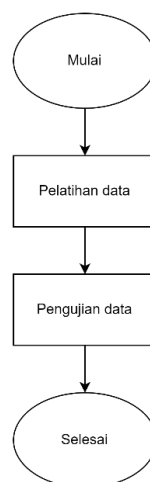
e. *Mockup*

Mockup merupakan desain kasar dari *user interface* aplikasi yang di dalamnya mengutamakan tampilan layout serta pembagian area warna. *Mockup* dirancang terlebih dahulu agar *developer* dapat memahami seperti apa tampilan *front end* yang akan dibuat namun dapat dengan mudah diperbaiki apabila terdapat bagian -bagian yang perlu direvisi berdasarkan analisis atau hasil diskusi. Desain *mockup* dapat dilihat pada Lampiran 7.

4.3.4 Tahap Implementasi

1. Pelatihan Model Pintar

Proses *training* atau proses latih adalah proses dimana *Convolutional Neural Network* akan dilatih untuk mengenali dan mempelajari suatu pola untuk dapat melakukan deteksi wajah dan melakukan pengenalan sesuai kelas yang diharapkan dengan tingkat akurasi yang tinggi. Gambar 19 merupakan alur dari pelatihan model pintar.



Gambar 19. Alur Pelatihan Model

a) Melakukan Pelatihan Data

Dalam tahapan pertama yaitu pelatihan data, diawali dengan melakukan instalasi dan import paket dari *tensorflow* atau pengembang dari model *SSDMobileNetV1*, pembuatan direktori memuat *dataset* yang telah dibuat sebelumnya serta beberapa paket Python lainnya. Ketika *dataset* telah berhasil dimuat maka proses pelatihan data dapat dilakukan. Terdapat parameter yang dibutuhkan agar proses training dapat berjalan sesuai konfigurasi yang telah dilakukan untuk membuat file yang nantinya akan menjadi sebuah file hasil akhir dari keseluruhan proses pelatihan yang mampu melakukan pengenalan wajah. Untuk menampilkan hasil pelatihan melalui *tensorboard* untuk melihat dan menganalisis nilai *loss*. Pelatihan dilakukan dengan pengulangan sebanyak 5000 step. Gambar 20 merupakan proses konfigurasi pelatihan.

```
train_config: {
  fine_tune_checkpoint_version: V2
  fine_tune_checkpoint: "ssd_mobilenet_v1_fpn_640x640_coco17_tpu-8/checkpoint/ckpt-0"
  fine_tune_checkpoint_type: "detection"
  batch_size: 4
  sync_replicas: true
  startup_delay_steps: 0
  replicas_to_aggregate: 8
  num_steps: 5000
  data_augmentation_options {
    random_horizontal_flip {
  }
}
```

Gambar 20. Konfigurasi Pelatihan

Gambar 21 merupakan kode untuk proses validasi, model akan memproses data valid dengan data train. Data validasi digunakan untuk mengevaluasi kinerja model selama pelatihan. Model tidak diperbarui dengan data valid, tetapi akurasi diukur berdasarkan data ini untuk memastikan bahwa model tidak overfitting dengan data train.

```
177 train_input_reader: {
178   label_map_path: "labelmap.pbtxt"
179   tf_record_input_reader {
180     input_path: "train.record"
181   }
182 } <- #177-182 train_input_reader:
183
184 eval_input_reader: {
185   label_map_path: "labelmap.pbtxt"
186   shuffle: false
187   num_epochs: 1
188   tf_record_input_reader {
189     input_path: "test.record"
190   }
191 }
```

Gambar 21. Kode Validasi

Setelah semua konfigurasi selesai, maka tahap pelatihan dapat dilakukan untuk membuat file yang nantinya akan menjadi sebuah *weight* atau file hasil akhir dari keseluruhan proses pelatihan yang mampu melakukan deteksi wajah. Gambar 22 merupakan proses pelatihan model.

```
INFO:tensorflow:Step 100 per-step time 1.256s
I0731 07:17:15.971048 91360 model_lib_v2.py:705] Step 100 per-step time 1.256s
INFO:tensorflow: {'Loss/classification_loss': 0.16954808,
'Loss/localization_loss': 0.15693633,
'Loss/regularization_loss': 0.77461547,
'Loss/total_loss': 1.1010998,
'learning_rate': 0.014666351}
```

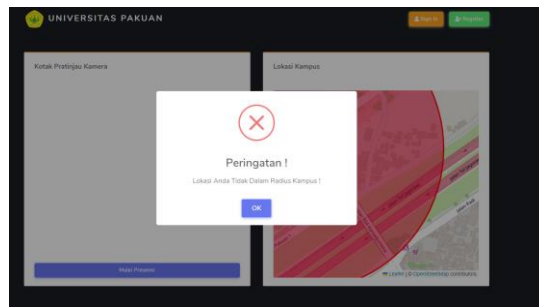
Gambar 22. Proses Pelatihan Model

BAB V HASIL DAN PEMBAHASAN

5.1 Hasil

5.1.1 *Location Based Service*

Halaman ini merupakan tampilan dari implementasi *location based service* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang mana jika jarak perangkat pengguna tidak dalam radius kampus yang sudah diatur sekitar 200 meter maka sistem akan memunculkan peringatan dan menonaktifkan *webcam* atau kamera. Gambar 23 merupakan hasil implementasi lbs pada *web*.

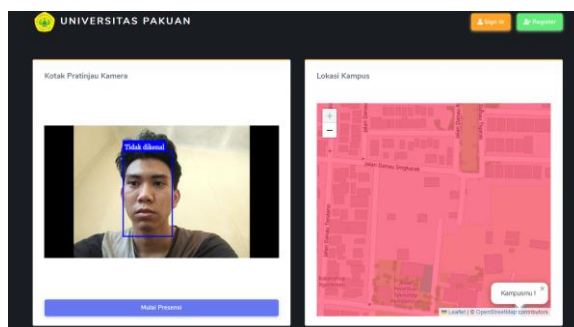


Gambar 23. Hasil Implementasi LBS

5.1.2 *Face Recognition*

1. Mahasiswa Tidak Terdaftar

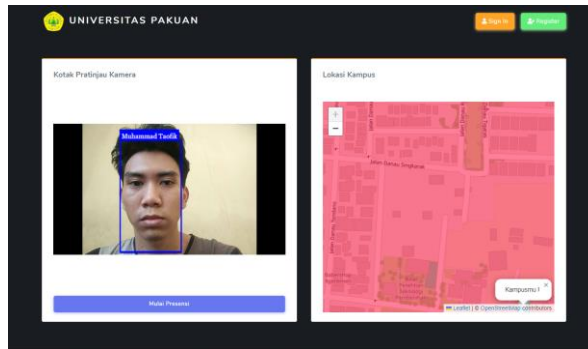
Halaman ini merupakan tampilan dari implementasi *face recognition* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang mana jarak perangkat pengguna dalam radius kampus yang sudah diatur sekitar 200 meter tetapi mahasiswa tidak terdaftar dalam *database* maka sistem akan memunculkan *bounding box* dengan label tidak dikenal dan tidak dapat melakukan presensi. Gambar 24 merupakan hasil jika mahasiswa tidak terdaftar.



Gambar 24. Mahasiswa Tidak Terdaftar

2. Mahasiswa Terdaftar

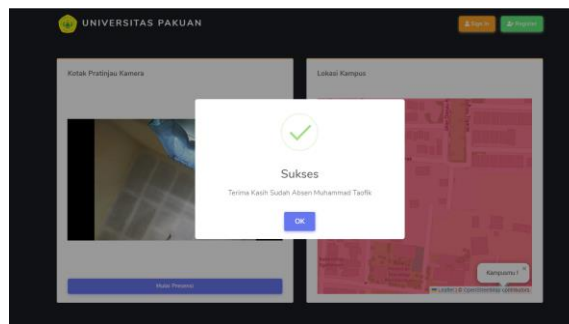
Halaman ini merupakan tampilan dari implementasi *face recognition* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang mana jarak perangkat pengguna dalam radius kampus yang sudah diatur sekitar 200 meter dan mahasiswa terdaftar dalam *database* maka sistem akan memunculkan *bounding box* dengan label nama mahasiswa dan dapat melakukan presensi. Gambar 25 merupakan hasil jika mahasiswa terdaftar.



Gambar 25. Mahasiswa terdaftar

3. Mahasiswa Berhasil Melakukan Presensi

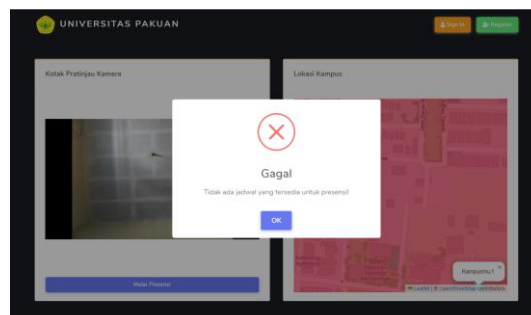
Halaman ini merupakan tampilan dari implementasi *face recognition* dari Sistem Presensi Mahasiswa Dengan Face Recognition Dan Location Based Service Berbasis Web yang mana jarak perangkat pengguna dalam radius kampus yang sudah diatur sekitar 200 meter, mahasiswa telah terdaftar dalam database dan memiliki jadwal maka sistem akan memunculkan peringatan presensi berhasil. Gambar 26 merupakan hasil jika presensi berhasil.



Gambar 26. Presensi Berhasil

4. Mahasiswa Tidak Memiliki Jadwal

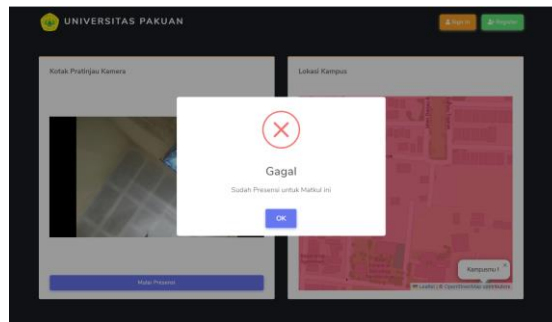
Halaman ini merupakan tampilan dari implementasi *face recognition* dari Sistem Presensi Mahasiswa Dengan Face Recognition Dan Location Based Service Berbasis Web yang mana jarak perangkat pengguna dalam radius kampus yang sudah diatur sekitar 200 meter dan mahasiswa telah terdaftar dalam database tetapi tidak memiliki jadwal pada hari itu maka sistem akan memunculkan peringatan presensi gagal. Gambar 27 merupakan hasil jika mahasiswa tidak memiliki jadwal.



Gambar 27. Tidak Memiliki Jadwal

5. Mahasiswa Sudah Melakukan Presensi

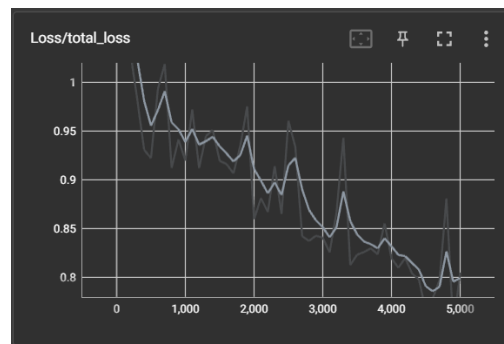
Halaman ini merupakan tampilan dari implementasi *face recognition* dari Sistem Presensi Mahasiswa Dengan Face Recognition Dan Location Based Service Berbasis Web yang mana jarak perangkat pengguna dalam radius kampus yang sudah diatur sekitar 200 meter dan mahasiswa telah terdaftar dalam database tetapi sudah melakukan presensi pada matkul itu maka sistem akan memunculkan peringatan presensi gagal. Gambar 28 merupakan hasil jika mahasiswa sudah melakukan presensi pada matkul yang sama.



Gambar 28. Sudah Melakukan Presensi

5.1.3 Pelatihan Model

Hasil pelatihan selama 5000 step menunjukkan perbandingan data *train* dan *valid* memiliki nilai *loss* yang cenderung menurun dengan nilai sekitar 0.8, dapat diputuskan bahwa model *pre-trained SSDMobileNetV1* yang telah dilatih memiliki hasil yang cukup baik. Gambar 29 merupakan *total_loss* dari pelatihan model.



Gambar 29. Hasil Pelatihan Model

5.2 Uji Coba

Setelah tahap implementasi sistem selesai, langkah berikutnya adalah melakukan pengujian terhadap sistem. Pengujian ini bertujuan untuk memastikan bahwa sistem yang telah dibangun sesuai dengan kebutuhan yang telah ditentukan, serta untuk mengurangi kemungkinan kesalahan yang mungkin terjadi dalam sistem tersebut. Hal ini penting agar sistem dapat beroperasi dengan baik.

5.2.1 Prosedur Pengujian

Tahapan pengujian akan dilakukan dengan 2 tahapan pengujian, yaitu pengujian *black box testing* dan kasus orang kembar. *Black box testing* sendiri adalah tahapan testing yang dilakukan untuk mengetahui apakah sistem sudah berjalan dengan baik sesuai dengan fungsionalitasnya, sedangkan kasus orang kembar digunakan untuk menguji performa model dari *library face-api.js*.

1. Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem dilakukan dengan metode *black box testing* untuk memeriksa apakah setiap fitur yang terdapat pada aplikasi *face recognition* berbasis web dapat berfungsi sesuai dengan yang diharapkan. Tabel prosedur pengujian dapat dilihat pada Lampiran 9.

2. Pengujian Kasus Orang Kembar

Pengujian kasus orang kembar dilakukan untuk mengetahui seberapa mampu model mengenali dan membedakan berdasarkan fitur wajah orang kembar, meskipun memiliki kemiripan fitur wajah yang sangat tinggi.

5.2.2 Hasil Pengujian

Berikut ini merupakan hasil dari dua pengujian yang telah dilakukan sesuai dengan prosedur yang ditetapkan

1. Hasil Pengujian Fungsionalitas Sistem

Berikut ini adalah daftar skenario *test case* uji coba yang telah dilaksanakan beserta hasilnya. Hasil Pengujian Fungsionalitas lainnya dapat dilihat pada Lampiran 10.

2. Hasil Pengujian Kasus Orang Kembar

a. Akusisi Data

Terdapat beberapa gambar dari orang kembar yang mana gambar tersebut akan menjadi gambar referensi dan gambar lainnya akan digunakan sebagai validasi. Dilakukan pengambilan beberapa pose wajah untuk meningkatkan akurasi dari fitur wajah. Gambar 30 dan 31 merupakan gambar wajah orang kembar.



Gambar 30. Gambar Referensi (Mawar)



Gambar 31. Gambar Validasi (Melati)

b. Deteksi Wajah

Selanjutnya dilakukan pendeteksian wajah dari referensi gambar menggunakan model *SSDMobileNetV1* yang akan mengkonversi gambar menjadi *array matrix* dengan nilai 0-250. Gambar 32 merupakan proses deteksi wajah referensi.



Gambar 32. Deteksi Wajah Referensi

c. *Feature Extraction*

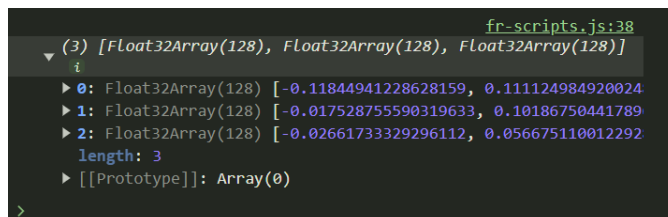
Dari hasil deteksi wajah pada langkah sebelumnya, kemudian dilakukan proses kalkulasi ekstraksi fitur 68 titik *landmark* wajah menggunakan model *face_landmark_68_model* yang akan menghasilkan array 68 titik koordinat x dan y dari masing-masing landmark wajah yang meliputi alis kanan, alis kiri, mata kanan, mata kiri, hidung, mulut, dan *outline* wajah serta mengubah ukuran citra menjadi 224 piksel. Gambar 33 merupakan landmark wajah referensi.



Gambar 33. Landmark Wajah Referensi

d. *Face Descriptor*

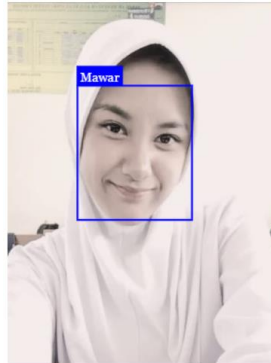
Proses selanjutnya adalah generasi *face descriptor*, yang dilakukan menggunakan model *face_recognition_model* yang menggunakan arsitektur ResNet-34 yang dari sebuah gambar wajah, yang akan menghasilkan array berisi nilai-nilai bertipe *float32* yang digunakan untuk mendeskripsikan wajah. Gambar 34 merupakan descriptor wajah referensi.



Gambar 34. Descriptor Wajah Referensi

e. Pencocokan Wajah

Setelah semua tahap dilakukan, proses selanjutnya yaitu mencocokkan gambar wajah orang kembar ini menggunakan algoritma *euclidean distance*. Gambar 35 merupakan hasil pengenalan wajah.



Gambar 35. Hasil Pengenalan Wajah

5.2.3 Evaluasi Pengujian

Berikut ini merupakan evaluasi dari masing-masing hasil pengujian yang telah dilakukan.

1. Evaluasi Pengujian Fungsionalitas Sistem

Berdasarkan hasil pengujian black box yang tertera pada masing-masing tabel skenario test case, dapat dinyatakan bahwa semua fitur pada Sistem Presensi Mahasiswa Dengan *Face Recognition Dan Location Based Service* Berbasis *Web* telah mampu menjalankan fungsinya dengan baik saat digunakan. Dengan kata lain, *control flow* atau alur kode pemrograman juga dapat dikatakan telah disusun dengan cukup baik sehingga dapat mencapai fungsi yang diharapkan. Oleh karena itu, sesuai dengan rumus presentase keberhasilan untuk *black box testing*, secara fungsional aplikasi dinyatakan memiliki presentase keberhasilan sebesar 100%.

$$\text{Presentase keberhasilan} = \frac{\text{Jumlah item uji berhasil}}{\text{Jumlah seluruh item uji}} \times 100\% \quad (6)$$

$$\text{Presentase Keberhasilan} = \frac{56}{56} \times 100\%$$

$$\text{Presentase Keberhasilan} = 100\%$$

2. Evaluasi Pengujian Kasus Orang Kembar

Berdasarkan hasil pengujian, dapat dinyatakan model yang digunakan menunjukkan kinerja yang tidak optimal pada kasus orang kembar. Meskipun model dirancang untuk mengenali wajah dengan akurasi tinggi, tetapi dalam kasus orang kembar yang memiliki fitur wajah yang sangat mirip model gagal dalam membedakan antara keduanya. Oleh karena itu, perlu dilakukan pengembangan lebih spesifik untuk meningkatkan kemampuan model tersebut.

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance* berhasil dirancang, dikembangkan dan semua fiturnya berfungsi dengan cukup baik. Hal ini dapat dilihat dengan sistem yang telah beroperasi dengan baik dan berhasil melewati 56 skenario pengujian *blackbox* dengan tingkat keberhasilan mencapai 100%. Terkhusus, bagian fitur utama sudah dapat digunakan dengan pengambilan 10 data *descriptor* pada setiap mahasiswa untuk melakukan pengenalan wajah dan memiliki akurasi pengenalan yang termasuk tinggi. Ini menunjukkan bahwa penerapan algoritma *euclidean distance* untuk menghitung jarak kedekatan antar *descriptor* wajah cocok digunakan dalam melakukan pengenalan wajah. Namun, perlu dicatat bahwa model yang digunakan mengalami kegagalan dalam pengujian kasus orang kembar sehingga perlu dilakukan pengembangan lebih spesifik untuk meningkatkan kemampuan model tersebut. Tetapi, dapat disimpulkan juga bahwa algoritma *convolutional neural network* dan metode penggunaan *descriptor* yang digunakan oleh *library face-api.js* dapat menghasilkan model *face recognition* yang akurat dan dapat berperan dengan baik pada Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*.

6.2 Saran

Adapun beberapa saran yang dapat diberikan terkait dengan hasil penelitian ini, antara lain:

1. Pengembangan Lebih Lanjut

Pengembangan sistem lebih lanjut juga disarankan seperti menambahkan fitur untuk penetapan radius secara dinamis atau beberapa fitur dan fungsionalitas baru yang dapat lebih mempermudah pengguna dalam melakukan kegiatan presensi. Selain itu, untuk menjaga agar sistem tetap mutakhir, dapat dilakukan penerapan teknologi terbaru.

2. Sosialisasi tata cara penggunaan sistem

Untuk memaksimalkan pemanfaatan sistem ini, disarankan untuk melakukan sosialisasi tata cara penggunaan sistem kepada semua pihak asprak, mahasiswa, dan admin. Sosialisasi dapat dilakukan melalui sesi pelatihan atau workshop yang melibatkan pengguna langsung dalam mengoperasikan sistem. Dengan adanya sosialisasi yang efektif, diharapkan pengguna akan lebih familiar dan percaya diri dalam menggunakan sistem, sehingga sistem dapat berjalan dengan optimal dan memberikan manfaat maksimal bagi semua pengguna.

3. Pertimbangan Teknologi Tambahan

Karena model mengalami kegagalan dalam melakukan pengenalan dalam kasus orang kembar, maka disarankan untuk mempertimbangkan integrasi dengan teknologi tambahan seperti *fingerprint* untuk meningkatkan akurasi identifikasi.

DAFTAR PUSTAKA

- Anis, Y., Purwatiningtyas, P., Retnowati, R., & Fajrina, E. A. N. (2022). Penerapan Framework Bootstrap Dalam Sistem Informasi Rekam Medis Data Posyandu dengan Metode Waterfall. *Jurnal Sistem Komputer dan Informatika (JSON)*, 4(2), 310. <https://doi.org/10.30865/json.v4i2.4833>
- Asvin, A., Suradi, M., & Syarwani, A. (2021). Sistem Absensi Menggunakan Teknologi Qr Code Dan Face Recognition. Dalam *INFORMASI v* (Vol. 62, Nomor 1).
- Desmile, J., Orisa, M., & Santi Wahyuni, F. (2021). Aplikasi Pencarian Bengkel Sepeda Motor Menggunakan Location Based Service Pada Wilayah Kota Tebing Tinggi Berbasis Andorid. Dalam *Jurnal Mahasiswa Teknik Informatika* (Vol. 5, Nomor 1).
- Farid Fauzi, M., Wibowo, A., & Setiawan, D. (2021). Implementasi Laravel Dan White Hat Seo Pada Pembuatan Website Banghm Jogja Properti. Dalam *Information System Journal (INFOS) |* (Vol. 4, Nomor 1).
- Fauzi Isputrawan, M. (2023). *Pengembangan Aplikasi Absensi Berbasis Web Menggunakan Face Recognition* (Vol. 17, Nomor 1). <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>
- Galih, M., Putra, W., & Sonjaya, I. (2023). *Perancangan Aplikasi QRCode Sebagai Media Presensi Mahasiswa Berbasis Web*.
- Habibi, M. (2023). *Aplikasi Pengenalan Tindakan Manusia Otomatis Dengan Model Yolov8 Berbasis Web*.
- Habie, K. F., Maku, R., & Hasyim, W. (2022). *Sistem Presensi Perkuliahan Menggunakan Teknologi Face Recognition Berbasis Website Di Universitas Muhammadiyah Gorontalo*.
- Hajar, A., Nabawi, I., Kartikawati, L., Yudana, F. R., Budi, S., & Prasetyantara, N. (2021). *Pengolahan Data Spasial-Geolocation untuk Menghitung Jarak 2 Titik Spatial-Geolocation Data Processing to Calculate 2 Point*.
- Herman, D. A., & Wijaya, H. (2021). Perancangan dan Pengembangan Aplikasi Donasi Darah Berbasis Web dengan Metode Rapid Application Development (RAD). *Jurnal Ilmu Komputer dan Bisnis*, 12(2a), 90–103. <https://doi.org/10.47927/jikb.v12i2a.195>
- Isum, R., Maryati, S., & Tryatmojo, B. (2019). *Raden Isum Suryani Maryati Akurasi Sistem Face Recognition Akurasi Sistem Face Recognition OpenCV Menggunakan Raspberry Pi Dengan Metode Haar Cascade KATA KUNCI Akurasi Face Recognition Raspberry Pi OpenCV Haar Cascade*.
- Jaini, N., & Asri, E. (2021). Fitri Nova 48 Sistem Manajemen Kehadiran Menggunakan Metode Face Recognition Berbasis Web JITSI. Dalam *Jurnal Ilmiah Teknologi Sistem Informasi* (Vol. 2, Nomor 2). <http://jurnal-itsi.org>

- Kadek, I., Arta, J., Bagus, N., & Nugraha, S. (2020). *Implementasi Aplikasi User Management Hotspot Mikrotik Berbasis Php Dengan Application Programming Interface (Api) Dan Framework Bootstrap*. 66. <https://bit.ly/jurnalresistor>
- Maulida Shaliha, R. (2023). *Rancang Bangun Aplikasi Face Recognition Berbasis Web dengan face-api.js dan Algoritma Euclidean Distance Sebagai Pendukung Sistem Smart Door*.
- Nagara, B. S., Oetari, D., Apriliani, Z., & Sutabri, T. (2023). Penerapan Metode Sdlc (System Development Life Cycle) Waterfall Pada Perancangan Aplikasi Belanja Online Berbasis Android Pada Cv Widi Agro Application Of The Waterfall Sdlc (System Development Life Cycle) Method In Designing Android-Based Online Shopping Applications On Cv Widi Agro. *Journal of Information Technology and Computer Science (INTECOMS)*, 6(2).
- Nilfaidah, N., Sa'ban Miru, A., & Lamada, M. (2021). *Pengembangan Sistem Absensi Mahasiswa Realtime Menggunakan Php, Mysql, Sms Gateway, Dan Framework Codeigniter*.
- Redhia Pratama, F. (2021). *Rancang Bangun Ojek Online Syari Berdasarkan Jenis Kelamin Dengan Fitur Deteksi Masker TensorFlow Lite*.
- Rianto Sitanggang. (2022). *Rancang Bangun Sistem Penjualan Tanaman Hias Berbasis Web Menggunakan Php Dan Mysql*.
- Santoso, B., & Kristianto, R. P. (2020). *Implementasi Penggunaan Opencv Pada Face Recognition Untuk Sistem Presensi Perkuliahan Mahasiswa*.
- Satwikayana, S., Wibowo, S. A., & Vendyansyah, N. (2021). Sistem Presensi Mahasiswa Otomatis Pada Zoom Meeting Menggunakan Face Recognition Dengan Metode Convolutional Neural Network Berbasis Web. Dalam *Jurnal Mahasiswa Teknik Informatika* (Vol. 5, Nomor 2).
- Siregar, D. R. S., Koryanto, L., & Faizah, N. (2023). Aplikasi Pencarian Hotel di Kota Jakarta Berbasis Android dengan Metode Location Based Service (LBS) Menggunakan Android Studio. *Computer Journal*, 1(1), 64–72. <https://doi.org/10.58477/cj.v1i1.65>
- Syakti, F., & Oktaviani, N. (2020). *JUSIM (Jurnal Sistem Informasi Musirawas) Pengembangan Aplikasi Location Based Service Fasilitas Kesehatan Menggunakan Model Scrum*.
- Tsani Abdurahman, U., Nursanto, J., & Tinggi Teknologi Muhammadiyah Cileungsi, S. (2022). Rancang Bangun Aplikasi Kehadiran Siswa Menggunakan Pengenalan Wajah Berbasis Android Dengan Metode Machine Learning. *JSAI: Journal Scientific and Applied Informatics*, 5(3). <https://doi.org/10.36085>
- Wahyu Setiya Putra, Y., & Fadlil Adhim, M. (2022). *Sistem Informasi Presensi Online Menggunakan Teknologi Face Recognition dan GPS*. 16(1).
- Widayati, Y. T., Prihati, Y., Widjaja, S., Prakoso, S. A., & Notobudojo, A. R. (2021). Implementasi Twitter Bootstrap dalam Pengembangan Aplikasi Web E-Commerce (Studi Kasus Toko Putra Reban Kendal). *TRANSFORMATIKA*, 19(1), 26–37.

