

Implementasi Convolutional Neural Network (CNN) untuk Sistem Presensi Mahasiswa Berbasis Pengenalan Wajah

Fahmi Yusuf¹⁾, Iwan Lesmana²⁾, Hegar Paningkat Bagja³⁾

*Fakultas Ilmu Komputer, Universitas Kuningan
Jalan Cut Nyak Dhien No.36 A Cijoho Kuningan Jawa Barat 45513 Telepon (0232) 2875097
fahmionline@uniku.ac.id¹, iwan.lesmana@uniku.ac.id², 20170810075@uniku.ac.id³
Corresponding Author : fahmionline@uniku.ac.id*

Abstrak

Sistem presensi memegang peran krusial dalam mendukung proses pembelajaran dan administrasi akademik di berbagai institusi, termasuk Fakultas Ilmu Komputer Universitas Kuningan. Saat ini, sistem presensi yang digunakan masih mengandalkan metode konvensional, yaitu lembar presensi berbasis *hard copy* yang ditandatangani oleh mahasiswa. Metode ini memiliki beberapa kelemahan, seperti waktu yang lama dalam merekap data ke sistem akademik, potensi manipulasi tanda tangan, serta risiko kehilangan atau kerusakan dokumen. Untuk mengatasi masalah tersebut, penelitian ini mengusulkan penerapan teknologi *face recognition* berbasis *Convolutional Neural Network* (CNN). CNN adalah pengembangan dari *Multi-layer Perceptron* (MLP) yang dirancang khusus untuk mengolah data dua dimensi, seperti citra wajah. Penelitian ini menggunakan model pengembangan prototipe yang terdiri dari tiga tahap utama: (1) analisis kebutuhan pengguna, (2) perancangan *mockup*, dan (3) pengujian rancangan. Proses identifikasi wajah menghasilkan deskripsi fitur yang merepresentasikan karakteristik wajah yang terdeteksi. Deskripsi fitur ini kemudian dibandingkan dengan data wajah yang tersimpan dalam database untuk melakukan pencocokan. Hasil pengujian menggunakan metode *K-Fold* menunjukkan bahwa model yang diusulkan mencapai akurasi sebesar **92%**. Dengan demikian, sistem presensi berbasis *face recognition* ini diharapkan dapat meningkatkan efisiensi, mengurangi potensi kecurangan, dan menyediakan solusi presensi yang lebih modern dan terintegrasi bagi Fakultas Ilmu Komputer Universitas Kuningan.

Kata Kunci : Convolutional Neural Network (CNN); Face Recognition; Prototype;

Abstract

The attendance system plays a crucial role in supporting the learning process and academic administration in various institutions, including the Faculty of Computer Science, Kuningan University. Currently, the attendance system used still relies on conventional methods, namely hard copy-based attendance sheets signed by students. This method has several disadvantages, such as a long time in recapitulating data to the academic system, the potential for signature manipulation, and the risk of losing or damaging documents. To overcome these problems, this research proposes the application of face recognition technology based on Convolutional Neural Network (CNN). CNN is a development of Multi-layer Perceptron (MLP) specifically designed to process two-dimensional data, such as facial images. This research uses a prototype development model consisting of three main stages: (1) user needs analysis, (2) mock-up design, and (3) design testing. The face identification process produces a feature description that represents the characteristics of the detected face. This feature description is then compared with the face data stored in the database to perform matching. The test results using the K-Fold method show that the proposed model achieves an accuracy of 92%. Thus, this face recognition-based attendance system is expected to increase efficiency, reduce the potential for fraud, and provide a more modern and integrated attendance solution for the Faculty of Computer Science, Kuningan University.

Keywords: Convolutional Neural Network (CNN); Face Recognition; Prototype;

1. PENDAHULUAN

Sistem presensi telah menjadi komponen penting yang banyak digunakan di berbagai institusi, seperti sekolah, universitas, dan perusahaan. Menurut [1], Dekan Fakultas Ilmu Komputer Universitas Kuningan menyatakan bahwa kehadiran atau presensi memegang peran krusial dalam mendukung proses pembelajaran dan kegiatan akademik di fakultas tersebut. Saat ini, sistem presensi yang digunakan masih mengandalkan metode konvensional, yaitu menggunakan lembar presensi berbasis *hard copy* yang ditandatangani oleh mahasiswa dan dikumpulkan kembali setelah diisi. Proses pengolahan data absensi dilakukan secara manual dengan memasukkan data ke dalam sistem informasi akademik, yang kemudian dikelola melalui komputerisasi. Metode ini menimbulkan berbagai masalah, seperti lamanya waktu yang dibutuhkan untuk merekap data, potensi manipulasi tanda tangan oleh mahasiswa, serta kesulitan yang dihadapi oleh staf departemen dan dosen dalam mengevaluasi kehadiran mahasiswa.

Untuk mengatasi permasalahan tersebut, teknologi *face recognition* (pengenalan wajah) dapat menjadi solusi yang efektif. Teknologi ini bekerja dengan cara mengidentifikasi citra wajah dan menghasilkan deskripsi fitur yang merepresentasikan karakteristik wajah yang terdeteksi. Deskripsi fitur tersebut kemudian dibandingkan dengan data wajah yang tersimpan dalam database untuk melakukan proses pencocokan. Salah satu pendekatan yang digunakan dalam teknologi ini adalah *Convolutional Neural Network* (CNN), yang merupakan pengembangan dari metode *Multi-layer Perceptron* (MLP). CNN dirancang khusus untuk mengolah data dua dimensi, seperti citra, dan mampu mencapai tingkat akurasi klasifikasi yang tinggi meskipun dengan proses *preprocessing* dan segmentasi yang minimal [2].

Penelitian bertujuan untuk mengembangkan sistem presensi yang lebih efisien, akurat, dan bebas dari masalah yang sering muncul pada metode konvensional. Dengan memanfaatkan teknologi *face recognition* dan algoritma CNN, diharapkan sistem presensi yang dihasilkan dapat meningkatkan tingkat akurasi, mengurangi potensi kecurangan, serta menyediakan solusi yang lebih modern dan terintegrasi bagi Fakultas Ilmu Komputer Universitas Kuningan.

2. METODE PENELITIAN

2.1. Metode Pengumpulan Data

a. Observasi

Pada tahap observasi, penulis melakukan pengamatan langsung terhadap proses pengisian presensi kehadiran di Fakultas Ilmu Komputer Universitas Kuningan. Melalui observasi ini, penulis mengamati sistem presensi yang sedang berjalan di fakultas tersebut. Hasil observasi memberikan gambaran nyata tentang praktik dan tantangan yang dihadapi dalam sistem presensi konvensional yang digunakan saat ini. Creswell mengungkapkan pendapatnya bahwa observasi adalah proses pemerolehan data dari tangan pertama, dengan cara melakukan pengamatan orang serta lokasi dilakukannya penelitian [3].

Juga dalam observasi dilakukan pengumpulan dataset wajah untuk training dilakukan dengan teknik seperti web scraping, akuisisi langsung melalui kamera. Data dalam keadaan beragam dalam pencahayaan, ekspresi, dan sudut wajah untuk meningkatkan akurasi model. Proses ini mencakup anotasi wajah, pembersihan data, serta memastikan kepatuhan terhadap regulasi privasi.

b. Wawancara