LAMPIRAN

Lampiran 1. Perhitungan Manual LBS

Perhitungan dilakukan untuk mencari jarak antara koordinat perangkat dan koordinat kampus menggunakan metode *equirectangular approximation*.

Koordinat Perangkat (k1)		Koordinat Kampus (k2)		Batas Radius
Latitude [0]	Longitude [1]	Latitude [0]	Longitude [1]	
-6.648272229626856	106.82668406899835	-6.599206553911968	106.81238825180455	200 m

1. Konversi Nilai Koordinat

Lakukan konversi pada koordinat k1-k2 dari bilangan *degree* ke bilangan radian. Berikut merupakan proses perubahannya.

$$k1_{radian} = \left[-6.648272229626856 \times \frac{\pi}{180}, 106.82668406899835 \times \frac{\pi}{180} \right]$$

[-0.11603423998, 1.864477366] → bilangan radian k1

$$k2_{radian} = \left[-6.599206553911968 \times \frac{\pi}{180}, 106.81238825180455 \times \frac{\pi}{180} \right]$$

[-0.11517788238, 1.8642278569] → bilangan radian k2

Koordinat Perangkat (k1)		Koordinat Kampus (k2)	
Latitude [0]	Longitude [1]	Latitude [0]	Longitude [1]
-0.11603423998	1.864477366	-0.11517788238	1.8642278569

2. Tentukan Nilai x

Setelah dilakukan konversi, tahap selanjutnya adalah mencari nilai x (posisi horizontal pada peta) dari k1-k2. Berikut merupakan proses perhitungannya.

$$x = (k2[1] - k1[1]) \times \cos\left(\frac{k1[0] + k2[0]}{2}\right)$$

$$k2[1] - k1[1] = 1.8642278569 - 1.864477366 = -0.0002495091$$

$$k1[0] + k2[0] = -0.11603423998 + (-0.11517788238) = -0.23121212236$$

$$\frac{k1[0] + k2[0]}{2} = \frac{-0.23121212236}{2} = -0.11560606118$$

$$x = (-0.0002495091) \times \cos(-0.11560606118)$$

$$\cos(-0.11560606118) \approx 0.9933119495$$

$$x \approx (-0.0002495091) \times 0.9933119495$$

$$-0.00024794 \rightarrow \text{nilai } x$$

Nilai x	- 0.00024794
---------	--------------

3. Tentukan Nilai y

Tahap selanjutnya adalah mencari nilai y (posisi vertikal pada peta) dari k1-k2. Berikut merupakan proses perhitungannya.

$$y = k2[0] - k1[0]$$

$$y = -0.11517788238 - (-0.11603423998)$$

$$y = -0.11517788238 + 0.11603423998$$

$$0.0008563576 \rightarrow \text{nilai } y$$

Nilai y	0.0008563576
---------	--------------

4. Tentukan Nilai R

Nilai R merupakan nilai mutlak pada metode *equirectangular approximation* dengan nilai sebagai berikut.

$R = 6371.009 \times 1000$
 $6371009 \rightarrow$ nilai R

Nilai R	6371009 m
---------	-----------

5. Tentukan Nilai d

Setelah semua nilai didapatkan, tahap terakhir adalah mengukur jaraknya dengan mencari nilai d (jarak antara kedua posisi) dari k1-k2 menggunakan algoritma *euclidean* untuk menghitung jarak linear antara dua titik. Berikut merupakan proses perhitungannya.

$$d = R \times \sqrt{x^2 + y^2}$$

Keterangan:

d = jarak antara kedua posisi

R = radius bumi

x = posisi horizontal pada peta

y = posisi vertikal pada peta

$$x = -0.00024794$$

$$y = 0.0008563576$$

$$x^2 = (-0.00024794)^2$$

$$x^2 = 0.00000006225009666 = 6.225009666 \times 10^{-8}$$

$$y^2 = (0.0008563576)^2$$

$$y^2 = 0.00000073358159576 = 7.3358159576 \times 10^{-7}$$

$$\sqrt{x^2 + y^2} = \sqrt{0.00000006225009666 + 7.3358159576 \times 10^{-7}}$$

$$\sqrt{x^2 + y^2} = \sqrt{7.9583169246 \times 10^{-7}}$$

$$\sqrt{x^2 + y^2} \approx 0.00089162$$

$$d(k1, k2) = 6371009 \times 0.00089162$$

$$5679.51 \rightarrow$$
 Nilai d

Nilai d	5679.51 m
---------	-----------

Jadi, total jarak perangkat pengguna ke Universitas Pakuan adalah 5679 meter. Berdasarkan batas radius yang ditentukan sekitar 200 meter maka dapat disimpulkan bahwa perangkat pengguna berada diluar radius kampus.

Lampiran 2. Perhitungan Manual Generasi *Descriptor*

1. Input Citra

Melalui tahap ekstraksi fitur 68 titik *landmark* menggunakan model *face landmark 68 model*. Kemudian citra diubah menjadi ukuran 224x224x3 atau dengan channel RGB.

[[(150,160,170), (150,160,170), (150,160,170),],
[(150,160,170), (150,160,170), (150,160,170),],
[(150,160,170), (150,160,170), (150,160,170),]]

2. Normalisasi Citra

a. Mengubah Skala Piksel ke Rentang [0,1]

[[(1/255, 2/255, 3/255), (1/255, 2/255, 3/255), (1/255, 2/255, 3/255)],
[(1/255, 2/255, 3/255), (1/255, 2/255, 3/255), (1/255, 2/255, 3/255)],
[(1/255, 2/255, 3/255), (1/255, 2/255, 3/255), (1/255, 2/255, 3/255)]]

Hasilnya

[[(0.0039, 0.0078, 0.0118), (0.0039, 0.0078, 0.0118), (0.0039, 0.0078, 0.0118)],
[(0.0039, 0.0078, 0.0118), (0.0039, 0.0078, 0.0118), (0.0039, 0.0078, 0.0118)],
[(0.0039, 0.0078, 0.0118), (0.0039, 0.0078, 0.0118), (0.0039, 0.0078, 0.0118)]]

b. Normalisasi dengan mean dan standard deviation

Mean (0.485, 0.456, 0.406)

Standard Deviation (0.229, 0.224, 0.225)

Rumus normalisasi :

$$\text{Normalisasi} = \frac{\text{input} - \text{mean}}{\text{std}}$$

Normalisasi Channel 1 (R)

Nilai Input : 0.0039

Mean : 0.485

Standard Deviation : 0.229

$$\text{Normalisasi} = \frac{0.0039 - 0.485}{0.229} \approx -2.1000001$$

Normalisasi Channel 2 (R)

Nilai Input : 0.0078

Mean : 0.456

Standard Deviation : 0.224

$$\text{Normalisasi} = \frac{0.0078 - 0.456}{0.224} \approx -2.0000002$$

Normalisasi Channel 3 (R)

Nilai Input : 0.0118

Mean : 0.406

Standard Deviation : 0.225

$$Normalisasi = \frac{0.0118-0.406}{0.225} \approx -1.545043$$

Hasilnya

[(-2.1000001, -2.0000002, -1545043), (-2.1000001, -2.0000002, -1545043),
 (-2.1000001, -2.0000002, -1545043)],
 (-2.1000001, -2.0000002, -1545043), (-2.1000001, -2.0000002, -1545043),
 (-2.1000001, -2.0000002, -1545043)],
 [(-2.1000001, -2.0000002, -1545043), (-2.1000001, -2.0000002, -1545043),
 (-2.1000001, -2.0000002, -1545043)]]

Contoh Perhitungan Konvolusi

1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6

1	1	1
1	1	1
1	1	1

1*1	2*1	3*1	4*1	5*1	6*1
1*1	2*1	3*1	4*1	5*1	6*1
1*1	2*1	3*1	4*1	5*1	6*1
1*1	2*1	3*1	4*1	5*1	6*1
1*1	2*1	3*1	4*1	5*1	6*1
1*1	2*1	3*1	4*1	5*1	6*1

18	27	36	45
18	27	36	45
18	27	36	45
18	27	36	45

$$= 1 + 2 + 3 + 1 + 2 + 3 + 1 + 2 + 3 = 18$$

$$= 2 + 3 + 4 + 2 + 3 + 4 + 2 + 3 + 4 = 27$$

$$= 3 + 4 + 5 + 3 + 4 + 5 + 3 + 4 + 5 = 36$$

$$= 4 + 5 + 6 + 4 + 5 + 6 + 4 + 5 + 6 = 45$$

1. Konvolusi Awal

A. Input: Citra berukuran 224×224×3

1. Convolutional Layer

Ukuran filter (F) = 7 x 7, Channel input Cin=3, Jumlah filter = 64, Stride S = 2, Padding P = 3

$$H_{out} = \left\lceil \frac{H_{in} - F + 2 \times P}{S} \right\rceil + 1 = \left\lceil \frac{224 - 7 + 2 \times 3}{2} \right\rceil + 1 = 112$$

$$W_{out} = \left\lceil \frac{W_{in} - F + 2 \times P}{S} \right\rceil + 1 = \left\lceil \frac{224 - 7 + 2 \times 3}{2} \right\rceil + 1 = 112$$

Output size: Hout×Wout×Cout

Output: 112×112×64

2. Max Pooling

Filter size $F = 3 \times 3$, Stride $S = 2$

$$H_{out} = \left\lfloor \frac{H_{in} - F + 2}{S} \right\rfloor + 1 = \left\lfloor \frac{112 - 3 + 2}{2} \right\rfloor + 1 = 56$$

$$W_{out} = \left\lfloor \frac{W_{in} - F + 2}{S} \right\rfloor + 1 = \left\lfloor \frac{112 - 3 + 2}{2} \right\rfloor + 1 = 56$$

Output size: $H_{out} \times W_{out} \times C_{out}$

Output: $56 \times 56 \times 64$

2. Residual Block

A. Input: $56 \times 56 \times 64$

1. Convolutional Block

Filter size $F = 3 \times 3$, Channel input $C_{in} = 64$, Channel output $C_{out} = 64$, Stride $S = 1$

$$H_{out} = \left\lfloor \frac{H_{in} - F}{S} \right\rfloor + 1 = \left\lfloor \frac{56 - 3 + 2}{1} \right\rfloor + 1 = 56$$

$$W_{out} = \left\lfloor \frac{W_{in} - F}{S} \right\rfloor + 1 = \left\lfloor \frac{56 - 3 + 2}{1} \right\rfloor + 1 = 56$$

Output size: $H_{out} \times W_{out} \times C_{out}$

Output: $56 \times 56 \times 64$

2. Identity Block (x3)

Output : $56 \times 56 \times 64$

B. Input: $56 \times 56 \times 64$

1. Convolutional Block

Filter size $F = 3 \times 3$, Channel input $C_{in} = 64$, Channel output $C_{out} = 128$, Stride $S = 2$

$$H_{out} = \left\lfloor \frac{H_{in} - F}{S} \right\rfloor + 1 = \left\lfloor \frac{56 - 3 + 2}{2} \right\rfloor + 1 = 28$$

$$W_{out} = \left\lfloor \frac{W_{in} - F}{S} \right\rfloor + 1 = \left\lfloor \frac{56 - 3 + 2}{2} \right\rfloor + 1 = 28$$

Output size: $H_{out} \times W_{out} \times C_{out}$

Output: $28 \times 28 \times 128$

2. Identity Block (x4)

Output : $28 \times 28 \times 128$

C. Input: $28 \times 28 \times 128$

1. Convolutional Block

Filter size $F = 3 \times 3$, Channel input $C_{in} = 128$, Channel output $C_{out} = 256$, Stride $S = 2$

$$H_{out} = \left\lfloor \frac{H_{in} - F}{S} \right\rfloor + 1 = \left\lfloor \frac{28 - 3 + 2}{2} \right\rfloor + 1 = 14$$

$$W_{out} = \left\lceil \frac{W_{in} - F}{S} \right\rceil + 1 = \left\lceil \frac{28 - 3 + 2}{2} \right\rceil + 1 = 14$$

Output size: $H_{out} \times W_{out} \times C_{out}$

Output: $14 \times 14 \times 256$

2. Identity Block (x6)

Output : $14 \times 14 \times 256$

D. Input: $14 \times 14 \times 256$

1. Convolutional Block

Filter size $F=3 \times 3$, Channel input $C_{in}=256$, Channel output $C_{out}=512$, Stride $S=2$

$$H_{out} = \left\lceil \frac{H_{in} - F}{S} \right\rceil + 1 = \left\lceil \frac{14 - 3 + 2}{2} \right\rceil + 1 = 7$$

$$W_{out} = \left\lceil \frac{W_{in} - F}{S} \right\rceil + 1 = \left\lceil \frac{14 - 3 + 2}{2} \right\rceil + 1 = 7$$

Output size: $H_{out} \times W_{out} \times C_{out}$

Output: $7 \times 7 \times 512$

2. Identity Block (x3)

Output : $7 \times 7 \times 512$

3. Average Pooling

$$Average = \left(\frac{7 - 7}{7} \right) + 1 = 1$$

Output : $1 \times 1 \times 512 = 512$

4. Fully Connected Layer

Jumlah Parameter Bobot = $512 \times 128 = 65,536$

Jumlah Parameter Bias = 128

Total Jumlah Parameter = $65,536 + 128 = 65,664$

5. Softmax

Output : 128 Nilai

Lampiran 3. Perhitungan Manual *Face Recognition*

1. Data *Descriptor* 1 dan 2

Tabel 14. Tabel *Descriptor* Lainnya

Descriptor 1	Descriptor 2
-0.011433359235525131	-0.0804155021905899
-0.00781457033008337	-0.042625341564416885
-0.1671474128961563	-0.07807295769453049
0.16017140448093414	0.09684731066226959
-0.13388550281524658	-0.062079042196273804
0.26697996258735657	0.18686026334762573
-0.018987877294421196	-0.04936249554157257
-0.18606413900852203	-0.19591942429542542
-0.12641090154647827	-0.12928320467472076
-0.02620745822787285	0.0495535172522068
0.18799780309200287	0.14935274422168732
-0.18663035333156586	-0.14555148780345917
-0.15655739605426788	-0.1257086843252182
-0.034712743014097214	-0.10217822343111038
-0.04356198012828827	-0.06827495992183685
0.06135711073875427	0.010048811323940754
-0.07233347743749619	-0.030030108988285065
0.05108283832669258	0.050006404519081116
0.03947477042675018	0.06669522076845169
-0.12364280223846436	-0.08949063718318939
-0.3408203721046448	-0.3593548536300659
-0.07856239378452301	-0.045831967145204544
-0.12012957781553268	-0.20298078656196594
0.051280856132507324	0.022970717400312424
-0.039780233055353165	-0.05316122621297836
-0.04319094121456146	-0.08438636362552643
-0.01799892634153366	0.04528135061264038
-0.25505757331848145	-0.1674506664276123
-0.07594215124845505	-0.07269756495952606
-0.016857748851180077	-0.040296223014593124
0.0874706506729126	0.06539302319288254
-0.035421255975961685	-0.001473007258027792
-0.09003175050020218	-0.054787106812000275
0.2310599386692047	0.1379285454750061
-0.06302661448717117	-0.11546909809112549
-0.2237083613872528	-0.21848323941230774
-0.00006098443554947153	-0.04833098128437996
-0.019010724499821663	0.010090219788253307
0.23508794605731964	0.14617100358009338
0.15583373606204987	0.27775099873542786,
0.02690131589770317	0.026000676676630974
-0.0006975516444072127	0.05563421919941902,

-0.0952940583229065	-0.059261854737997055
0.10031751543283463	0.08742032945156097
-0.2597452998161316	-0.20356501638889313
0.07490081340074539	0.027544952929019928
0.08314067125320435	0.10965417325496674
0.0804760605096817	0.06972502171993256
0.06005264073610306	-0.0092666856944561
-0.07491320371627808	-0.0092666856944561
-0.1504443883895874	-0.1742822527885437
0.06163880228996277	-0.013585811480879784
0.09406880289316177	0.056565653532743454
-0.21518535912036896	-0.15657497942447662
-0.01394043117761612	0.01177128218114376
0.06445684283971786	0.023214787244796753
-0.08978518843650818	-0.13540664315223694
-0.00888174306601286	-0.10950927436351776
-0.043205149471759796	-0.02434687316417694
0.2799185812473297	0.24552305042743683
0.1334376335144043	0.12776713073253632,
-0.1300152689218521	-0.09784175455570221
-0.10187054425477982	-0.13626860082149506
0.08124195039272308	0.1695721298456192,
-0.14335109293460846	-0.2100876122713089
0.025691788643598557	-0.02152998559176922
0.05066097155213356	0.07054762542247772
-0.13880768418312073	-0.1348874568939209
-0.24160729348659515	-0.12681743502616882
-0.3375192880630493	-0.2845343053340912
0.0567564032971859	-0.02136077731847763,
0.5082237720489502	0.4458670914173126
0.16116105020046234	0.06573069095611572
-0.18346349895000458	-0.24670062959194183
-0.022469215095043182	0.008075546473264694
-0.08321793377399445	-0.17229682207107544,
0.04210919141769409	-0.008536270819604397
0.19829250872135162	0.041745029389858246
0.13459967076778412	0.12628482282161713
0.018245773389935493	-0.040857184678316116
0.029645878821611404	0.005661054980009794
-0.09872818738222122	-0.14621217548847198
0.0729668065905571	-0.012615199200809002
0.17108488082885742	0.16600066423416138
-0.08949573338031769	-0.058314695954322815
-0.10190406441688538	0.038059838116168976
0.2384365200996399	0.2112186998128891
-0.01636447198688984	0.03671376034617424
0.11143171787261963	0.03339969366788864

-0.004533296916633844	0.04816330596804619
0.0794738382101059	-0.07213112711906433
-0.012409805320203304	-0.05954684317111969
0.07529408484697342	-0.03096969984471798
-0.08836494386196136	-0.11245004832744598
0.07034030556678772	0.010644514113664627
0.05014505609869957	0.045668359845876694
0.05431609973311424	-0.05845452845096588
0.06136614456772804	-0.026372624561190605
0.10380249470472336	0.04494161158800125
-0.11939563602209091	-0.1725679636001587
0.12446966022253036,	0.13317479193210602
-0.0012754974886775017	0.009310363791882992
0.029413726180791855	0.05169792100787163
0.043301668018102646	0.032306522130966187
-0.04187493771314621	-0.0708945021033287
-0.09188570082187653	-0.03946959972381592
-0.10020024329423904	-0.07299841195344925
0.0997365340590477	0.1995743364095688
-0.26754114031791687	-0.14588937163352966
0.20131899416446686	0.10606615990400314
0.21313174068927765,	0.16739599406719208
0.0020279129967093468	0.039624691009521484
0.1977376639842987	0.18364116549491882
0.0761423259973526	0.06402163952589035
0.05219700559973717	0.12287275493144989
-0.002980406628921628	-0.05834941565990448
-0.05538604408502579	-0.06227650120854378
-0.2543727457523346	-0.20955859124660492
0.04124259203672409	0.030750999227166176
0.10659737139940262	0.07805449515581131
-0.023118676617741585	-0.05045618489384651
0.14041996002197266	0.03893769532442093
0.016318773850798607	0.059323329478502274

2. Selisih *Descriptor* 1 dan 2

Tabel 15. Tabel Selisih *Descriptor* Lainnya

Descriptor 1	Descriptor 2	Selisih
-0.011433359235525131	-0.0804155021905899	0.06898214296
-0.00781457033008337	-0.042625341564416885	0.03481077123
-0.1671474128961563	-0.07807295769453049	-0.0890744552
0.16017140448093414	0.09684731066226959	0.06332409382
-0.13388550281524658	-0.062079042196273804	-0.07180646062
0.26697996258735657	0.18686026334762573	0.08011969924
-0.018987877294421196	-0.04936249554157257	0.03037461825
-0.18606413900852203	-0.19591942429542542	0.009855285287

-0.12641090154647827	-0.12928320467472076	0.002872303128
-0.02620745822787285	0.0495535172522068	-0.07576097548
0.18799780309200287	0.14935274422168732	0.03864505887
-0.18663035333156586	-0.14555148780345917	-0.04107886553
-0.15655739605426788	-0.1257086843252182	-0.03084871173
-0.034712743014097214	-0.10217822343111038	0.06746548042
-0.04356198012828827	-0.06827495992183685	0.02471297979
0.06135711073875427	0.010048811323940754	0.05130829941
-0.07233347743749619	-0.030030108988285065	-0.04230336845
0.05108283832669258	0.050006404519081116	0.001076433808
0.03947477042675018	0.06669522076845169	-0.02722045034
-0.12364280223846436	-0.08949063718318939	-0.03415216506
-0.3408203721046448	-0.3593548536300659	0.01853448153
-0.07856239378452301	-0.045831967145204544	-0.03273042664
-0.12012957781553268	-0.20298078656196594	0.08285120875
0.051280856132507324	0.022970717400312424	0.02831013873
-0.039780233055353165	-0.05316122621297836	0.01338099316
-0.04319094121456146	-0.08438636362552643	0.04119542241
-0.01799892634153366	0.04528135061264038	-0.06328027695
-0.25505757331848145	-0.1674506664276123	-0.08760690689
-0.07594215124845505	-0.07269756495952606	-0.003244586289
-0.016857748851180077	-0.040296223014593124	0.02343847416
0.0874706506729126	0.06539302319288254	0.02207762748
-0.035421255975961685	-0.001473007258027792	-0.03394824872
-0.09003175050020218	-0.054787106812000275	-0.03524464369
0.2310599386692047	0.1379285454750061	0.09313139319
-0.06302661448717117	-0.11546909809112549	0.0524424836
-0.2237083613872528	-0.21848323941230774	-0.005225121975
-0.00006098443554947153	-0.04833098128437996	0.04826999685
-0.019010724499821663	0.010090219788253307	-0.02910094429
0.23508794605731964	0.14617100358009338	0.08891694248
0.15583373606204987	0.27775099873542786	-0.1219172627
0.02690131589770317	0.026000676676630974	0.000900639221
-0.0006975516444072127	0.05563421919941902	-0.05633177084
-0.0952940583229065	-0.059261854737997055	-0.03603220358
0.10031751543283463	0.08742032945156097	0.01289718598
-0.2597452998161316	-0.20356501638889313	-0.05618028343
0.07490081340074539	0.027544952929019928	0.04735586047
0.08314067125320435	0.10965417325496674	-0.026513502
0.0804760605096817	0.06972502171993256	0.01075103879
0.06005264073610306	-0.0092666856944561	0.06931932643
-0.07491320371627808	-0.0092666856944561	-0.06564651802
-0.1504443883895874	-0.1742822527885437	0.0238378644
0.06163880228996277	-0.013585811480879784	0.07522461377
0.09406880289316177	0.056565653532743454	0.03750314936
-0.21518535912036896	-0.15657497942447662	-0.0586103797
-0.01394043117761612	0.01177128218114376	-0.02571171336

0.06445684283971786	0.023214787244796753	0.04124205559
-0.08978518843650818	-0.13540664315223694	0.04562145472
-0.00888174306601286	-0.10950927436351776	0.1006275313
-0.043205149471759796	-0.02434687316417694	-0.01885827631
0.2799185812473297	0.24552305042743683	0.03439553082
0.1334376335144043	0.12776713073253632	0.005670502782
-0.1300152689218521	-0.09784175455570221	-0.03217351437
-0.10187054425477982	-0.13626860082149506	0.03439805657
0.08124195039272308	0.1695721298456192	-0.08833017945
-0.14335109293460846	-0.2100876122713089	0.06673651934
0.025691788643598557	-0.02152998559176922	0.004161803052
0.05066097155213356	0.07054762542247772	-0.01988665387
-0.13880768418312073	-0.1348874568939209	-0.003920227289
-0.24160729348659515	-0.12681743502616882	-0.1147898585
-0.3375192880630493	-0.2845343053340912	-0.05298498273
0.0567564032971859	-0.02136077731847763	0.07811718062
0.5082237720489502	0.4458670914173126	0.06235668063
0.16116105020046234	0.06573069095611572	0.09543035924
-0.18346349895000458	-0.24670062959194183	0.06323713064
-0.022469215095043182	0.008075546473264694	-0.03054476157
-0.08321793377399445	-0.17229682207107544	0.0890788883
0.04210919141769409	-0.008536270819604397	0.05064546224
0.19829250872135162	0.041745029389858246	0.1565474793
0.13459967076778412	0.12628482282161713	0.008314847946
0.018245773389935493	-0.040857184678316116	0.05910295807
0.029645878821611404	0.005661054980009794	0.02398482384
-0.09872818738222122	-0.14621217548847198	0.04748398811
0.0729668065905571	-0.012615199200809002	0.08558200579
0.17108488082885742	0.16600066423416138	0.005084216595
-0.08949573338031769	-0.058314695954322815	-0.03118103743
-0.10190406441688538	0.038059838116168976	-0.1399639025
0.2384365200996399	0.2112186998128891	0.02721782029
-0.01636447198688984	0.03671376034617424	-0.05307823233
0.11143171787261963	0.03339969366788864	0.0780320242
-0.004533296916633844	0.04816330596804619	-0.05269660288
0.0794738382101059	-0.07213112711906433	0.1516049653
-0.012409805320203304	-0.05954684317111969	0.04713703785
0.07529408484697342	-0.03096969984471798	0.1062637847
-0.08836494386196136	-0.11245004832744598	0.02408510447
0.07034030556678772	0.010644514113664627	0.05969579145
0.05014505609869957	0.045668359845876694	0.004476696253
0.05431609973311424	-0.05845452845096588	0.1127706282
0.06136614456772804	-0.026372624561190605	0.08773876913
0.10380249470472336	0.04494161158800125	0.05886088312
-0.11939563602209091	-0.1725679636001587	0.05317232758
0.12446966022253036	0.13317479193210602	-0.00870513171
-0.0012754974886775017	0.009310363791882992	-0.01058586128

0.029413726180791855	0.05169792100787163	-0.02228419483
0.043301668018102646	0.032306522130966187	0.01099514589
-0.04187493771314621	-0.0708945021033287	0.02901956439
-0.09188570082187653	-0.03946959972381592	-0.0524161011
-0.10020024329423904	-0.07299841195344925	-0.02720183134
0.0997365340590477	0.1995743364095688	-0.09983780235
-0.26754114031791687	-0.14588937163352966	-0.1216517687
0.20131899416446686	0.10606615990400314	0.09525283426
0.21313174068927765	0.16739599406719208	0.04573574662
0.0020279129967093468	0.039624691009521484	-0.03759677801
0.1977376639842987	0.18364116549491882	0.01409649849
0.0761423259973526	0.06402163952589035	0.01212068647
0.05219700559973717	0.12287275493144989	-0.07067574933
-0.002980406628921628	-0.05834941565990448	0.05536900903
-0.05538604408502579	-0.06227650120854378	0.006890457124
-0.2543727457523346	-0.20955859124660492	-0.04481415451
0.04124259203672409	0.030750999227166176	0.01049159281
0.10659737139940262	0.07805449515581131	0.02854287624
-0.023118676617741585	-0.05045618489384651	0.02733750828
0.14041996002197266	0.03893769532442093	0.1014822647
0.016318773850798607	0.059323329478502274	-0.04300455563

3. Kuadrat Selisih *Descriptor*

Tabel 16. Tabel Kuadrat Selsih Lainnya

Kuadrat Selisih	Hasil
0.06898214296 ²	0.004758536047
0.03481077123 ²	0.001211789794
-0.0890744552 ²	0.007934258569
0.06332409382 ²	0.004009940858
-0.07180646062 ²	0.005156167787
0.08011969924 ²	0.006419166206
0.03037461825 ²	0.000922617434
0.009855285287 ²	0.0000971266481
0.002872303128 ²	0.00000825012526
-0.07576097548 ²	0.005739725406
0.03864505887 ²	0.001493440575
-0.04107886553 ²	0.001687473193
-0.03084871173 ²	0.000951643015
0.06746548042 ²	0.004551591048
0.02471297979 ²	0.00061073137
0.05130829941 ²	0.002632541588
-0.04230336845 ²	0.001789574982
0.001076433808 ²	0.00000115870974
-0.02722045034 ²	0.000740952917
-0.03415216506 ²	0.001166370378
0.01853448153 ²	0.000343527006

-0.03273042664 ²	0.001071280828
0.08285120875 ²	0.006864322791
0.02831013873 ²	0.000801463955
0.01338099316 ²	0.000179050978
0.04119542241 ²	0.001697062828
-0.06328027695 ²	0.004004393451
-0.08760690689 ²	0.007674970135
-0.003244586289 ²	0.0000105273402
0.02343847416 ²	0.000549362071
0.02207762748 ²	0.000487421635
-0.03394824872 ²	0.001152483591
-0.03524464369 ²	0.001242184909
0.09313139319 ²	0.008673456398
0.0524424836 ²	0.002750214086
-0.005225121975 ²	0.0000273018997
0.04826999685 ²	0.002329992596
-0.02910094429 ²	0.000846864959
0.08891694248 ²	0.00790622266
-0.1219172627 ²	0.01486381894
0.000900639221 ²	0.000000811151006
-0.05633177084 ²	0.003173268406
-0.03603220358 ²	0.001298319695
0.01289718598 ²	0.000166337406
-0.05618028343 ²	0.003156224246
0.04735586047 ²	0.002242577521
-0.026513502 ²	0.000702965788
0.01075103879 ²	0.000115584835
0.06931932643 ²	0.004805169017
-0.06564651802 ²	0.004309465328
0.0238378644 ²	0.000568243779
0.07522461377 ²	0.005658742517
0.03750314936 ²	0.001406486212
-0.0586103797 ²	0.003435176609
-0.02571171336 ²	0.000661092204
0.04124205559 ²	0.001700907149
0.04562145472 ²	0.002081317131
0.1006275313 ²	0.01012590006
-0.01885827631 ²	0.000355634585
0.03439553082 ²	0.00118305254
0.005670502782 ²	0.0000321546018
-0.03217351437 ²	0.001035135027
0.03439805657 ²	0.001183226296
-0.08833017945 ²	0.007802220602
0.06673651934 ²	0.004453763014
0.004161803052 ²	0.0000173206046
-0.01988665387 ²	0.000395479002
-0.003920227289 ²	0.000015368182

-0.1147898585 ²	0.01317671161
-0.05298498273 ²	0.002807408395
0.07811718062 ²	0.006102293908
0.06235668063 ²	0.003888355619
0.09543035924 ²	0.009106953465
0.06323713064 ²	0.003998934692
-0.03054476157 ²	0.000932982459
0.0890788883 ²	0.007935048341
0.05064546224 ²	0.002564962846
0.1565474793 ²	0.02450711328
0.008314847946 ²	0.0000691366964
0.05910295807 ²	0.003493159653
0.02398482384 ²	0.000575271775
0.04748398811 ²	0.002254729127
0.08558200579 ²	0.007324279715
0.005084216595 ²	0.0000258492584
-0.03118103743 ²	0.000972257095
-0.1399639025 ²	0.019589894
0.02721782029 ²	0.000740809741
-0.05307823233 ²	0.002817298747
0.0780320242 ²	0.006088996801
-0.05269660288 ²	0.002776931955
0.1516049653 ²	0.0229840655
0.04713703785 ²	0.002221900337
0.1062637847 ²	0.01129199194
0.02408510447 ²	0.000580092257
0.05969579145 ²	0.003563587517
0.004476696253 ²	0.0000200408093
0.1127706282 ²	0.01271721458
0.08773876913 ²	0.007698091608
0.05886088312 ²	0.003464603562
0.05317232758 ²	0.00282729642
-0.00870513171 ²	0.0000757793181
-0.01058586128 ²	0.000112060459
-0.02228419483 ²	0.000496585339
0.01099514589 ²	0.000120893233
0.02901956439 ²	0.000842135117
-0.0524161011 ²	0.002747447655
-0.02720183134 ²	0.000739939628
-0.09983780235 ²	0.009967586778
-0.1216517687 ²	0.01479915283
0.09525283426 ²	0.009073102435
0.04573574662 ²	0.002091758519
-0.03759677801 ²	0.001413517717
0.01409649849 ²	0.00019871127
0.01212068647 ²	0.000146911041
-0.07067574933 ²	0.004995061543

0.05536900903 ²	0.003065727161
0.006890457124 ²	0.0000474783994
-0.04481415451 ²	0.002008308444
0.01049159281 ²	0.00011007352
0.02854287624 ²	0.000814695784
0.02733750828 ²	0.000747339359
0.1014822647 ²	0.01029865005
-0.04300455563 ²	0.001849391805

4. Penjumlahan Hasil Kuadrat

Tabel 17. Jumlah Kuadrat Lainnya

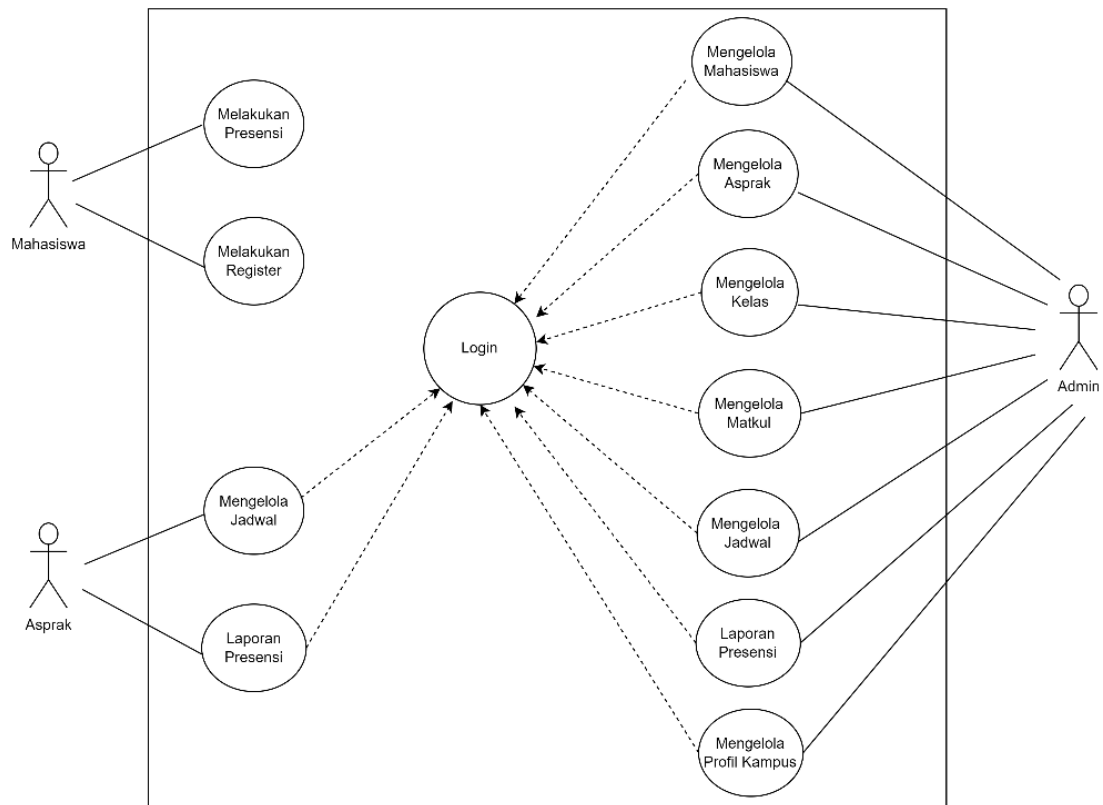
Penjumlahan Kuadrat
0.004758536047
0.001211789794
0.007934258569
0.004009940858
0.005156167787
0.04441793422
0.006419166206
0.000922617434
0.0000971266481
0.00000825012526
0.005739725406
0.001493440575
0.001687473193
0.000951643015
0.004551591048
0.00061073137
0.002632541588
0.001789574982
0.00000115870974
0.000740952917
0.001166370378
0.000343527006
0.001071280828
0.006864322791
0.000801463955
0.000179050978
0.001697062828
0.004004393451
0.007674970135
0.0000105273402
0.000549362071
0.000487421635
0.001152483591
0.001242184909

0.008673456398
0.002750214086
0.0000273018997
0.002329992596
0.000846864959
0.00790622266
0.01486381894
0.000000811151006
0.003173268406
0.001298319695
0.000166337406
0.003156224246
0.002242577521
0.000702965788
0.000115584835
0.004805169017
0.004309465328
0.000568243779
0.005658742517
0.001406486212
0.003435176609
0.000661092204
0.001700907149
0.002081317131
0.01012590006
0.000355634585
0.00118305254
0.0000321546018
0.001035135027
0.001183226296
0.007802220602
0.004453763014
0.0000173206046
0.000395479002
0.000015368182
0.01317671161
0.002807408395
0.006102293908
0.003888355619
0.009106953465
0.003998934692
0.000932982459
0.007935048341
0.002564962846
0.02450711328
0.0000691366964
0.003493159653

0.000575271775
0.002254729127
0.007324279715
0.0000258492584
0.000972257095
0.019589894
0.000740809741
0.002817298747
0.006088996801
0.002776931955
0.0229840655
0.002221900337
0.01129199194
0.000580092257
0.003563587517
0.0000200408093
0.01271721458
0.007698091608
0.003464603562
0.00282729642
0.0000757793181
0.000112060459
0.000496585339
0.000120893233
0.000842135117
0.002747447655
0.000739939628
0.009967586778
0.01479915283
0.009073102435
0.002091758519
0.001413517717
0.00019871127
0.000146911041
0.004995061543
0.003065727161
0.0000474783994
0.002008308444
0.00011007352
0.000814695784
0.000747339359
0.01029865005
0.001849391805

Lampiran 4. Use Case Diagram

1. Mahasiswa adalah aktor yang dapat berinteraksi dengan sistem dengan menggunakan fitur-fitur seperti melakukan presensi dengan *face recognition* serta dapat melakukan *register* untuk mendaftarkan diri.
2. Asprak adalah aktor yang dapat melihat jadwal dan melihat presensi sesuai jadwal dia mengajar. Selain itu, asprak juga dapat membuat laporan presensi dari jadwal dia mengajar.
3. Admin adalah aktor yang dapat mengelola banyak hal seperti mengelola mahasiswa, asprak, kelas, matkul, jadwal, profil kampus serta dapat membuat laporan presensi.

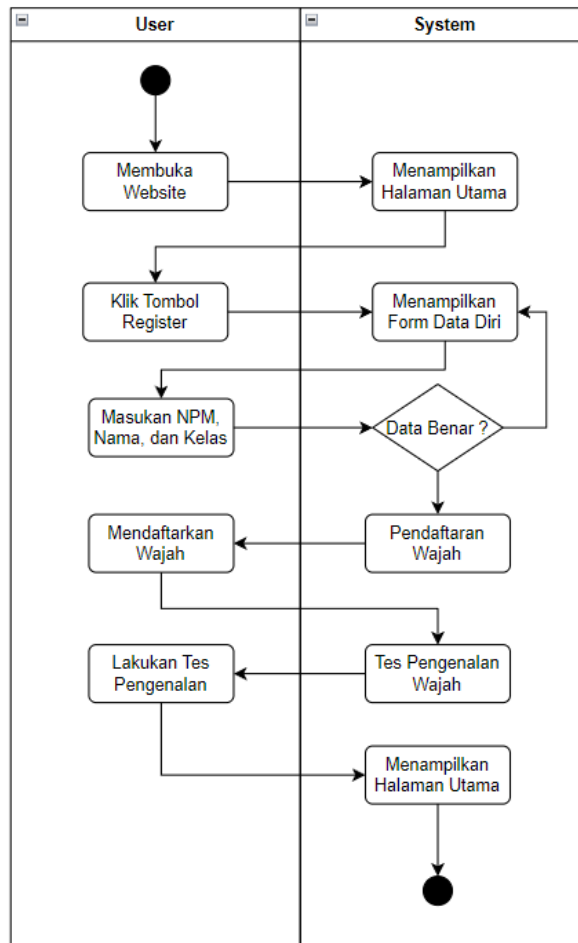


Gambar 36. Use Case Diagram

Lampiran 5. Activity Diagram

1. Activity Register (Mahasiswa)

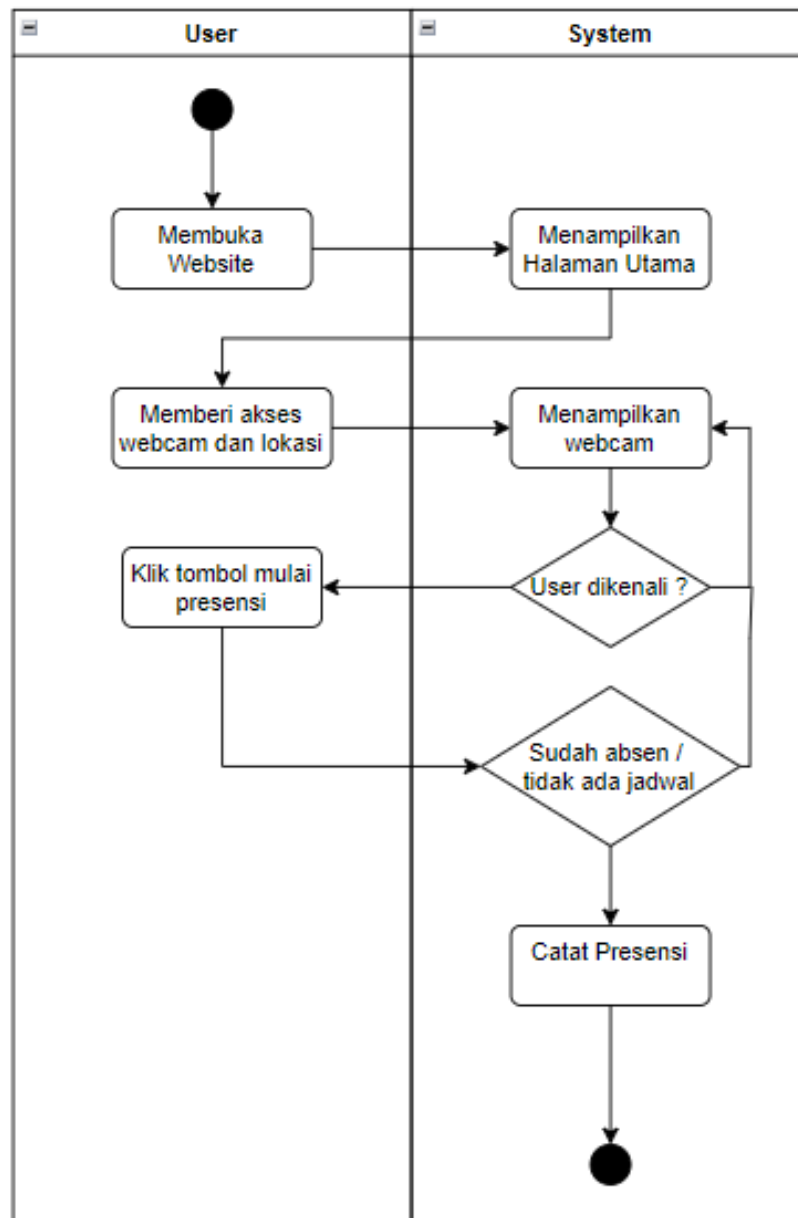
Pertama, mahasiswa membuka *website* dan menekan tombol register. Kemudian, *website* akan menampilkan *form* untuk memasukan data diri jika data yang dimasukan sudah benar sesuai validasi maka *website* akan menampilkan halaman pendaftaran wajah. Lalu, setelah mahasiswa mendaftarkan wajah maka *website* akan menampilkan halaman tes pengenalan wajah jika mahasiswa sudah melakukan tes pengenalan wajah maka mahasiswa akan diarahkan kembali ke halaman utama. Gambar 37 merupakan *activity diagram* dari proses *register*.



Gambar 37. Activity Register

2. Activity Presensi (Mahasiswa)

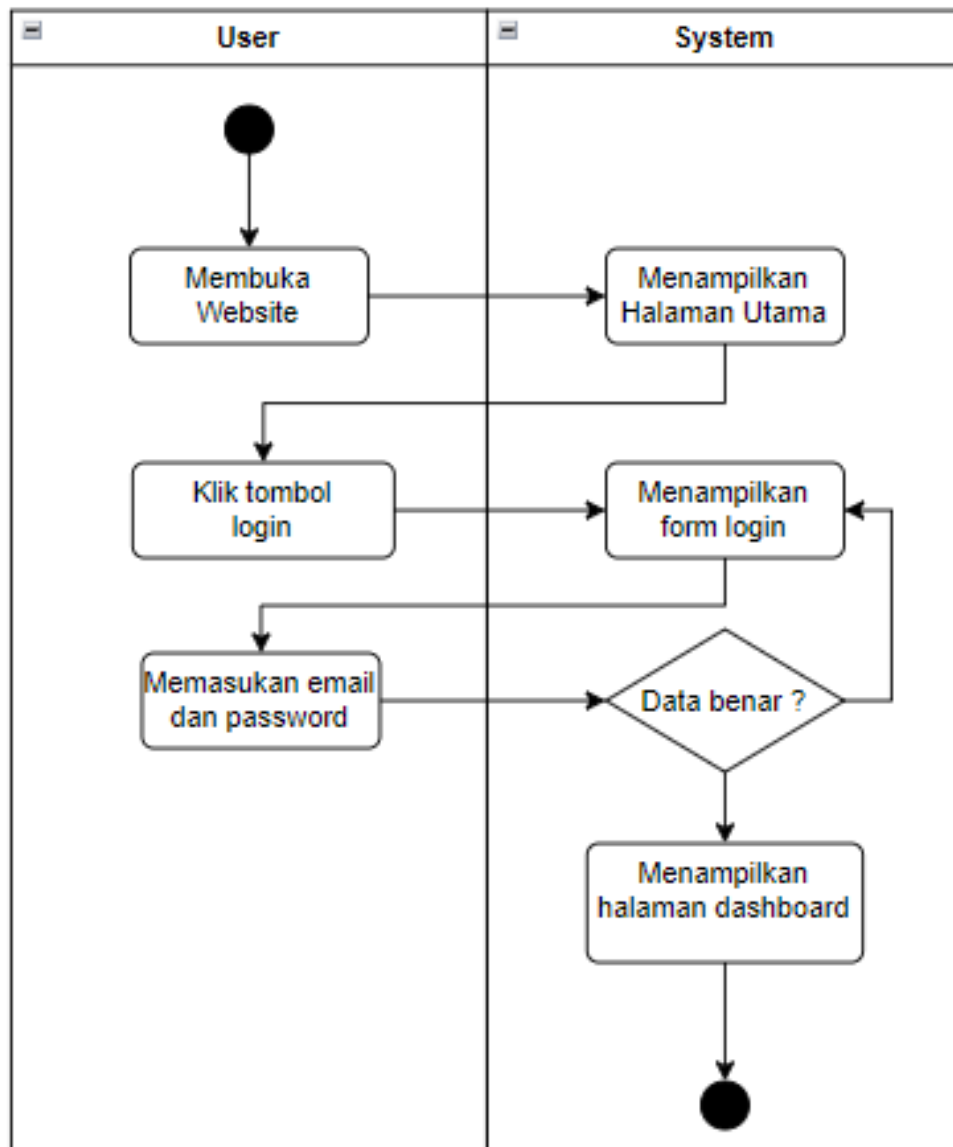
Pertama mahasiswa membuka *website* dan memberikan akses webcam dan lokasi. Kemudian, *website* akan menampilkan webcam dan akan melakukan proses pengenalan wajah. Lalu, ketika sudah dikenali mahasiswa dapat melakukan presensi dengan menekan tombol mulai presensi jika mahasiswa sudah melakukan presensi sebelumnya atau mahasiswa tidak memiliki jadwal pada hari itu maka presensi tidak akan dicatat sebaliknya jika validasi benar maka presensi akan dicatat. Gambar 38 merupakan *activity diagram* dari proses presensi.



Gambar 38. Activity Presensi

3. Activity Login (Admin / Asprak)

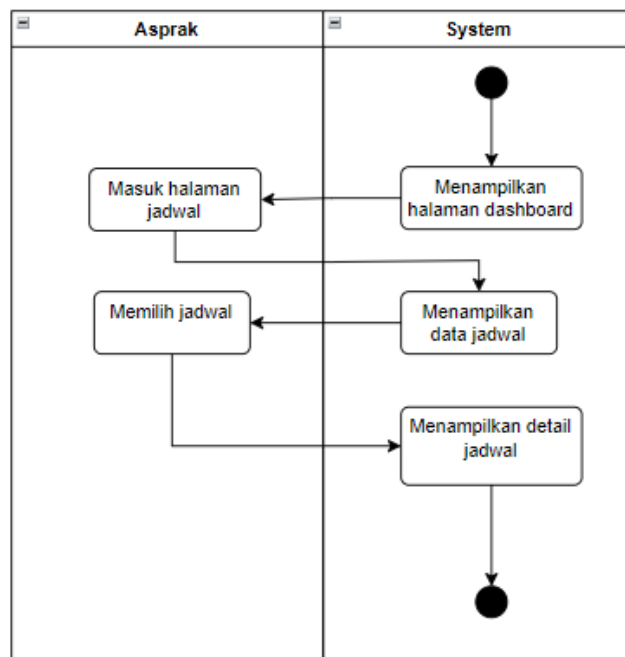
Aktivitas *login* dapat dilakukan oleh admin dan asprak dengan memasuki halaman *login* aplikasi terlebih dahulu. Lalu, admin atau asprak harus mengisi *form* email dan *password*. Ketika admin atau asprak sudah memasukan email dan *passwordnya* maka sistem akan melakukan validasi dan pengecekan, jika data yang dimasukan benar maka halaman akan dialihkan ke halaman *dashboard*. Jika gagal akan ditampilkan pesan *error*. Gambar 39 merupakan activity dari proses *login*.



Gambar 39. *Activity Login*

4. *Activity* Lihat Jadwal (Asprak)

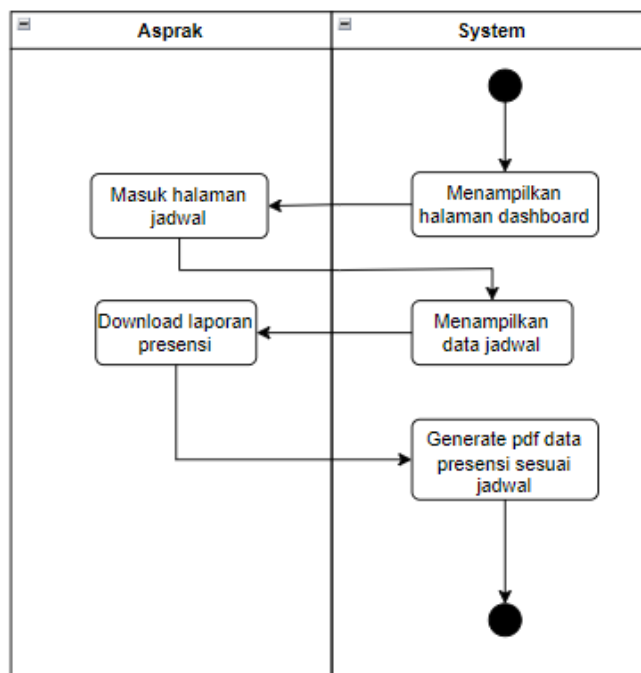
Asprak dapat melihat jadwal dia mengajar dengan cara pergi ke halaman jadwal yang berada pada *sidebar*, sistem akan melakukan pengambilan data sesuai dengan kode asprak dan mencari jadwal yang sesuai dan menampilkannya dengan bentuk list pada halaman *website*. Gambar 40 merupakan *activity* diagram dari jadwal asprak.



Gambar 40. Activity Jadwal Asprak

5. Activity Laporan Presensi (Asprak)

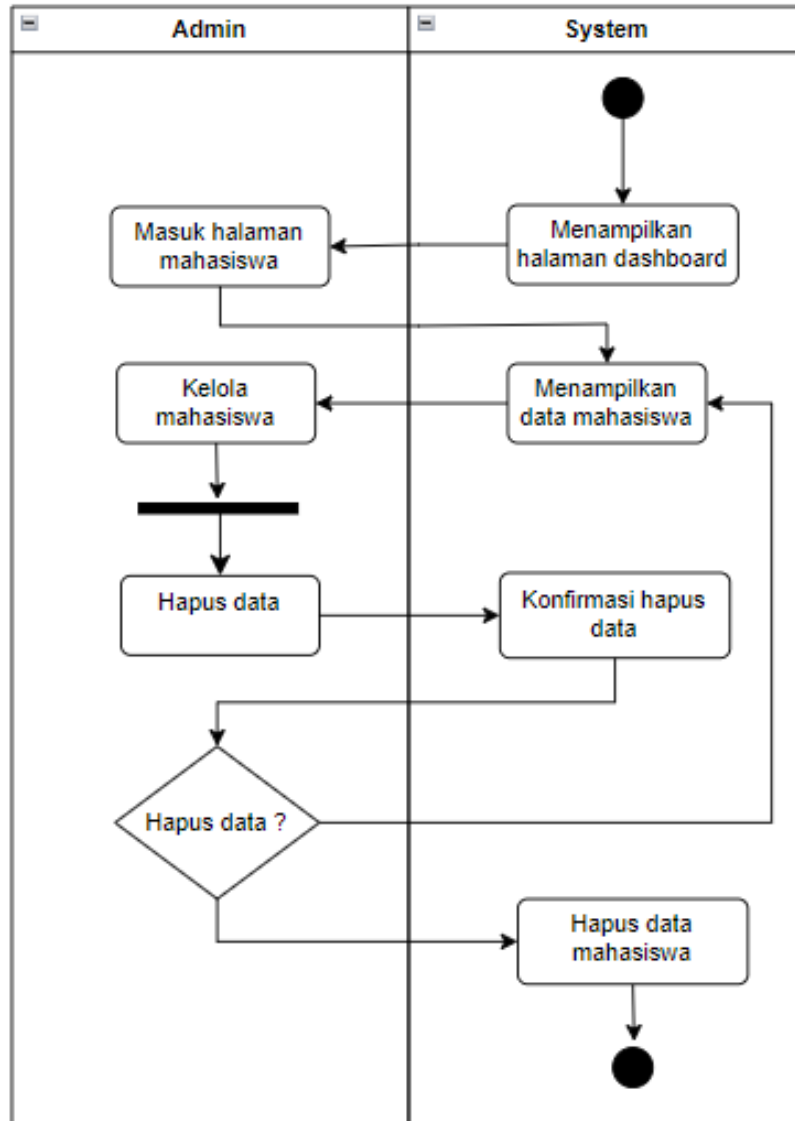
Asprak dapat menggenerate laporan presensi dengan pergi ke halaman jadwal lalu memilih jadwal yang akan *digenerate* dan *mendownload* laporan presensi, sistem akan *menggenerate* Pdf file yang berisi laporan presensi mahasiswa jadwal tersebut. Gambar 41 merupakan *activity* diagram dari proses pembuatan laporan presensi oleh asprak.



Gambar 41. Activity Laporan Asprak

6. *Activity* Mengelola Mahasiswa (Admin)

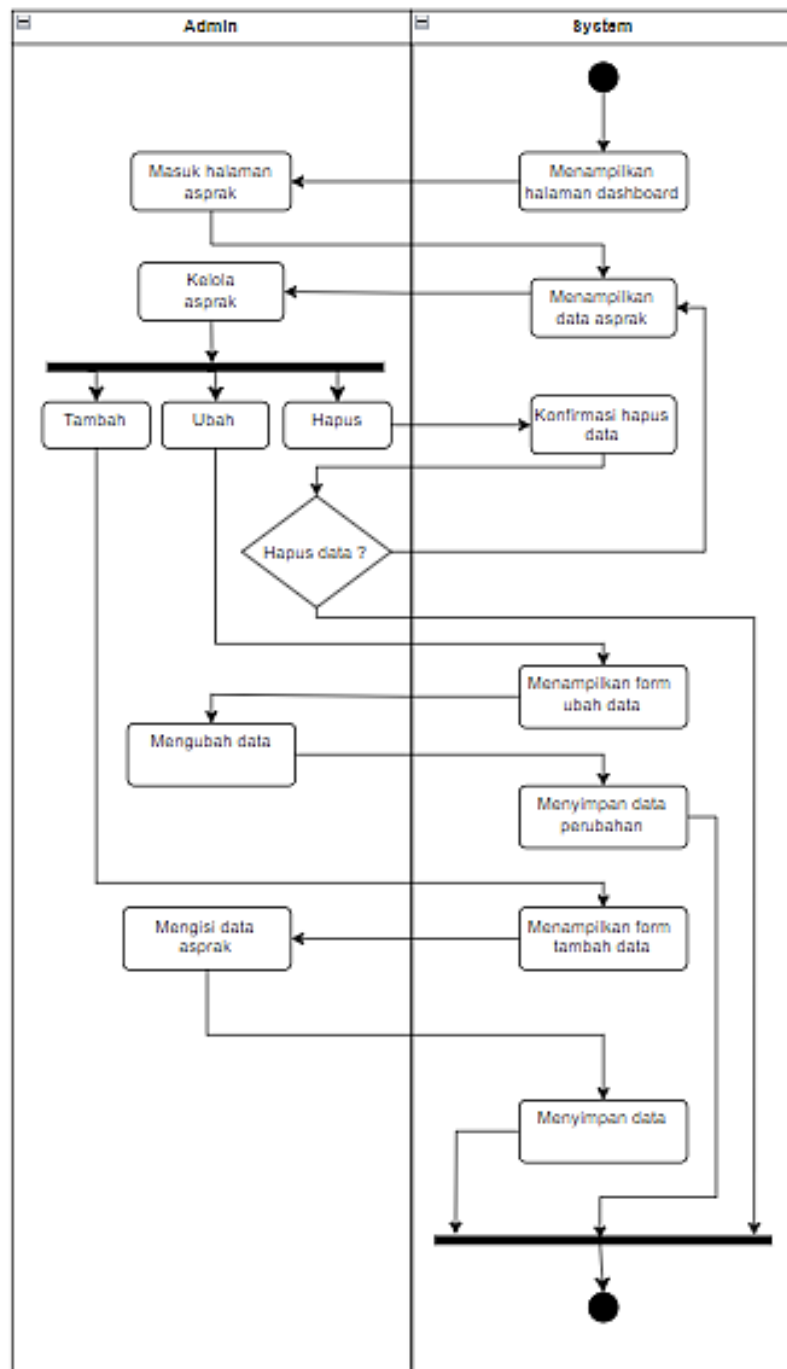
Admin dapat mengelola data mahasiswa seperti melihat, dan menghapus. Admin dapat pergi ke halaman mahasiswa lalu sistem akan menampilkan seluruh data mahasiswa yang sudah terdaftar selanjutnya admin dapat melakukan modifikasi seperti menghapus data. Gambar 42 merupakan *activity* diagram dari proses mengelola mahasiswa.



Gambar 42. *Activity* Mengelola Mahasiswa

7. *Activity* Mengelola Asprak (Admin)

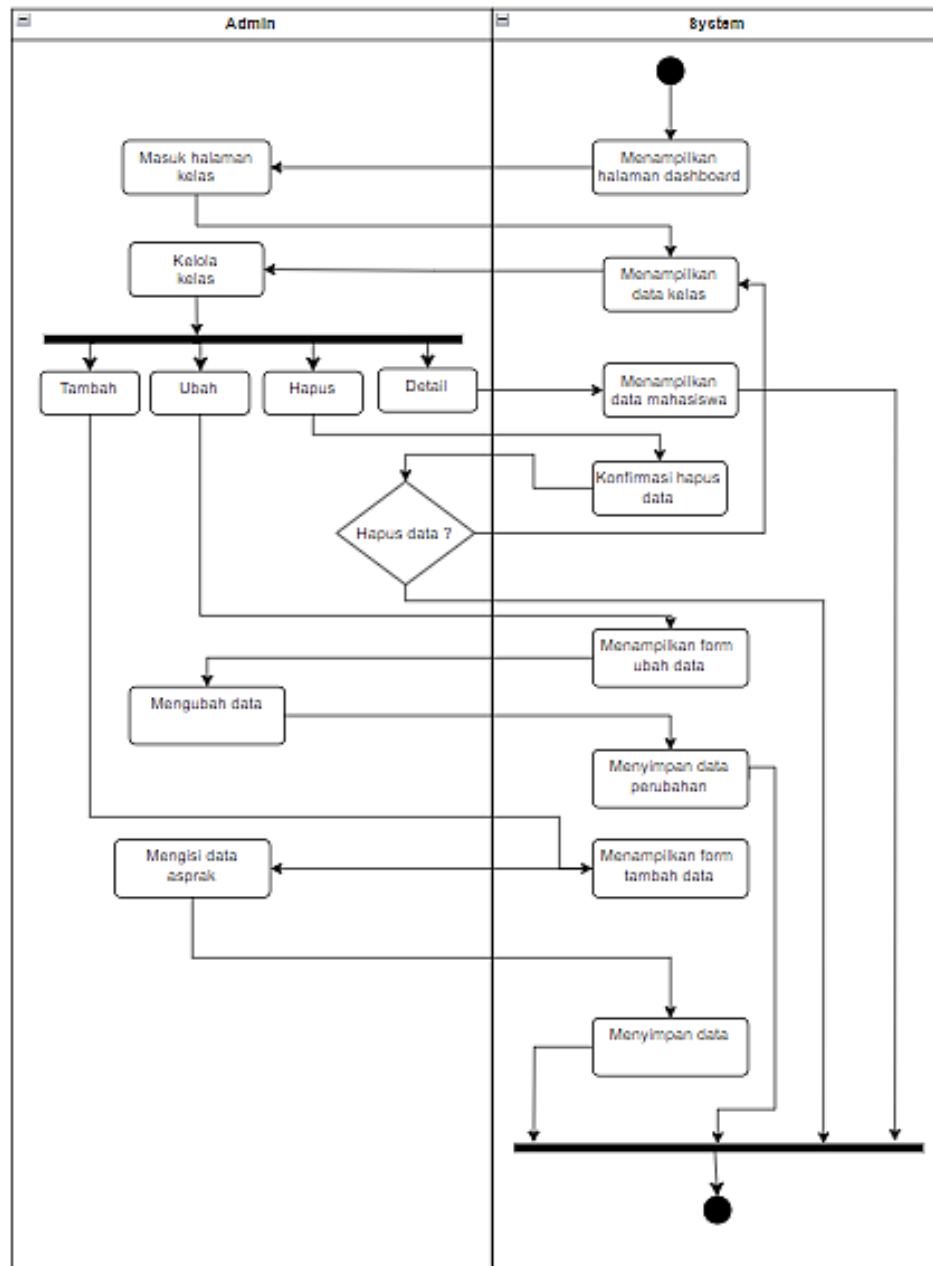
Admin memiliki kemampuan untuk mengelola informasi mengenai asprak, termasuk melihat, menambahkan, mengubah, dan menghapusnya. Admin dapat mengakses halaman asprak, di mana sistem akan menampilkan seluruh data asprak yang terdaftar. Selanjutnya, admin memiliki otoritas untuk melakukan perubahan dan penghapusan data tersebut sesuai kebutuhan. Gambar 43 merupakan *activity* diagram dari proses mengelola asprak.



Gambar 43. Activity Mengelola Asprak

8. *Activity Mengelola Kelas (Admin)*

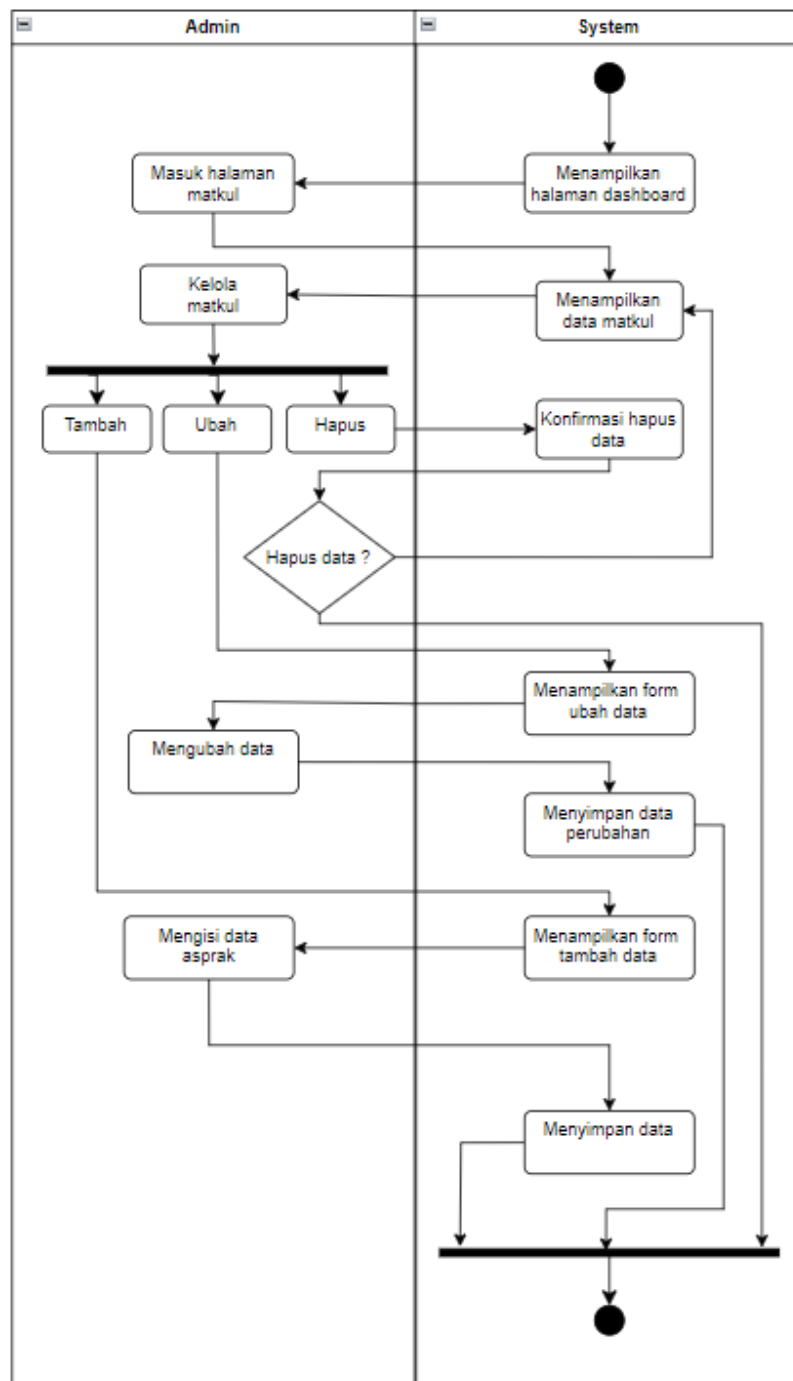
Admin memiliki hak akses mengelola kelas secara menyeluruh. Admin dapat melakukan tindakan seperti melihat, menambahkan, menghapus, mengubah data kelas dan dapat melihat detail daftar mahasiswa yang terdaftar dalam kelas tersebut. Admin dapat mengakses halaman kelas dan dapat melakukan modifikasi sesuai kebutuhan yang ada. Gambar 44 merupakan *activity* diagram proses mengelola kelas.



Gambar 44. Activity Mengelola Kelas

9. Activity Mengelola Matkul (Admin)

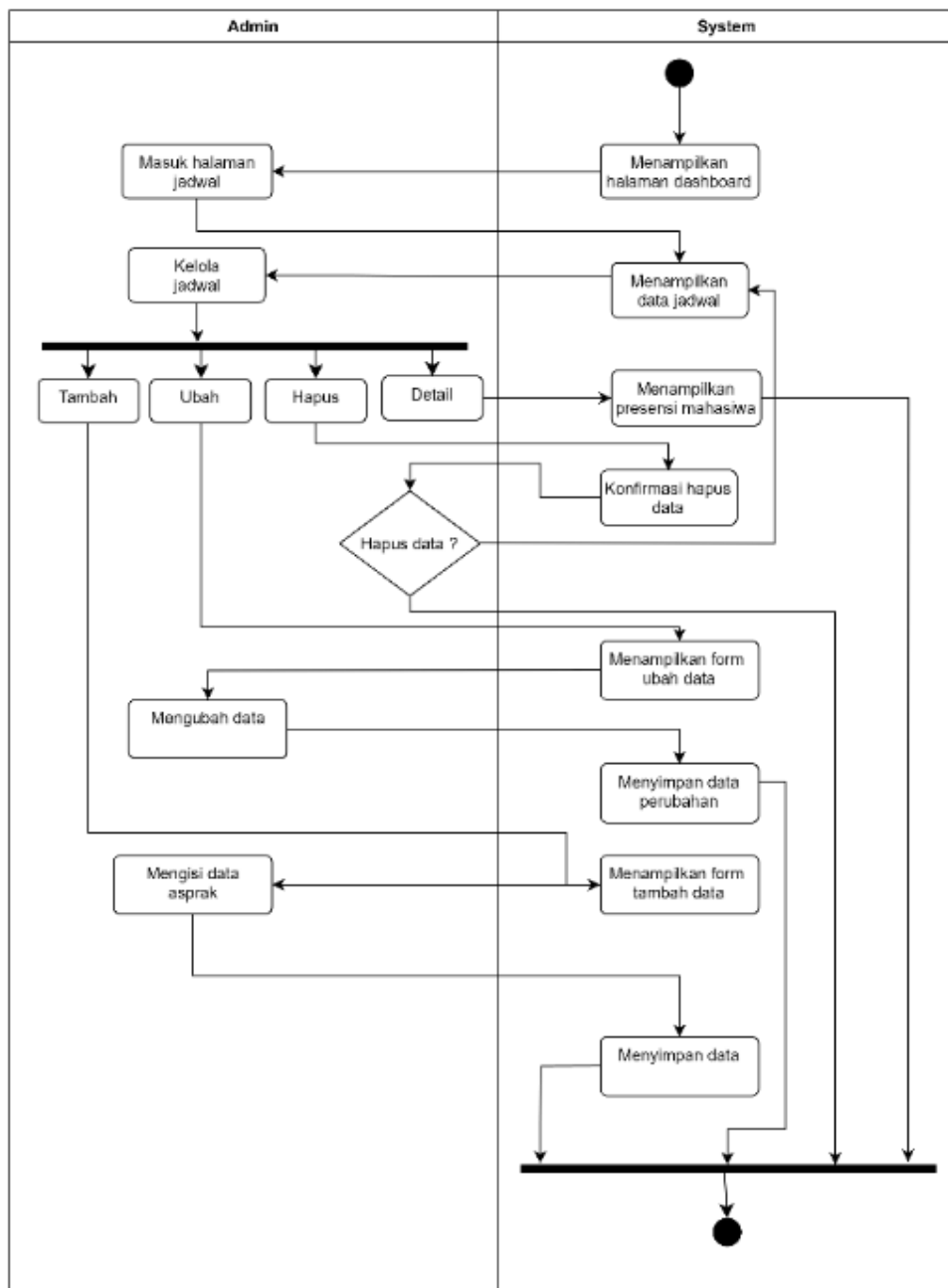
Admin juga memiliki hak akses untuk mengelola mata kuliah secara keseluruhan seperti dapat melihat, menambahkan, menghapus dan mengubah data mata kuliah. Admin dapat pergi ke halaman mata kuliah dan melakukan modifikasi sesuai kebutuhan. Gambar 45 merupakan *activity* diagram proses mengelola matkul.



Gambar 45. Activity Mengelola Matkul

10. Activity Mengelola Jadwal (Admin)

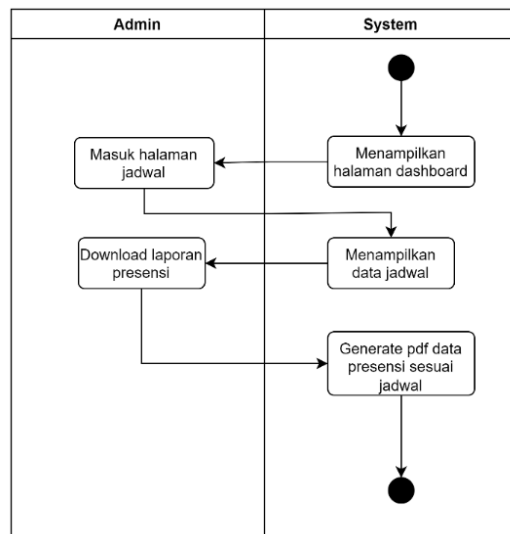
Admin juga dapat melakukan perubahan pada data jadwal dan memiliki hak akses akan hal tersebut. Admin dapat mengelola jadwal seperti menambahkan, mengubah, menghapus data jadwal, dan dapat melihat presensi kehadiran mahasiswa pada waktu jadwal berlangsung. Setelah sistem menampilkan data jadwal keseluruhan maka admin dapat melakukan modifikasi sesuai kebutuhan. Gambar 46 merupakan *activity* diagram dari proses mengelola jadwal.



Gambar 46. Activity Mengelola Jadwal

11. Activity Laporan Presensi (Admin)

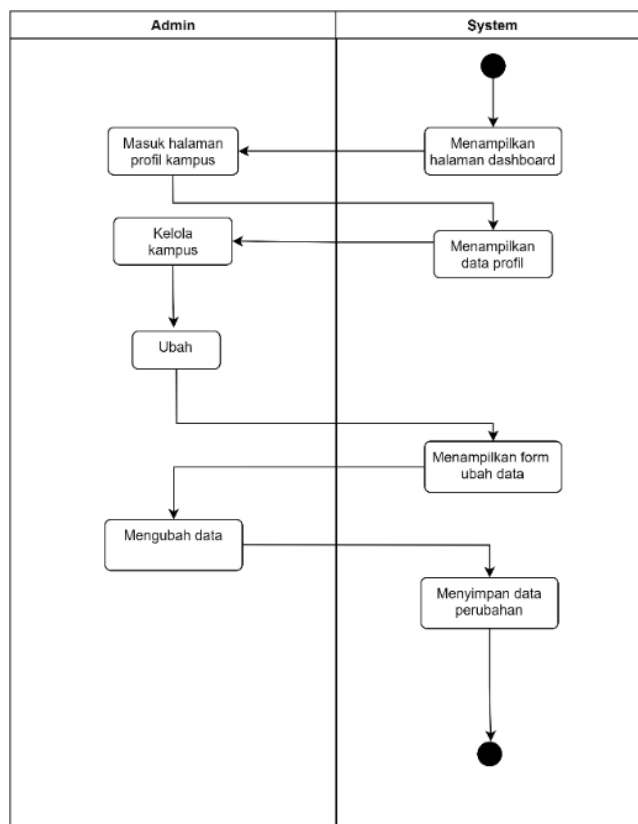
Admin dapat *generate* laporan presensi dengan pergi ke halaman jadwal lalu memilih jadwal yang akan *generate* dan *download* laporan presensi, sistem akan *generate* Pdf file yang berisi laporan presensi mahasiswa jadwal tersebut. Gambar 47 merupakan *activity* diagram proses laporan admin.



Gambar 47. Activity Laporan Admin

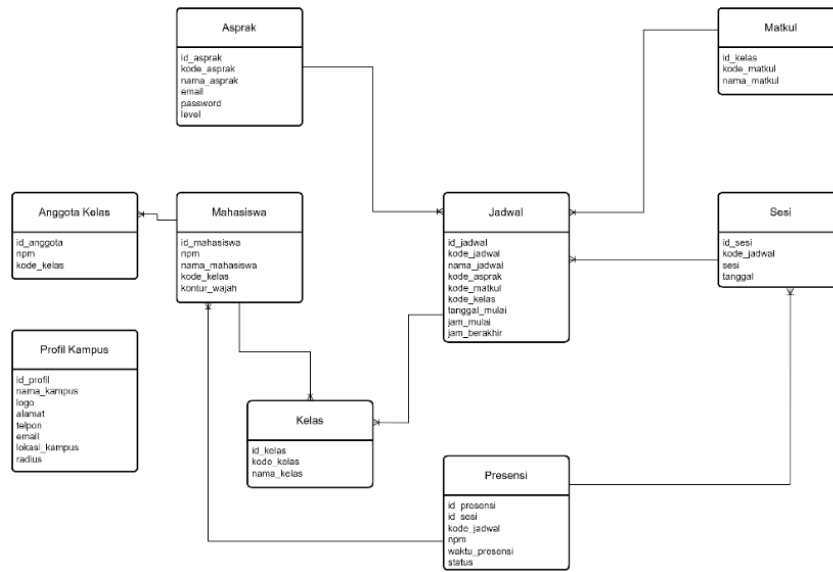
12. Activity Mengelola Profil Kampus (Admin)

Admin dapat melakukan perubahan pada profil kampus dan memiliki hak akses akan hal tersebut. Admin dapat mengubah data profil kampus dan dapat mengubah koordinat serta radius kampus untuk validasi saat proses presensi. Gambar 48 merupakan *activity* diagram proses mengelola profil kampus.



Gambar 48. Activity Mengelola Profil Kampus

Lampiran 6. ERD



Gambar 49. ERD

Sesuai yang tertera pada gambar, terdapat sembilan entitas dalam *database* yang digunakan pada Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* Menggunakan Algoritma *Euclidean Distance*.

1. Tabel Mahasiswa

Tabel ini digunakan untuk menyimpan data mahasiswa dan kontur wajah yang akan digunakan untuk presensi. Tabel ini memiliki relasi dengan tabel Kelas.

Tabel 18. Tabel Mahasiswa

No	Atribut	Tipe Data	Keterangan
1	id_mahasiswa	integer	Primary Key
2	Npm	varchar	Foreign key
3	nama_mahasiswa	varchar	Not null
4	kode_kelas	varchar	Foreign key
5	kontur_wajah	longtext	Not null

2. Tabel Asprak

Tabel ini digunakan untuk menyimpan data asprak termasuk email dan *password* yang akan digunakan untuk login ke halaman *dashboard*.

Tabel 19. Tabel Asprak

No	Atribut	Tipe Data	Keterangan
1	id_asprak	integer	Primary Key
2	kode_asprak	varchar	Foreign key
3	nama_asprak	varchar	Not null

4	Email	varchar	Not null
5	Password	varchar	Not null
6	Level	integer	Not null

3. Tabel Kelas

Tabel ini digunakan untuk menyimpan data kelas. Tabel ini akan dipakai saat pembuatan jadwal pada tabel Jadwal.

Tabel 20. Tabel Kelas

No	Atribut	Tipe Data	Keterangan
1	id_kelas	integer	Primary Key
2	kode_kelas	varchar	Foreign key
3	nama_kelas	varchar	Not null

4. Tabel Matkul

Tabel digunakan untuk menyimpan data matkul. Tabel ini akan dipakai saat pembuatan jadwal pada tabel Jadwal.

Tabel 21. Tabel Matkul

No	Atribut	Tipe Data	Keterangan
1	id_matkul	integer	Primary Key
2	kode_matkul	varchar	Foreign key
3	nama_matkul	varchar	Not null

5. Tabel Jadwal

Tabel digunakan untuk menyimpan data jadwal. Tabel ini memiliki relasi dengan tabel Asprak, Matkul, dan Kelas.

Tabel 22. Tabel Jadwal

No	Atribut	Tipe Data	Keterangan
1	id_jadwal	integer	Primary Key
2	kode_jadwal	varchar	Foreign key
3	kode_asprak	varchar	Foreign key
4	kode_matkul	varchar	Foreign key
5	kode_kelas	varchar	Foreign key
6	tanggal_mulai	date	Not null
7	jam_mulai	time	Not null
8	jam_berakhir	time	Not null

6. Tabel Anggota Kelas

Tabel digunakan untuk menyimpan data anggota kelas. Tabel ini memiliki relasi dengan tabel Kelas

Tabel 23. Tabel Anggota Kelas

No	Atribut	Tipe Data	Keterangan
1	id_anggota	integer	Primary Key
2	npm	varchar	Foreign key
3	kode_kelas	varchar	Foreign key

7. Tabel Sesi

Tabel digunakan untuk menyimpan data sesi. Tabel ini memiliki relasi dengan tabel Jadwal

Tabel 24. Tabel Sesi

No	Atribut	Tipe Data	Keterangan
1	id_sesi	integer	Primary Key
2	kode_jadwal	varchar	Foreign key
3	sesi	bigint	Not null
4	tanggal	date	Not null

8. Tabel Presensi

Tabel digunakan untuk menyimpan data presensi. Tabel ini memiliki relasi dengan tabel Jadwal dan Mahasiswa

Tabel 25. Tabel Presensi

No	Atribut	Tipe Data	Keterangan
1	id_presensi	integer	Primary Key
2	id_sesi	integer	Foreign key
3	kode_jadwal	varchar	Foreign key
4	npm	varchar	Foreign key
5	waktu_presensi	time	Not null
6	status	enum	Not null

9. Tabel Profil Kampus

Tabel digunakan untuk menyimpan data profil kampus. Tabel ini juga digunakan untuk menyimpan Lokasi kampus dan radius yang akan digunakan untuk validasi saat proses presensi.

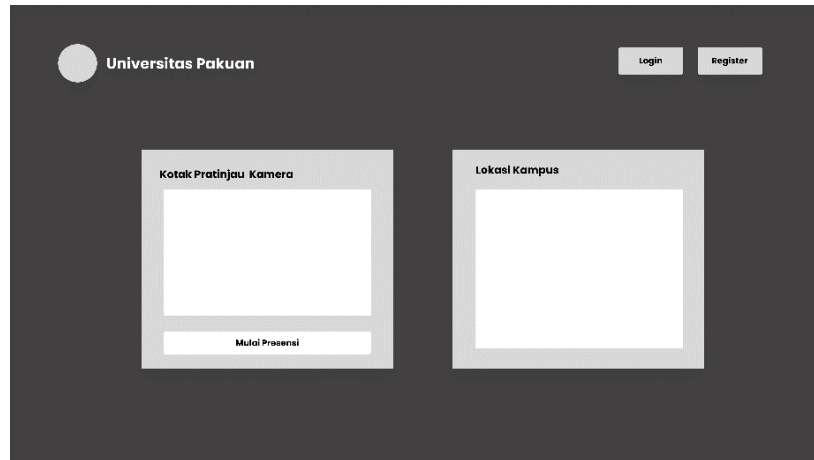
Tabel 26. Tabel Profil Kampus

No	Atribut	Tipe Data	Keterangan
1	id_profil	integer	Primary Key

2	nama_kampus	varchar	Not null
3	logo	varchar	Not null
4	alamat	text	Not null
5	telpon	varchar	Not null
6	email	varchar	Not null
7	lokasi_kampus	varchar	Not null
8	radius	integer	Not null

Lampiran 7. Mockup Website

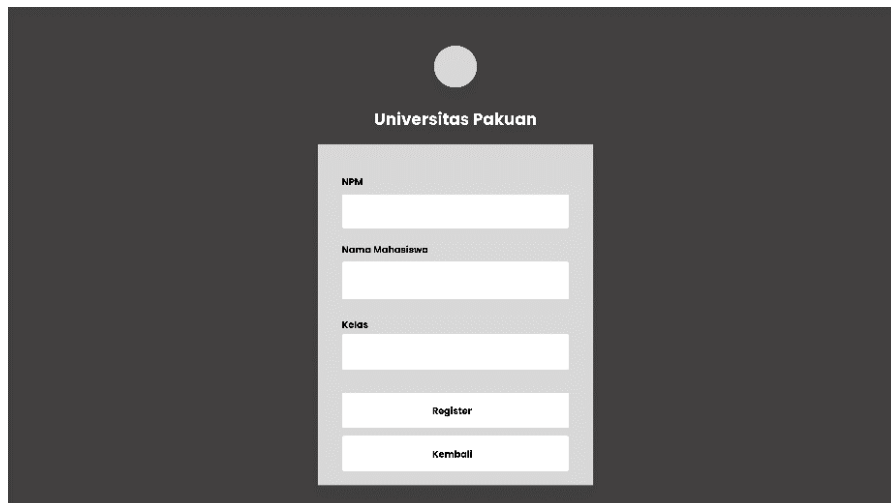
1. Mockup Halaman Utama



Gambar 50. Halaman Utama

Gambar 50 merupakan *mockup* dari halaman utama dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang akan muncul pertama kali ketika diakses. Pada halaman ini, terdapat kotak pratinjau kamera dan kotak lokasi kampus, serta tombol yang dapat digunakan oleh mahasiswa untuk melakukan registrasi dan tombol untuk admin melakukan *login*.

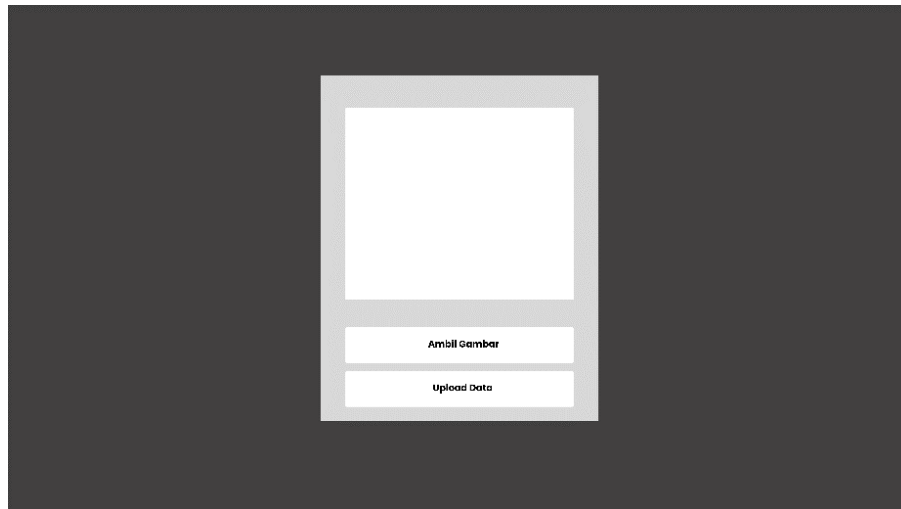
2. Mockup Halaman Register



Gambar 51. Mockup Halaman Register

Gambar 51 merupakan *mockup* dari halaman *register* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web*. Pada halaman ini, terdapat *form* untuk memasukkan NPM, nama mahasiswa dan kelas yang digunakan untuk melakukan registrasi data.

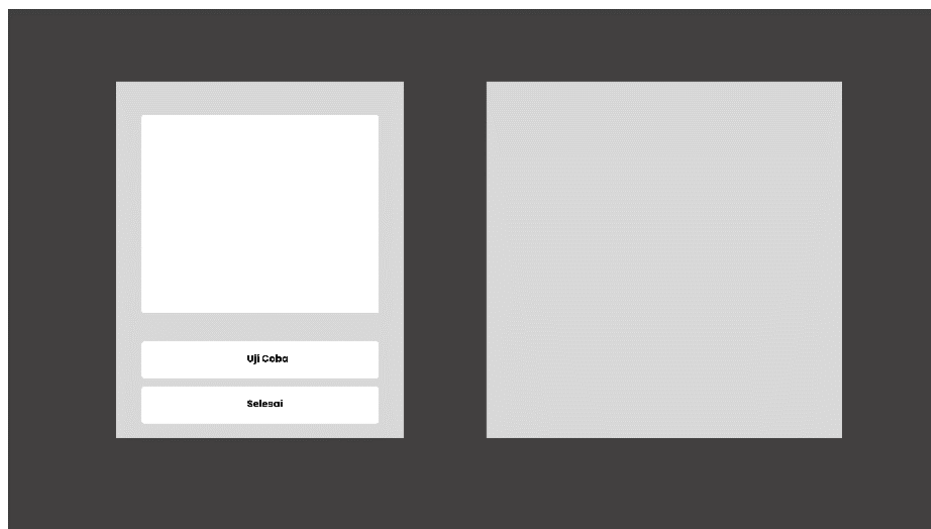
3. Mockup Halaman Register – Input Wajah



Gambar 52. *Mockup* Input Wajah

Gambar 52 merupakan *mockup* dari halaman input wajah dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web*. Pada halaman ini, terdapat kotak pratinjau kamera yang digunakan untuk melakukan input wajah mahasiswa.

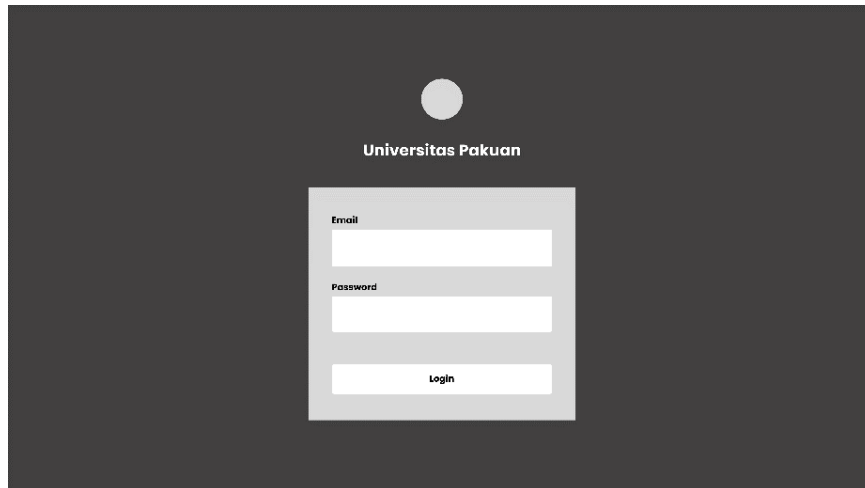
4. *Mockup* Halaman *Register* – Uji Coba Rekognisi



Gambar 53. *Mockup* Uji Coba Rekognisi

Gambar 53 merupakan *mockup* dari halaman uji coba rekognisi dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web*. Pada halaman ini, terdapat kotak pratinjau kamera yang digunakan untuk melakukan uji coba rekognisi wajah dan ada tabel untuk melakukan perulangan yang menampilkan uji coba rekognisi valid atau tidak.

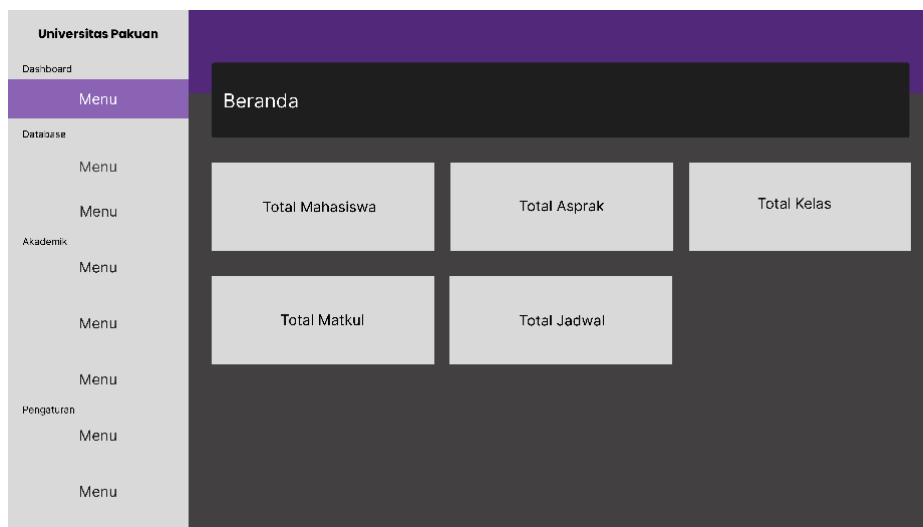
5. Halaman *Login*



Gambar 54. *Mockup* Halaman *Login*

Gambar 54 merupakan *mockup* dari halaman *login* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web*. Pada halaman ini, terdapat *form* untuk memasukkan email dan password yang sudah terdaftar yang digunakan untuk mengakses halaman *dashboard*.

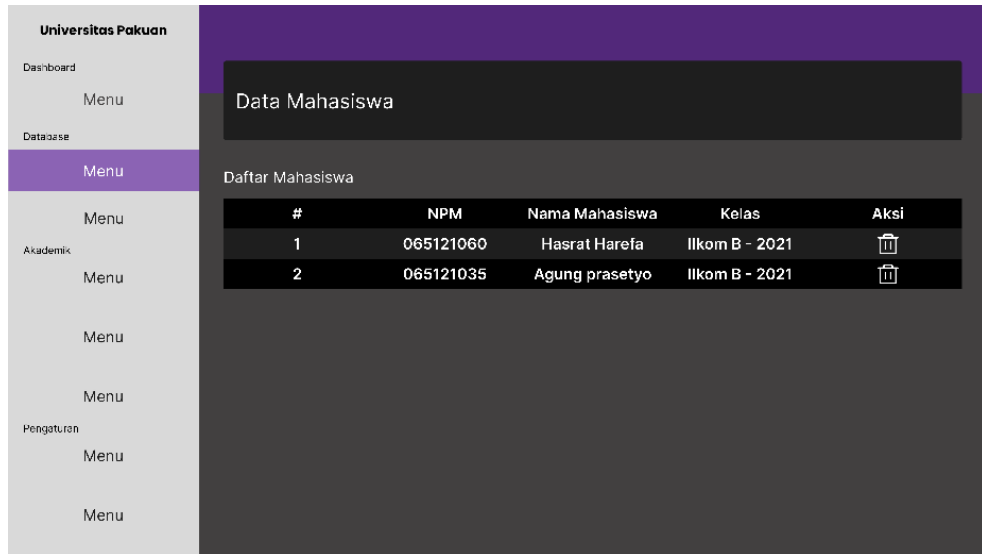
6. Halaman *Dashboard*



Gambar 55. *Mockup* Halaman *Dashboard*

Gambar 55 merupakan *mockup* dari halaman *dashboard* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang muncul pertama kali setelah admin melakukan *login*. Pada halaman ini, terdapat judul aplikasi pada bagian atas serta ringkasan jumlah data mahasiswa, asprak, kelas, matkul, dan jadwal yang tercatat di dalam *database* pada bagian bawahnya. Selain itu, pada bagian kiri terdapat *navbar* yang berisi daftar menu pada halaman *dashboard*. *Navbar* tersebut berperan untuk memudahkan admin ketika ingin berpindah ke halaman lain.

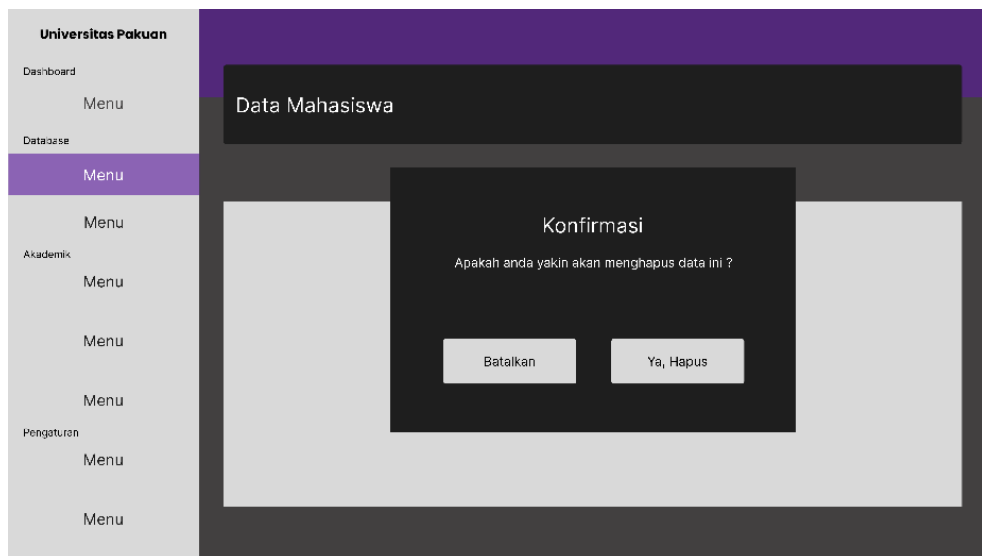
7. Halaman Data Mahasiswa



Gambar 56. *Mockup* Halaman Data Mahasiswa

Gambar 56 merupakan *mockup* dari halaman Data Mahasiswa pada *dashboard* admin yang menampilkan daftar seluruh mahasiswa pada *database* dalam bentuk tabel. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data mahasiswa.

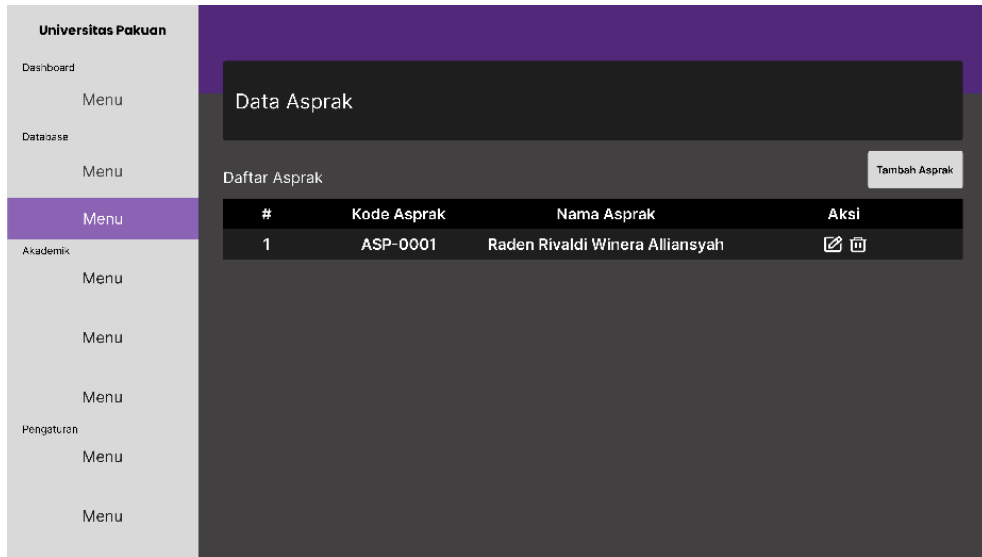
8. Halaman Data Mahasiswa – Hapus Data Mahasiswa



Gambar 57. *Mockup* Hapus Data Mahasiswa

Gambar 57 merupakan *mockup* hapus data mahasiswa dari halaman Data Mahasiswa pada *dashboard* admin yang menampilkan *pop up* untuk konfirmasi penghapusan data ketika tombol hapus ditekan.

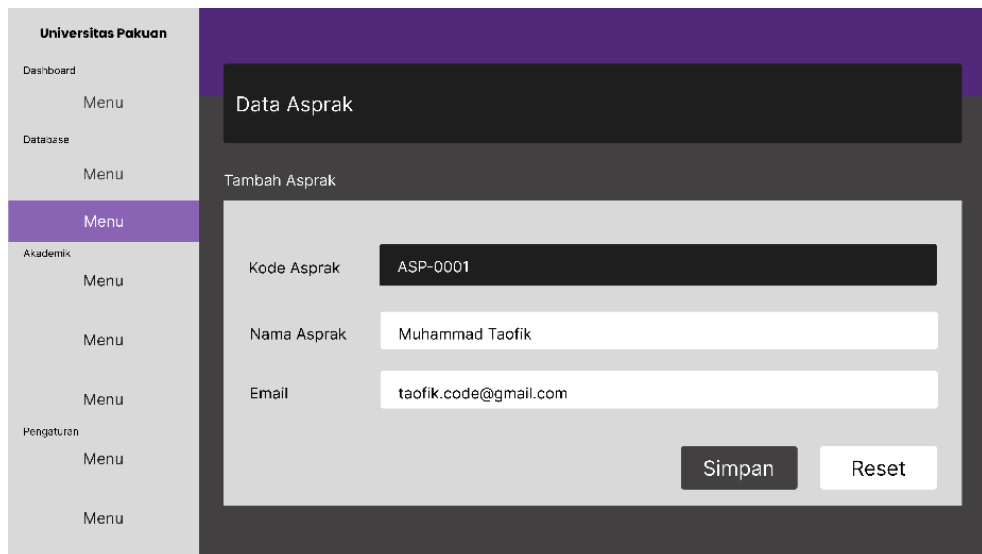
9. Halaman Data Asprak



Gambar 58. *Mockup* Halaman Data Asprak

Gambar 58 merupakan mockup dari halaman Data Asprak pada *dashboard* admin yang menampilkan daftar seluruh asprak pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah asprak yang digunakan untuk menambah data asprak. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data asprak dan tombol *edit* untuk mengubah data asprak.

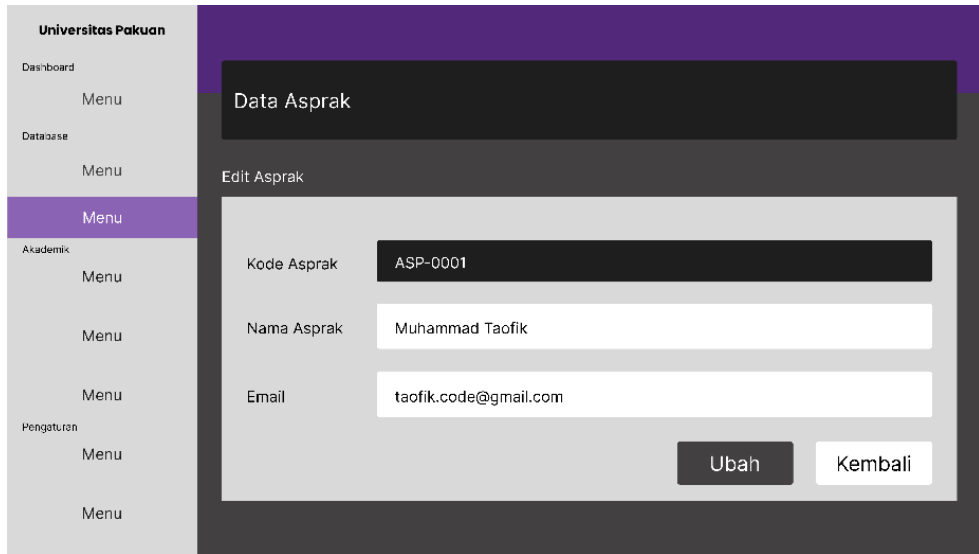
10. Halaman Data Asprak – Tambah Data Asprak



Gambar 59. *Mockup* tambah Data asprak

Gambar 59 merupakan *mockup* tambah data asprak dari halaman Data Asprak pada *dashboard* admin yang menampilkan *form* untuk mengisi data asprak yang akan ditambahkan.

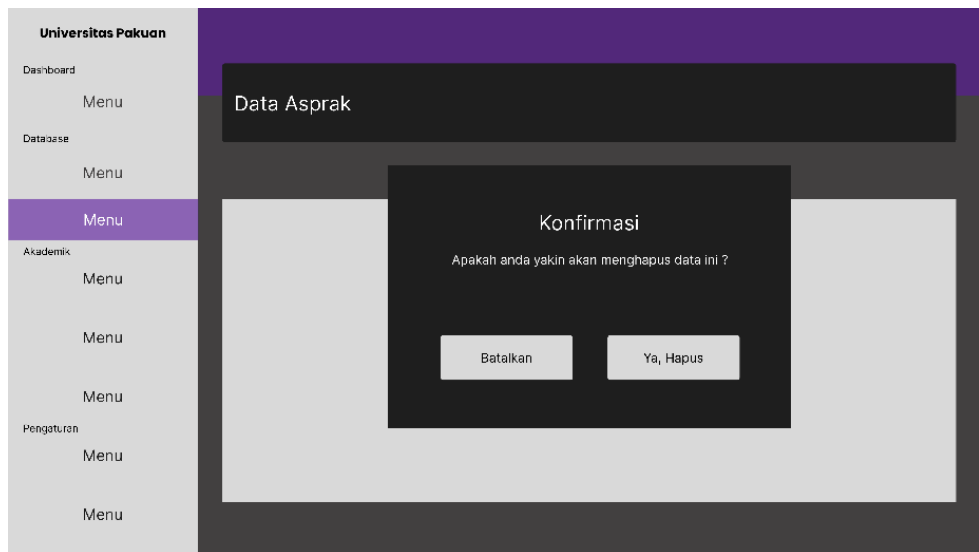
11. Halaman Data Asprak – *Edit* Data Asprak



Gambar 60. *Mockup* Edit Data Asprak

Gambar 60 merupakan *mockup* edit data asprak dari halaman Data Asprak pada *dashboard* admin yang menampilkan *form* untuk mengubah data asprak yang akan diubah.

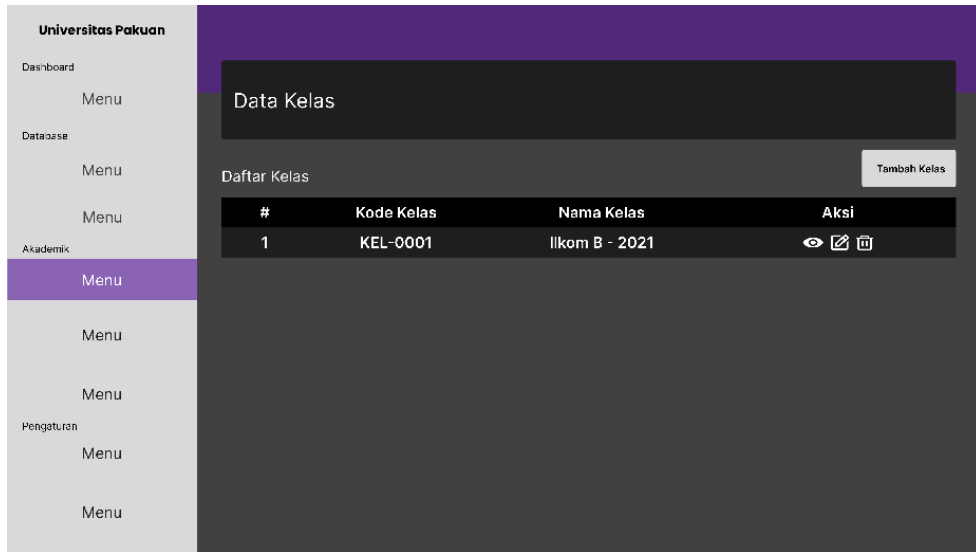
12. Halaman Data Asprak – Hapus Data Asprak



Gambar 61. *Mockup* Hapus Data Asprak

Gambar 61 merupakan *mockup* hapus data asprak dari halaman Data Asprak pada *dashboard* admin menampilkan *pop up* untuk konfirmasi penghapusan data ketika tombol hapus ditekan.

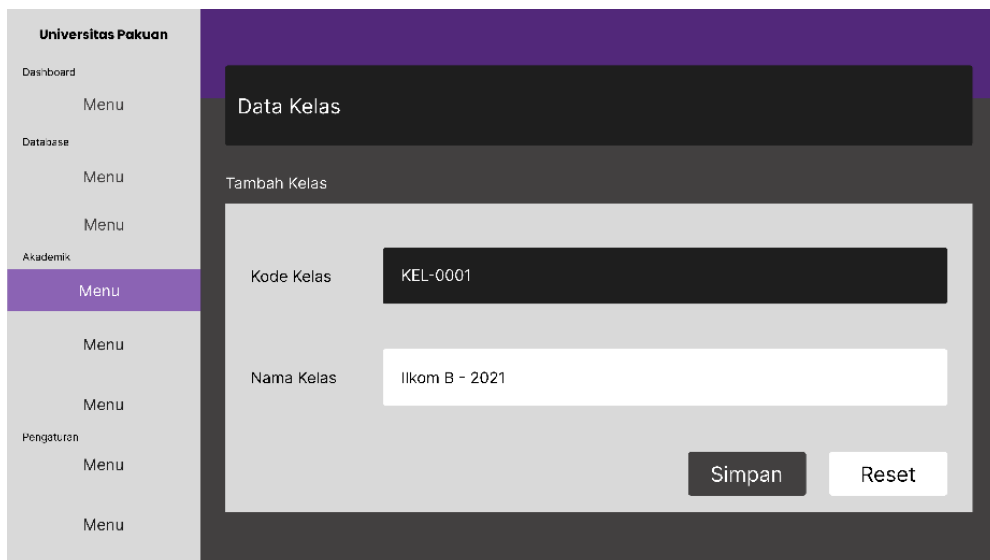
13. Halaman Data Kelas



Gambar 62. *Mockup* Halaman Data Kelas

Gambar 62 merupakan *mockup* dari halaman Data Kelas pada *dashboard* admin yang menampilkan daftar seluruh kelas pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah kelas yang digunakan untuk menambah data kelas. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol melihat detail kelas, tombol untuk menghapus data kelas dan tombol *edit* untuk mengubah data kelas.

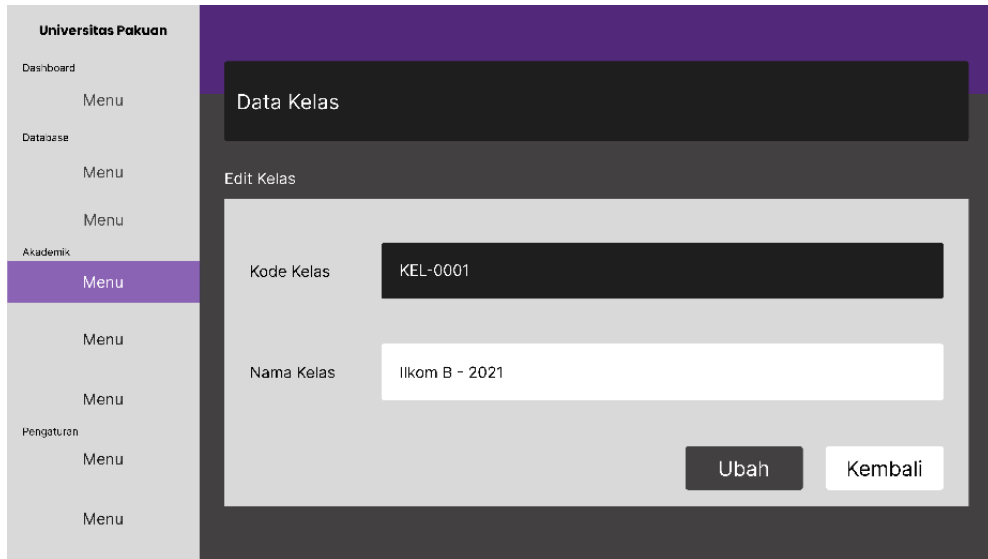
14. Halaman Data Kelas – Tambah Data Kelas



Gambar 63. *Mockup* Tambah Data Kelas

Gambar 63 merupakan *mockup* tambah data kelas dari halaman Data Kelas pada *dashboard* admin yang menampilkan form untuk mengisi data kelas yang akan ditambahkan.

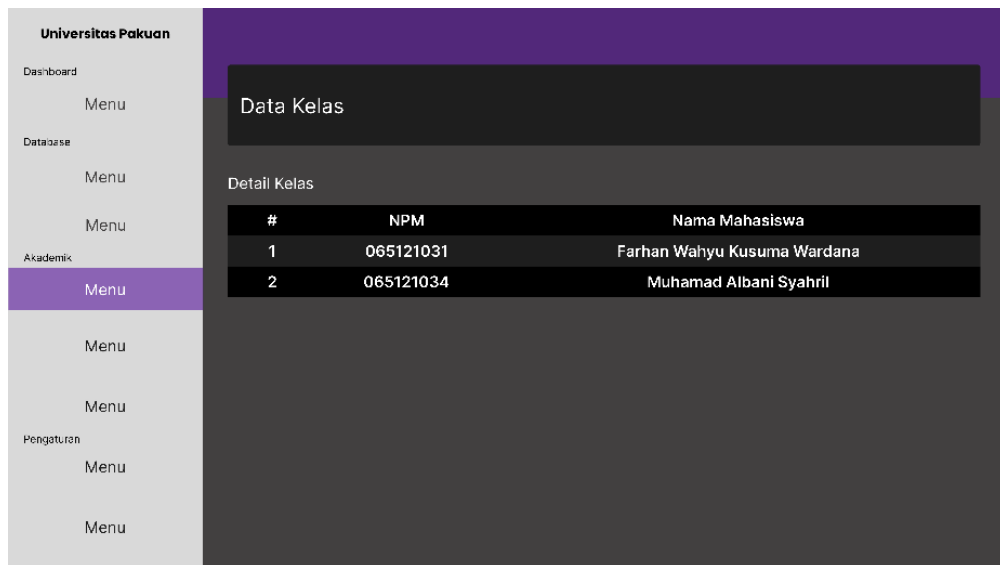
15. Halaman Data Kelas – *Edit* Data Kelas



Gambar 64. *Mockup* Edit Data Kelas

Gambar 64 merupakan *mockup edit* data kelas dari halaman Data Kelas pada *dashboard* admin yang menampilkan *form* untuk mengubah data kelas yang akan diubah.

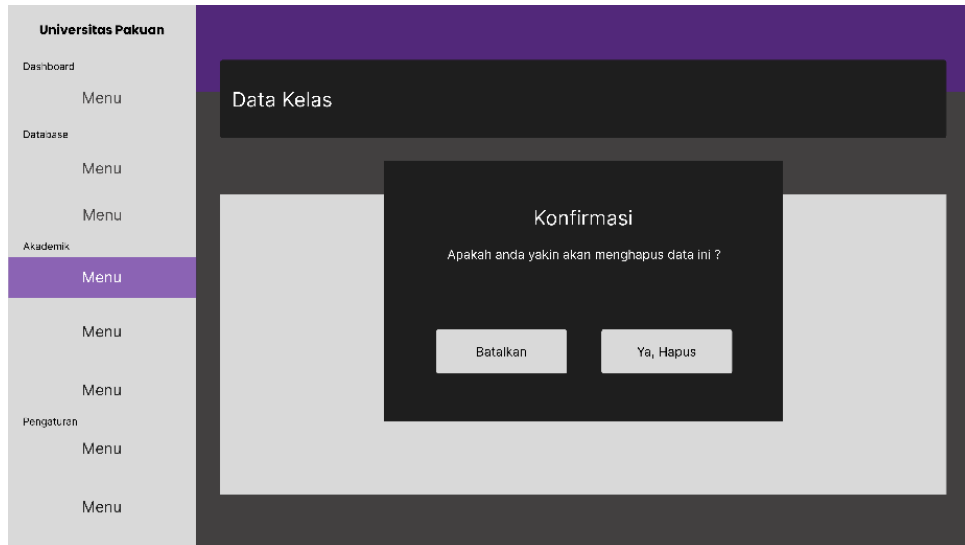
16. Halaman Data Kelas – Detail Data Kelas



Gambar 65. *Mockup* Detail Data Kelas

Gambar 65 merupakan *mockup* detail data kelas dari halaman Data Kelas pada *dashboard* admin yang menampilkan semua mahasiswa yang sesuai dengan kelasnya.

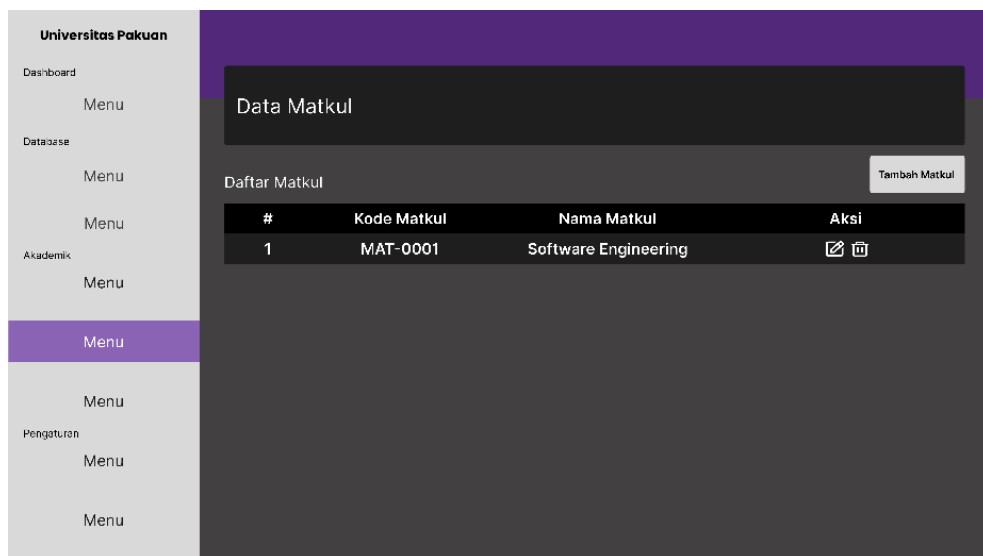
17. Halaman Data Kelas – Hapus Data Kelas



Gambar 66. *Mockup* Hapus Data Kelas

Gambar 66 merupakan *mockup* hapus data kelas dari halaman Data Kelas pada *dashboard* admin menampilkan *pop up* untuk konfirmasi penghapusan data ketika tombol hapus ditekan.

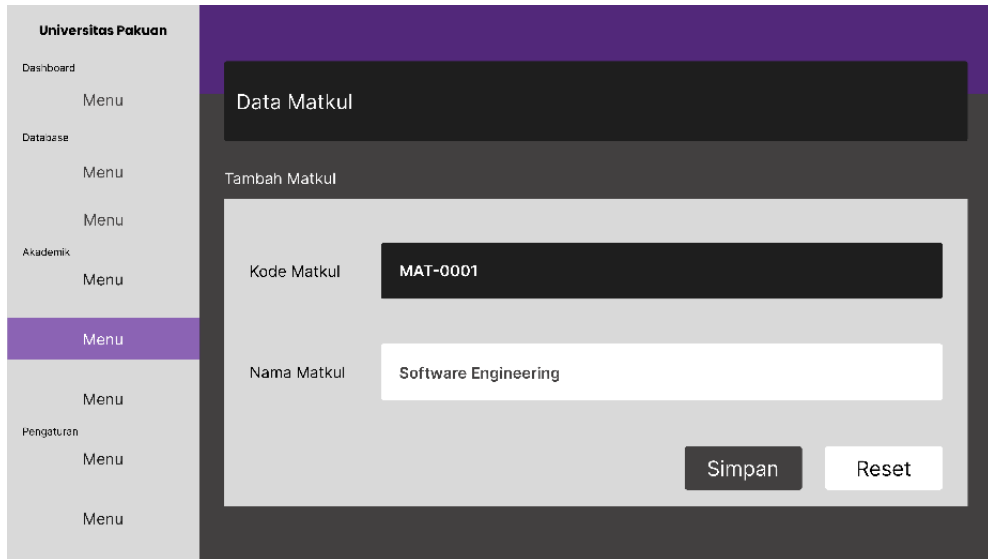
18. Halaman Data Matkul



Gambar 67. *Mockup* Halaman Data Matkul

Gambar 67 merupakan *mockup* dari halaman Data Matkul pada *dashboard* admin yang menampilkan daftar seluruh matkul pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah matkul yang digunakan untuk menambah data matkul. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data matkul dan tombol *edit* untuk mengubah data matkul.

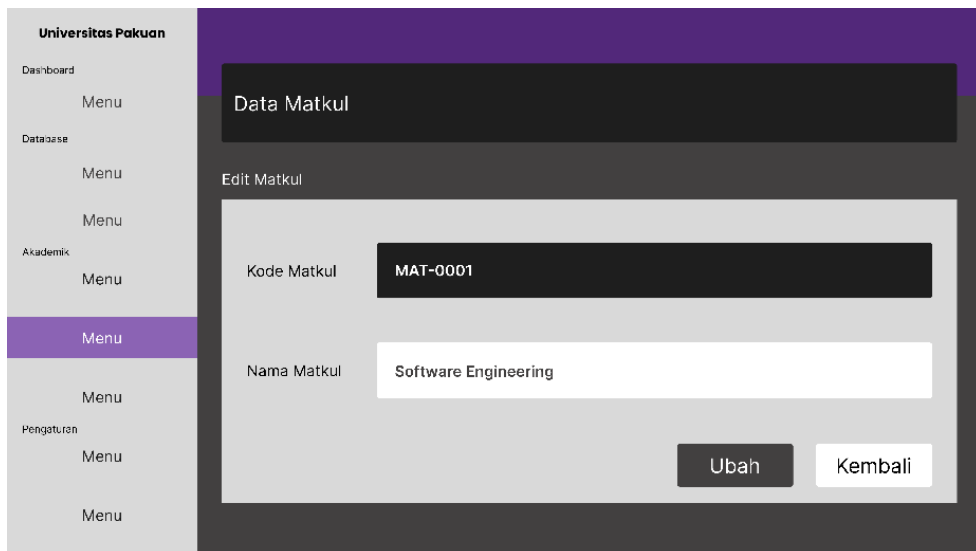
19. Halaman Data Matkul – Tambah Data Matkul



Gambar 68. *Mockup* Tambah Data Matkul

Gambar 68 merupakan *mockup* tambah data matkul dari halaman Data Matkul pada *dashboard* admin yang menampilkan *form* untuk mengisi data matkul yang akan ditambahkan.

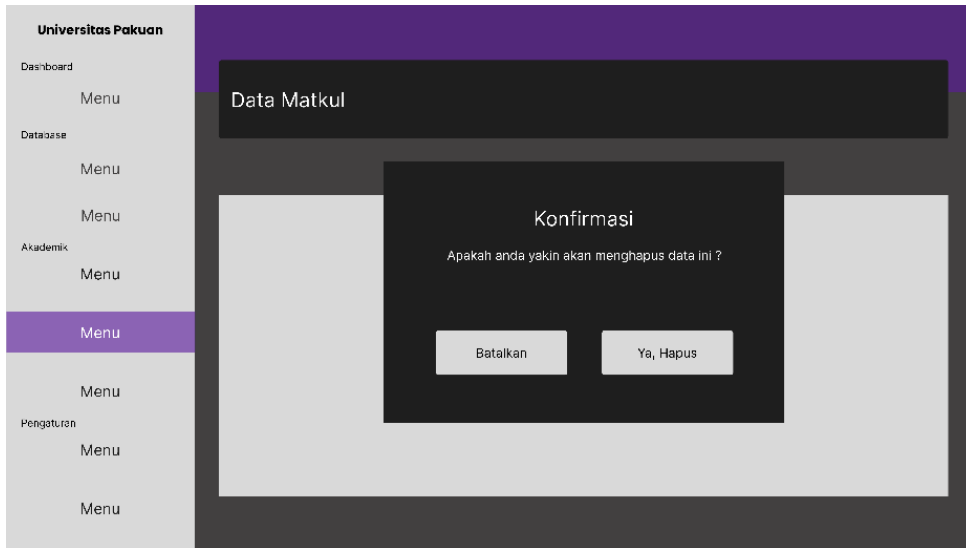
20. Halaman Data Matkul – *Edit* Data Matkul



Gambar 69. *Mockup* Edit Data Matkul

Gambar 69 merupakan *mockup* edit data matkul dari halaman Data Matkul pada *dashboard* admin yang menampilkan *form* untuk mengubah data matkul yang akan diubah.

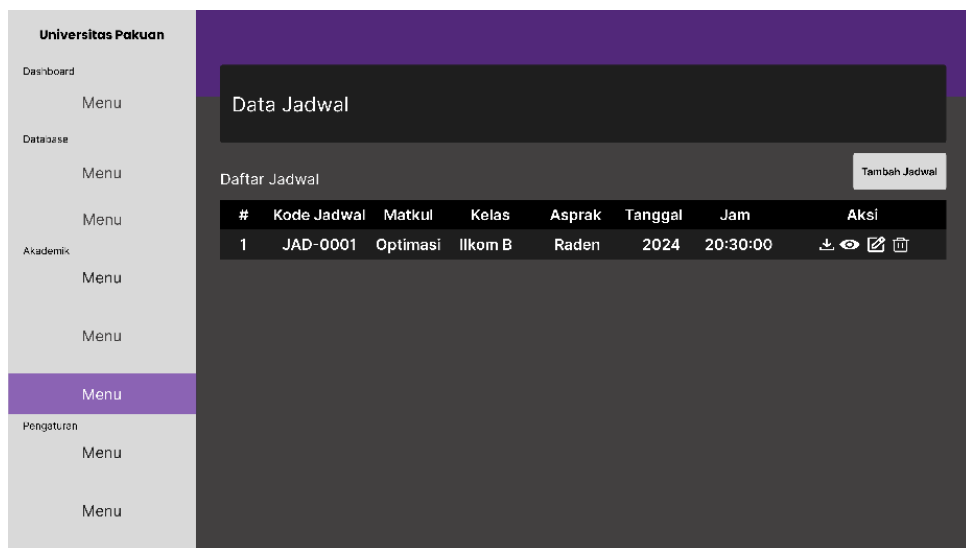
21. Halaman Data Matkul – Hapus Data Matkul



Gambar 70. *Mockup* Hapus Data Matkul

Gambar 70 merupakan *mockup* hapus data matkul dari halaman Data Matkul pada *dashboard* admin menampilkan *pop up* untuk konfirmasi penghapusan data ketika tombol hapus ditekan.

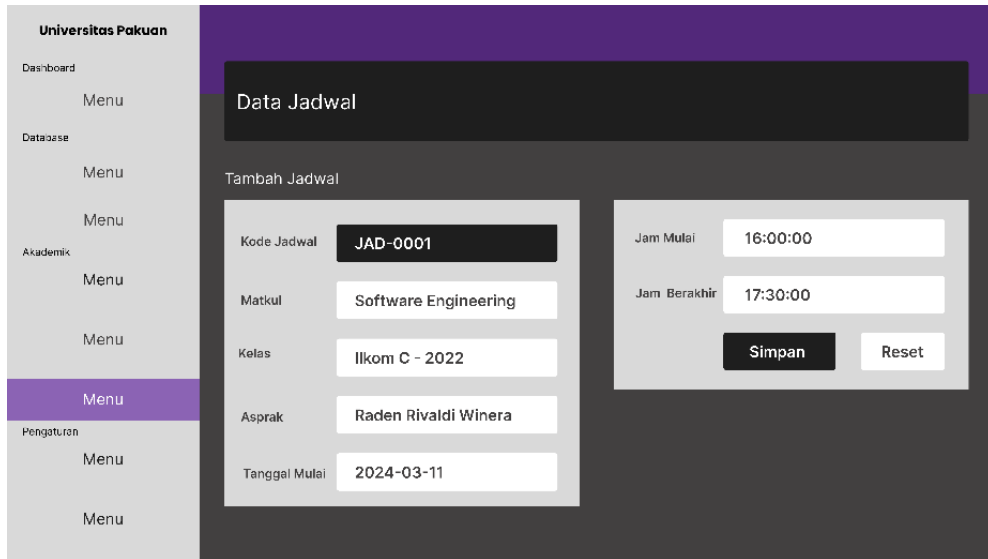
22. Halaman Data Jadwal



Gambar 71. *Mockup* Halaman Data Jadwal

Gambar 71 merupakan *mockup* dari halaman Data Matkul pada *dashboard* admin yang menampilkan daftar seluruh matkul pada database dalam bentuk tabel. Pada bagian atas, ada tombol tambah matkul yang digunakan untuk menambah data matkul. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data matkul dan tombol edit untuk mengubah data matkul.

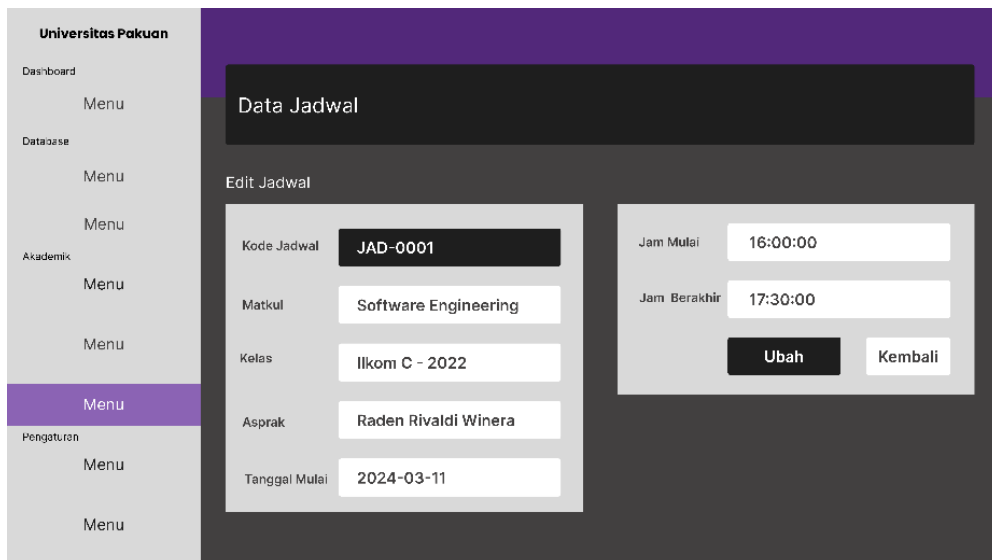
23. Halaman Data Jadwal – Tambah Data Jadwal



Gambar 72. *Mockup* Tambah Data Jadwal

Gambar 72 merupakan *mockup* tambah data jadwal dari halaman Data Jadwal pada *dashboard* admin yang menampilkan *form* untuk mengisi data jadwal yang akan ditambahkan.

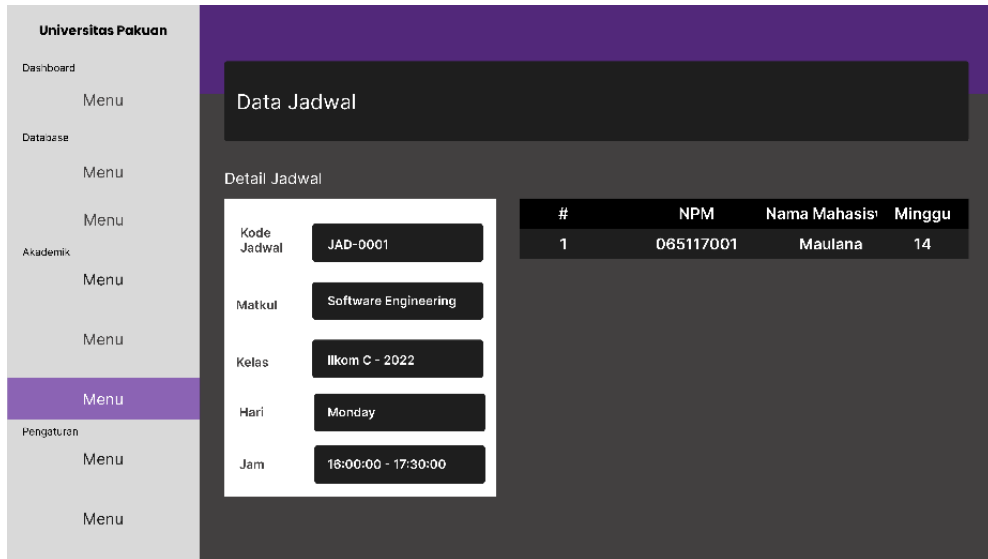
24. Halaman Data Jadwal – *Edit* Data Jadwal



Gambar 73. *Mockup* Edit Data Jadwal

Gambar 73 merupakan *mockup edit* data jadwal dari halaman Data Jadwal pada *dashboard* admin yang menampilkan *form* untuk mengubah data jadwal yang akan diubah.

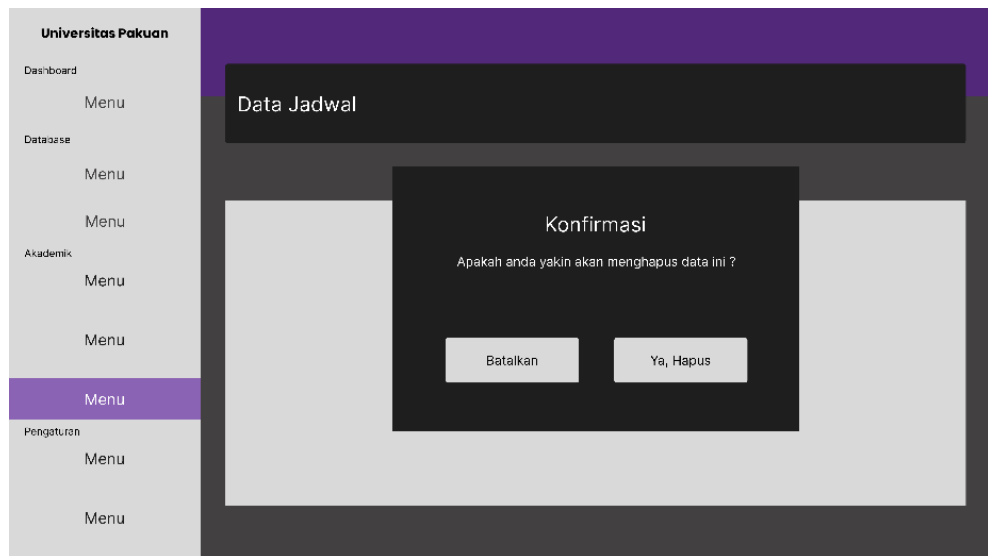
25. Halaman Data Jadwal – Detail Data Jadwal



Gambar 74. *Mockup* Detail Data Jadwal

Gambar 74 merupakan *mockup* detail data jadwal dari halaman Data Jadwal pada *dashboard* admin yang menampilkan informasi jadwal dan menampilkan presensi mahasiswa yang sesuai jadwalnya.

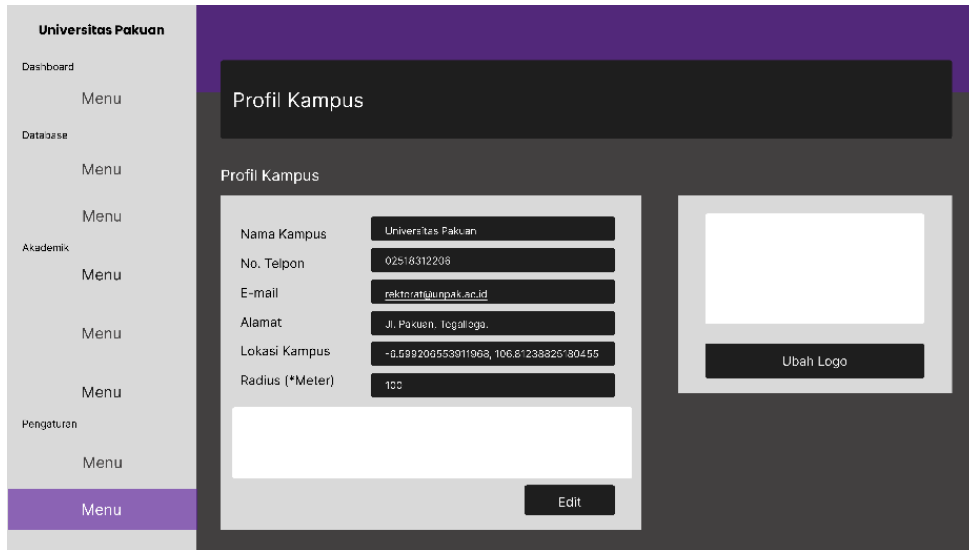
26. Halaman Data Jadwal – Hapus Data Jadwal



Gambar 75. *Mockup* Hapus Data Jadwal

Gambar 75 merupakan *mockup* hapus data jadwal dari halaman Data Jadwal pada *dashboard* admin menampilkan *pop up* untuk konfirmasi penghapusan data ketika tombol hapus ditekan.

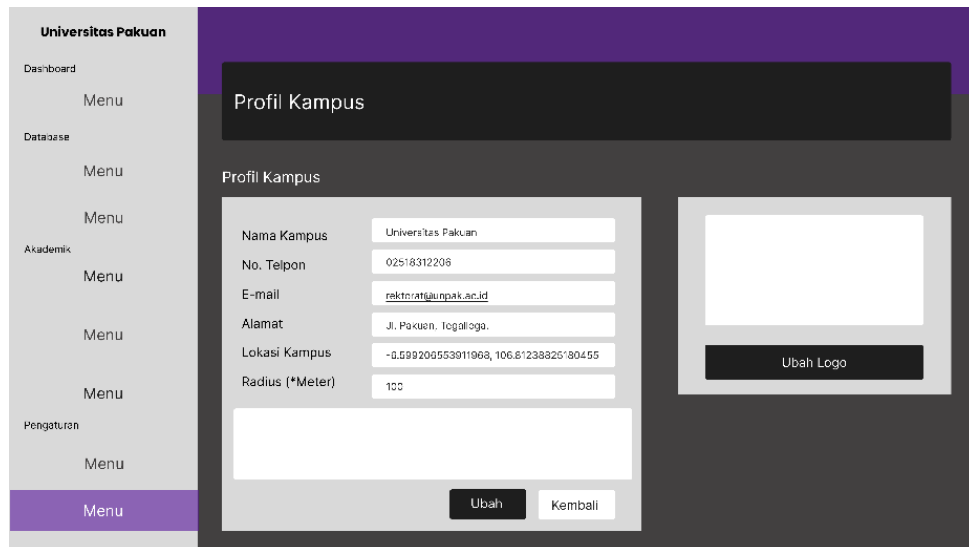
27. Halaman Profil Kampus



Gambar 76. *Mockup* Halaman Profil Kampus

Gambar 76 merupakan *mockup* dari halaman Profil Kampus pada dashboard admin yang menampilkan seluruh informasi kampus yang ada pada *database*. Pada bagian kanan, ada *form* logo untuk menampilkan logo kampus. Pada kedua form, terdapat tombol untuk mengubah informasi dan logo kampus.

28. Halaman Profil Kampus – *Edit* Profil Kampus



Gambar 77. *Mockup* Edit Profil Kampus

Gambar 77 merupakan *mockup edit* profil kampus dari halaman Profil Kampus pada *dashboard* admin yang menampilkan form untuk mengubah informasi kampus yang akan diubah.

29. Laporan Presensi Mahasiswa

Laporan Presensi

Kelas : Ilkom O - 2021
Matkul : Optimasi

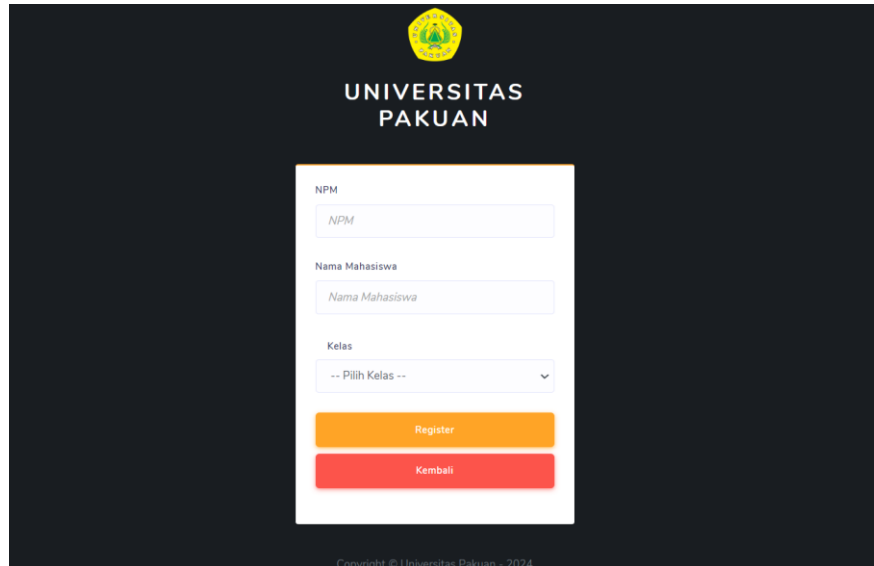
#	NPM	Nama Mahasiswa	1	2	3	4	5	6	7	8	9	...	14
1	065121005	Wisnu Dwi Pradana	-	-	-	-	-	-	-	-	-	-	H
2	065121012	Muhammad Alfi	-	-	-	-	-	-	-	-	-	-	-
3	065121016	Muhammad Syehan	-	-	-	-	-	-	-	-	-	-	H

Gambar 78. *Mockup* Laporan Presensi

Gambar 78 merupakan *mockup* laporan presensi mahasiswa dari laporan ini dapat di *download* dari Halaman Jadwal sesuai jadwal yang dipilih.

Lampiran 8. Tampilan Sistem Lainnya

1. Halaman *Register*



The screenshot shows the registration interface for Universitas Pakuan. At the top center is the university's logo, a yellow shield with a green emblem, and the text "UNIVERSITAS PAKUAN" below it. The main content is a white registration form with the following elements:

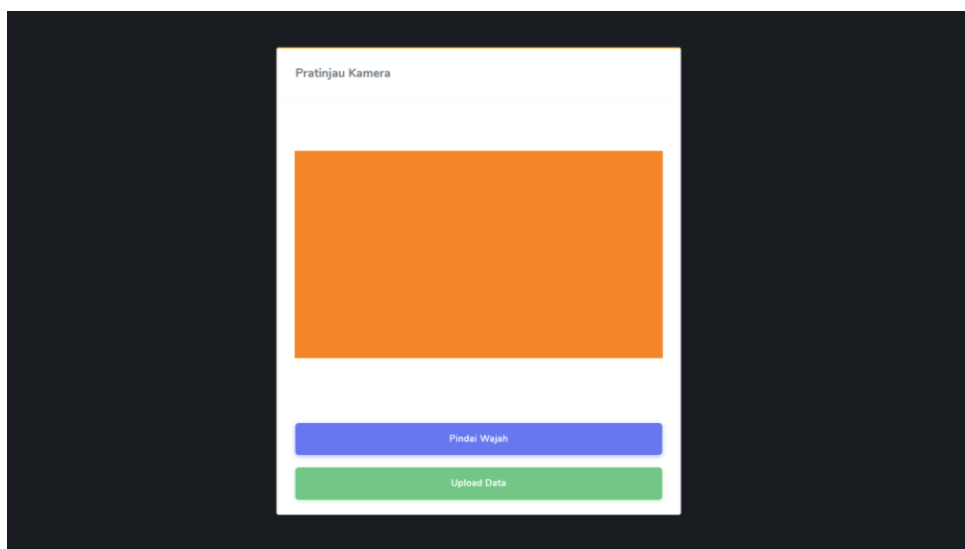
- A text input field labeled "NPM" with a placeholder "NPM".
- A text input field labeled "Nama Mahasiswa" with a placeholder "Nama Mahasiswa".
- A dropdown menu labeled "Kelas" with the text "-- Pilih Kelas --" and a downward arrow.
- An orange button labeled "Register".
- A red button labeled "Kembali".

At the bottom of the form, there is a small copyright notice: "Copyright © Universitas Pakuan - 2024".

Gambar 79. Halaman *Register*

Gambar 79 merupakan tampilan *register* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang digunakan mahasiswa untuk mendaftarkan diri dengan menampilkan *form* data diri dan meminta mahasiswa mengisinya

2. Halaman Input Wajah

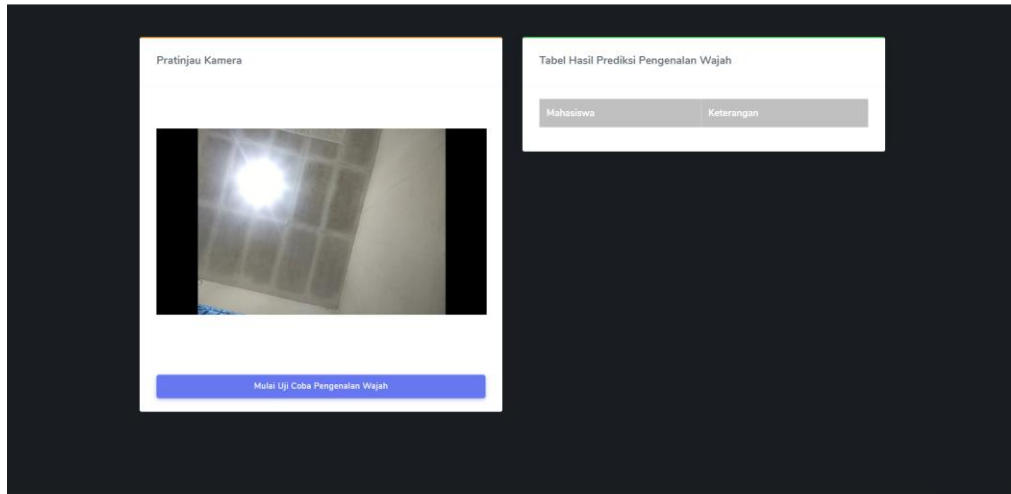


The screenshot displays the face input interface. It consists of a white rectangular area on a dark background. At the top of this area is the title "Pratinjau Kamera". Below the title is a large orange rectangle, which serves as a placeholder for the camera's view. At the bottom of the white area, there are two buttons: a blue button labeled "Pindai Wajah" and a green button labeled "Upload Data".

Gambar 80. Halaman Input Wajah

Gambar 80 merupakan tampilan input wajah dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang digunakan mahasiswa untuk mendaftarkan wajahnya dengan menampilkan *webcam*.

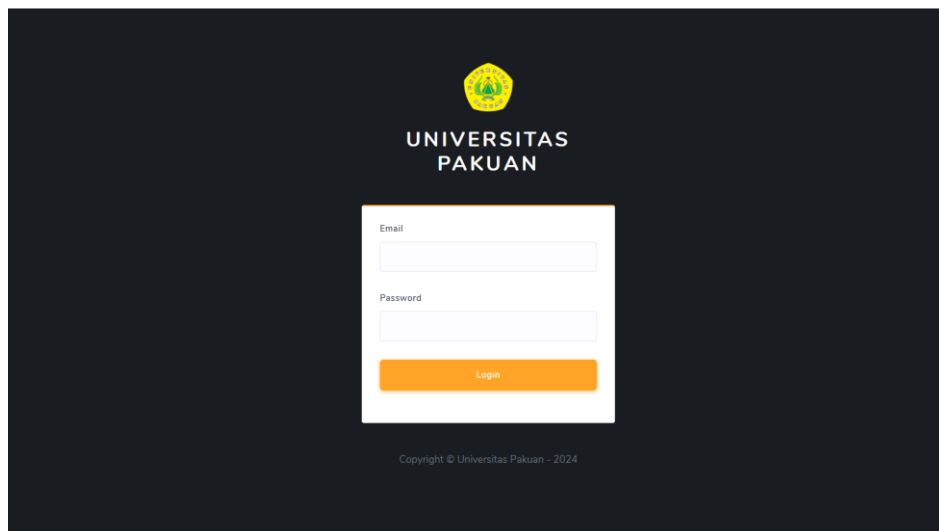
3. Halaman Uji Rekognisi



Gambar 81. Halaman Uji Rekognisi

Gambar 81 halaman uji coba rekognisi dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web*. Pada halaman ini, terdapat kotak pratinjau kamera yang digunakan untuk melakukan uji coba rekognisi wajah dan ada tabel untuk melakukan perulangan yang menampilkan uji coba rekognisi valid atau tidak.

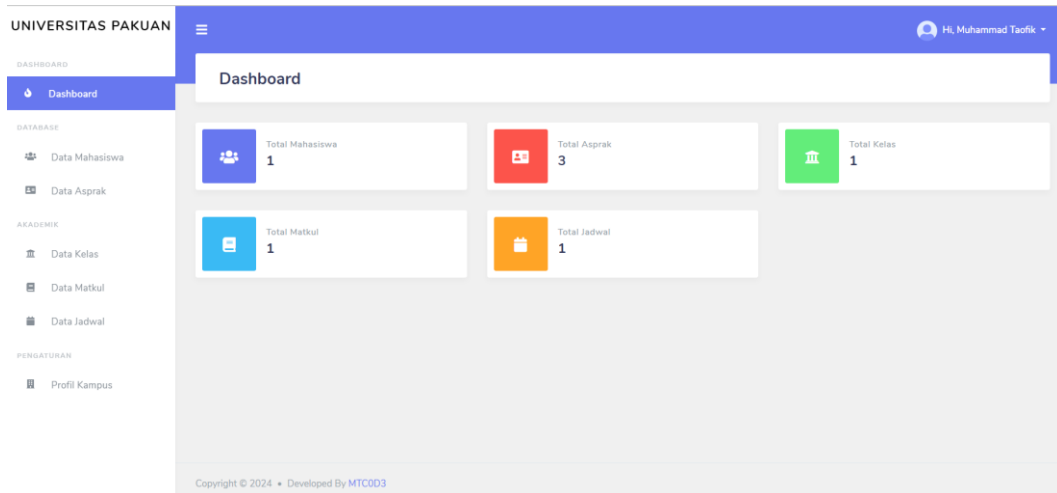
4. Halaman *Login*



Gambar 82. Halaman *Login*

Gambar 82 merupakan halaman *login* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web*. Pada halaman ini, terdapat *form* untuk memasukkan email dan password yang sudah terdaftar yang digunakan untuk mengakses halaman *dashboard*.

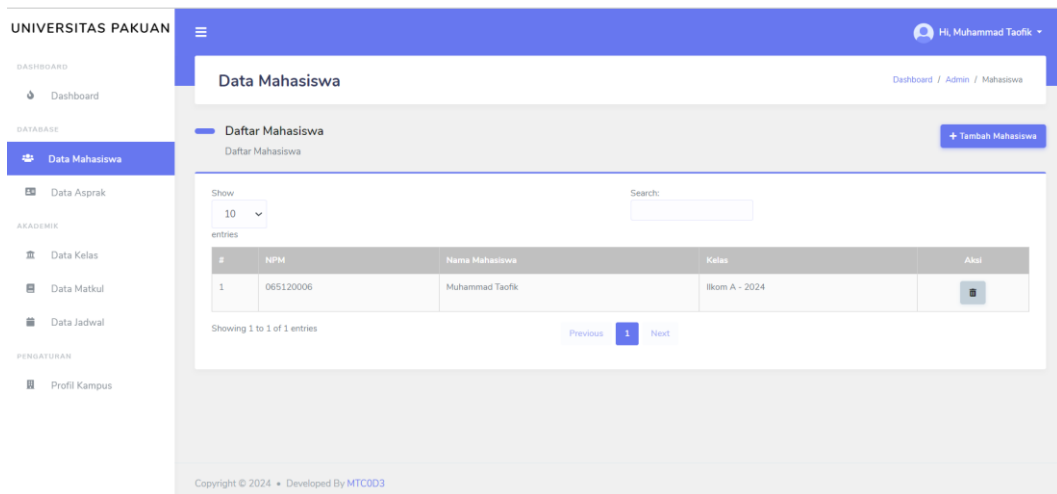
5. Halaman *Dashboard*



Gambar 83. Halaman *Dashboard*

Gambar 83 merupakan halaman *dashboard* dari Sistem Presensi Mahasiswa Dengan *Face Recognition* Dan *Location Based Service* Berbasis *Web* yang muncul pertama kali setelah admin melakukan *login*. Pada halaman ini, terdapat judul aplikasi pada bagian atas serta ringkasan jumlah data mahasiswa, asprak, kelas, matkul, dan jadwal yang tercatat di dalam *database* pada bagian bawahnya. Selain itu, pada bagian kiri terdapat *navbar* yang berisi daftar menu pada halaman *dashboard*. *Navbar* tersebut berperan untuk memudahkan admin ketika ingin berpindah ke halaman lain.

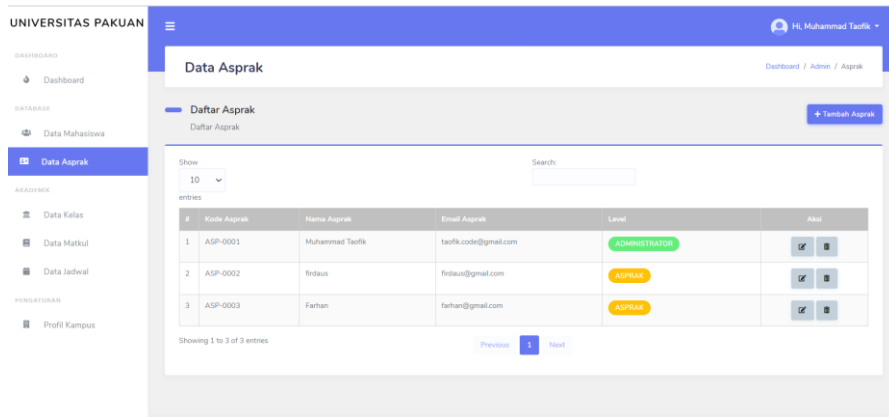
6. Halaman Mahasiswa



Gambar 84. Halaman Data Mahasiswa

Gambar 84 merupakan halaman Data Mahasiswa pada *dashboard* admin yang menampilkan daftar seluruh mahasiswa pada *database* dalam bentuk tabel. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data mahasiswa.

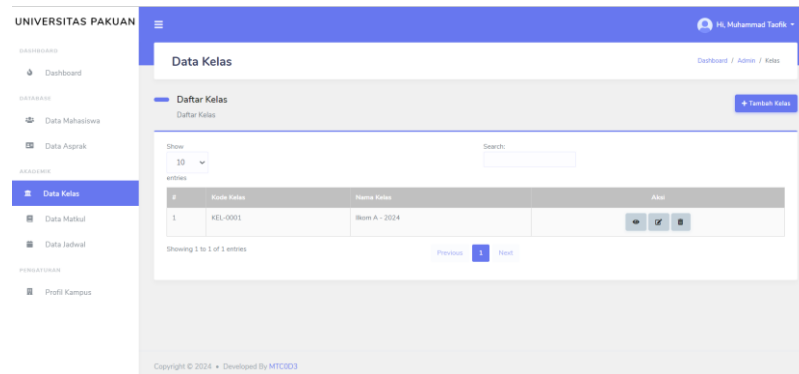
7. Halaman Asprak



Gambar 85. Halaman Data Asprak

Gambar 85 merupakan halaman Data Asprak pada *dashboard* admin yang menampilkan daftar seluruh asprak pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah asprak yang digunakan untuk menambah data asprak. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data asprak dan tombol *edit* untuk mengubah data asprak.

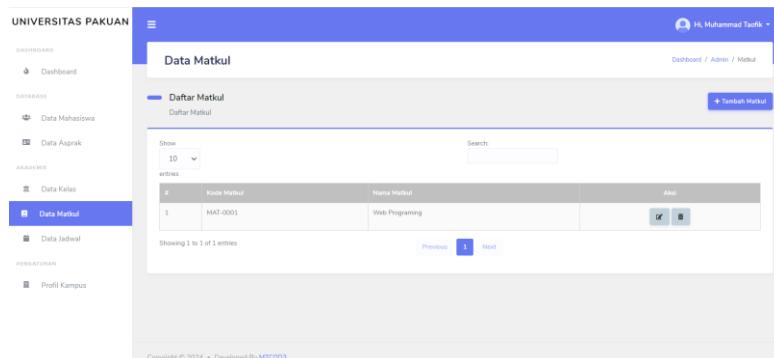
8. Halaman Kelas



Gambar 86. Halaman Data Kelas

Gambar 86 merupakan halaman Data Kelas pada *dashboard* admin yang menampilkan daftar seluruh kelas pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah kelas yang digunakan untuk menambah data kelas. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol melihat detail kelas, tombol untuk menghapus data kelas dan tombol *edit* untuk mengubah data kelas.

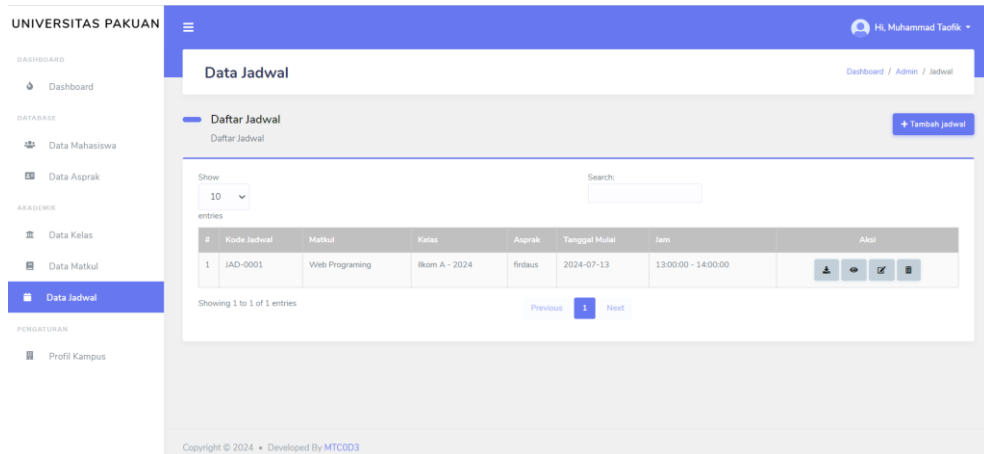
9. Halaman Matkul



Gambar 87. Halaman Data Matkul

Gambar 87 merupakan halaman Data Matkul pada *dashboard* admin yang menampilkan daftar seluruh matkul pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah matkul yang digunakan untuk menambah data matkul. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data matkul dan tombol *edit* untuk mengubah data matkul.

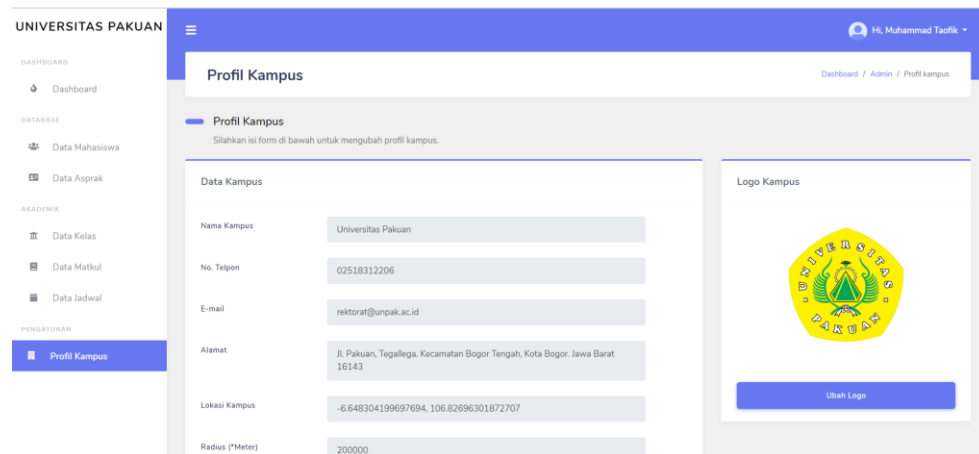
10. Halaman Jadwal



Gambar 88. Halaman Data Jadwal

Gambar 88 merupakan halaman Data Matkul pada *dashboard* admin yang menampilkan daftar seluruh matkul pada *database* dalam bentuk tabel. Pada bagian atas, ada tombol tambah matkul yang digunakan untuk menambah data matkul. Selain itu, tabel juga memiliki kolom aksi yang berisi tombol untuk menghapus data matkul dan tombol *edit* untuk mengubah data matkul.

11. Halaman Profil Kampus



Gambar 89. Halaman Profil Kampus

Gambar 89 merupakan halaman Profil Kampus pada *dashboard* admin yang menampilkan seluruh informasi kampus yang ada pada *database*. Pada bagian kanan, ada *form* logo untuk menampilkan logo kampus. Pada kedua *form*, terdapat tombol untuk mengubah informasi dan logo kampus.

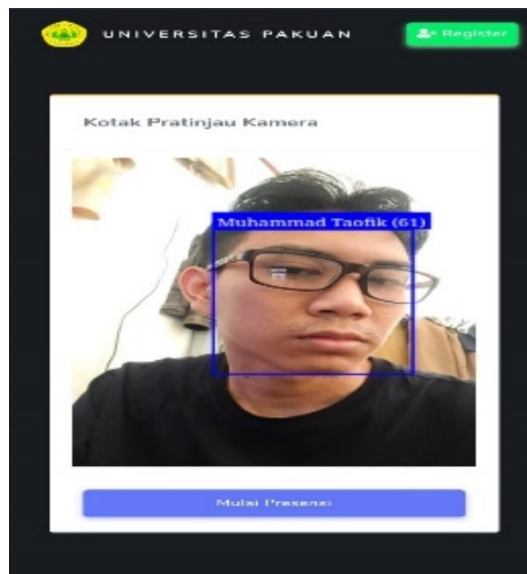
12. Laporan Presensi Mahasiswa

Laporan Presensi																
Kelas : Ilkom O - 2021																
Course : Optimasi																
#	NPM	Nama Mahasiswa	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	065121005	Wisnu Dwi Pradana	-	-	-	-	-	-	-	-	-	-	-	-	-	H
2	065121012	Muhammad Ali Gufron	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	065121016	Muhammad Syehan	-	-	-	-	-	-	-	-	-	-	-	-	-	H
4	065121017	Adiyuda pratama	-	-	-	-	-	-	-	-	-	-	-	-	-	H
5	065121023	Randy Apriliansyah	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	065121056	ALFISYAHIRIN RAMADHAN	-	-	-	-	-	-	-	-	-	-	-	-	-	H
7	065121105	M. Hanan Brian Prakoso	-	-	-	-	-	-	-	-	-	-	-	-	-	H
8	065121155	MUHAMAD ABDUL WAHAB	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Gambar 90. Laporan Presensi Mahasiswa

Gambar 90 merupakan laporan presensi mahasiswa dari laporan ini dapat di *download* dari halaman data jadwal sesuai jadwal yang dipilih.

13. Mahasiswa Terdaftar Menggunakan Kacamata



Gambar 91. Mahasiswa Terdaftar Menggunakan Kacamata

Gambar 91 merupakan tampilan dimana jarak perangkat pengguna dalam radius kampus yang sudah diatur sekitar 200 meter dan mahasiswa terdaftar dalam *database* tetapi terdapat kasus dimana mahasiswa menggunakan kacamata maka sistem akan memunculkan *bounding box* dengan label nama mahasiswa dan dapat melakukan presensi.

Lampiran 9. Prosedur Pengajian

Tabel 27. Test Case Skenario 1

No	Item Uji	Detail Pengujian
1	Halaman Utama	Menampilkan halaman utama
		Melakukan presensi dengan <i>face recognition</i>
2	Authentikasi	Melakukan register
		Melakukan login
		Melakukan logout
3	Halaman Dashboard	Menampilkan data dashboard
4	Halaman Mahasiswa	Menampilkan seluruh data mahasiswa
		Menghapus data mahasiswa
5	Halaman Asprak	Menampilkan seluruh data asprak
		Menambahkan data asprak
		Mengubah data asprak
		Menghapus data asprak
6	Halaman Kelas	Menampilkan seluruh data kelas
		Menambahkan data kelas
		Mengubah data kelas
		Menghapus data kelas
		Melihat detail kelas
7	Halaman Matkul	Menampilkan seluruh data matkul
		Menambahkan data matkul
		Mengubah data matkul
		Menghapus data matkul
8	Halaman Jadwal	Menampilkan seluruh data jadwal
		Menambahkan data jadwal
		Mengubah data jadwal
		Menghapus data jadwal
		Mendownload data presensi mahasiswa
9	Profil Kampus	Menampilkan seluruh data profil kampus
		Mengubah data profil kampus

Lampiran 10. Hasil Pengujian Fungsionalitas

a. Skenario 2: Memeriksa Fungsionalitas Autentikasi

Tabel 28. Test Case Skenario 2

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	Register dengan memasukan data yang benar	<ul style="list-style-type: none"> - Masuk halaman register - Input npm, nama dan kelas - Klik tombol register 	Data berhasil ditambahkan dan beralih ke halaman input wajah	Berhasil
Negatif	Register dengan memasukan data yang salah atau kosong	<ul style="list-style-type: none"> - Masuk halaman register - Input npm, nama dan kelas atau kosongkan form - Klik tombol register 	Data gagal ditambahkan dan menampilkan pesan kesalahan.	Berhasil
Positif	Register wajah	<ul style="list-style-type: none"> - Masuk halaman capture - Aktifkan kamera - Arahkan wajah ke kamera - Klik tombol pindai wajah - Klik tombol upload data 	Wajah terdeteksi dan tersimpan kemudian beralih kehalaman uji coba pengenalan	Berhasil
Negatif	Register wajah	<ul style="list-style-type: none"> - Masuk halaman capture - Aktifkan kamera - Arahkan wajah ke kamera - Klik tombol pindai wajah 	Tidak dapat melakukan pemindaian wajah dan tetap dihalaman capture	Berhasil
Positif	Uji coba <i>face recognition</i>	<ul style="list-style-type: none"> - Masuk halaman test-recognize - Aktifkan kamera - Arahkan wajah ke kamera - Klik tombol mulai uji coba pengenalan wajah 	Maka tabel akan terisi dengan nama yang benar dan valid kemudian beralih ke halaman utamas	Berhasil

		- klik tombol selesai		
Negatif	Uji coba <i>face recognition</i>	- Masuk halaman test-recognize - Matikan kamera - Klik tombol mulai uji coba pengenalan wajah	Maka proses uji coba tidak dapat dilakukan dan tetap dihalaman test-recognize	Berhasil
Positif	Login dengan email dan password yang benar	- Masuk halaman login - Input email dan password - Klik tombol login	Login berhasil dan beralih ke halaman dashboard sesuai level yang login	Berhasil
Negatif	Login dengan email dan password yang benar	- Masuk halaman login - Input email dan password - Klik tombol login	Login gagal dan muncul warning pop up yang menandakan bahwa email dan password salah	Berhasil
Negatif	Tidak melakukan login saat mengakses URL halaman dashboard	- Masuk halaman dashboard	Akses gagal dan dialihkan kembali ke halaman login	Berhasil
Positif	Logout	- Masuk halaman dashboard - Klik tombol logout - Klik ya pada confirm box	Token dihapus dan beralih ke halaman login	Berhasil
Negatif	Logout namun dibatalkan	- Masuk halaman dashboard - Klik tombol logout - Klik batalkan pada confirm box	Tetap berada pada halaman dashboard	Berhasil

b. Scenario 3: Memeriksa Fungsionalitas Halaman Dashboard

Tabel 29. Test Case Skenario 3

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	User berhasil login ke sistem	- Masuk halaman login - Input email dan password	Menampilkan halaman dashboard dan datanya sesuai	Berhasil

		- Klik tombol login	level akun yang login	
--	--	---------------------	-----------------------	--

c. Scenario 4: Memeriksa Fungsionalitas Halaman Mahasiswa

Tabel 30. Test Case Skenario 4

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	Menampilkan seluruh data mahasiswa	- Masuk halaman dashboard - Pilih menu data mahasiswa	Berhasil menampilkan seluruh data mahasiswa	Berhasil
Positif	Hapus data mahasiswa	- Masuk halaman dashboard - Pilih menu data mahasiswa - Klik tombol hapus - Klik ya pada confirm box	Berhasil menghapus data mahasiswa	Berhasil
Negatif	Hapus data mahasiswa namun dibatalkan	- Masuk halaman dashboard - Pilih menu data mahasiswa - Klik tombol hapus - Klik batalkan pada confirm box	Data mahasiswa tidak dihapus	Berhasil

d. Scenario 5: Memeriksa Fungsionalitas Halaman Asprak

Tabel 31. Test Case Skenario 5

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	Menampilkan seluruh data asprak	- Masuk halaman dashboard - Pilih menu data asprak	Berhasil menampilkan seluruh data asprak	Berhasil
Positif	Tambah data asprak	- Masuk halaman dashboard - Pilih menu	Berhasil menambah data asprak	Berhasil

		<ul style="list-style-type: none"> - data asprak - Klik tombol tambah asprak - Input data asprak baru - Klik tombol simpan data 		
Negatif	Tambah data asprak tanpa input data	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data asprak - Klik tombol tambah asprak - Klik tombol simpan data 	Muncul alert dan data tidak ditambahkan	Berhasil
Positif	Edit data asprak	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data asprak - Klik tombol edit asprak - Input perubahan data asprak - Klik tombol simpan perubahan 	Berhasil mengubah data asprak	Berhasil
Negatif	Edit data asprak namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data asprak - Klik tombol edit asprak - Klik tombol kembali 	Refresh halaman dan data tidak diperbarui	Berhasil
Positif	Hapus data asprak	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data asprak - Klik tombol hapus - Klik ya pada confirm box 	Berhasil menghapus data asprak	Berhasil

Negatif	Hapus data asprak namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data asprak - Klik tombol hapus - Klik batalkan pada confirm box 	Data asprak tidak dihapus	Berhasil
---------	------------------------------------	--	---------------------------	----------

e. Scenario 6: Memeriksa Fungsionalitas Halaman Kelas

Tabel 32. Test Case Skenario 6

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	Menampilkan seluruh data kelas	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data kelas 	Berhasil menampilkan seluruh data kelas	Berhasil
Positif	Tambah data kelas	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data kelas - Klik tombol tambah kelas - Input data kelas baru - Klik tombol simpan data 	Berhasil menambah data kelas	Berhasil
Negatif	Tambah data kelas tanpa input data	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data kelas - Klik tombol tambah kelas - Klik tombol simpan data 	Muncul alert dan data tidak ditambahkan	Berhasil
Positif	Edit data kelas	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data kelas - Klik tombol edit kelas - Input 	Berhasil mengubah data kelas	Berhasil

		perubahan data kelas - Klik tombol simpan perubahan		
Negatif	Edit data kelas namun dibatalkan	- Masuk halaman dashboard - Pilih menu data kelas - Klik tombol edit kelas - Klik tombol kembali	Refresh halaman dan data tidak diperbarui	Berhasil
Positif	Hapus data kelas	- Masuk halaman dashboard - Pilih menu data kelas - Klik tombol hapus - Klik ya pada confirm box	Berhasil menghapus data kelas	Berhasil
Negatif	Hapus data asprak namun dibatalkan	- Masuk halaman dashboard - Pilih menu data kelas - Klik tombol hapus - Klik batalkan pada confirm box	Data kelas tidak dihapus	Berhasil
Positif	Melihat detail kelas	- Masuk halaman dashboard - Pilih menu data kelas - Klik tombol detail	Menampilkan seluruh mahasiswa yang termasuk dalam kelas tersebut	Berhasil

f. Scenario 7: Memeriksa Fungsionalitas Halaman Matkul

Tabel 33. *Test Case* Skenario 7

Tipe	Aktivitas	Prosedur	Hasil	Status
------	-----------	----------	-------	--------

Positif	Menampilkan seluruh data matkul	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul 	Berhasil menampilkan seluruh data matkul	Berhasil
Positif	Tambah data matkul	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul - Klik tombol tambah matkul - Input data kelas baru - Klik tombol simpan data 	Berhasil menambah data matkul	Berhasil
Negatif	Tambah data matkul tanpa input data	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul - Klik tombol tambah matkul - Klik tombol simpan data 	Muncul alert dan data tidak ditambahkan	Berhasil
Positif	Edit data matkul	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul - Klik tombol edit matkul - Input perubahan data matkul - Klik tombol simpan perubahan 	Berhasil mengubah data matkul	Berhasil
Negatif	Edit data matkul namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul - Klik tombol edit matkul - Klik tombol kembali 	Refresh halaman dan data tidak diperbarui	Berhasil

Positif	Hapus data matkul	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul - Klik tombol hapus - Klik ya pada confirm box 	Berhasil menghapus data matkul	Berhasil
Negatif	Hapus data matkul namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data matkul - Klik tombol hapus - Klik batalkan pada confirm box 	Data matkul tidak dihapus	Berhasil

g. Scenario 8: Memeriksa Fungsionalitas Halaman Jadwal

Tabel 34. Test Case Skenario 8

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	Menampilkan seluruh data jadwal	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal 	Berhasil menampilkan seluruh data jadwal	Berhasil
Positif	Tambah data jadwal	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol tambah jadwal - Input data jadwal baru - Klik tombol simpan data 	Berhasil menambah data jadwal	Berhasil
Negatif	Tambah data jadwal tanpa input data	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol tambah jadwal - Klik tombol simpan data 	Muncul alert dan data tidak ditambahkan	Berhasil

Positif	Edit data jadwal	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol edit jadwal - Input perubahan data jadwal - Klik tombol simpan perubahan 	Berhasil mengubah data jadwal	Berhasil
Negatif	Edit data jadwal namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol edit jadwal - Klik tombol kembali 	Refresh halaman dan data tidak diperbarui	Berhasil
Positif	Hapus data jadwal	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol hapus - Klik ya pada confirm box 	Berhasil menghapus data jadwal	Berhasil
Negatif	Hapus data jadwal namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol hapus - Klik batalkan pada confirm box 	Data jadwal tidak dihapus	Berhasil
Positif	Melihat detail jadwal	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol detail 	Menampilkan seluruh presensi mahasiswa pada jadwal tersebut	Berhasil

Positif	Mendownload laporan presensi sesuai jadwal	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu data jadwal - Klik tombol download 	Mendownload laporan presensi sesuai dengan jadwal	Berhasil
---------	--	---	---	----------

h. Scenario 9: Memeriksa Fungsionalitas Halaman Profil Kampus

Tabel 35. Test Case Skenario 9

Tipe	Aktivitas	Prosedur	Hasil	Status
Positif	Menampilkan seluruh informasi profil kampus	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu profil kampus 	Berhasil menampilkan seluruh informasi profil kampus	Berhasil
Positif	Edit profil kampus	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu profil kampus - Klik tombol edit profil kampus - Input perubahan profil kampus - Klik tombol simpan perubahan 	Berhasil mengubah profil kampus	Berhasil
Negatif	Edit profil kampus namun dibatalkan	<ul style="list-style-type: none"> - Masuk halaman dashboard - Pilih menu profil kampus - Klik tombol edit profil kampus - Klik tombol kembali 	Refresh halaman dan data tidak diperbarui	Berhasil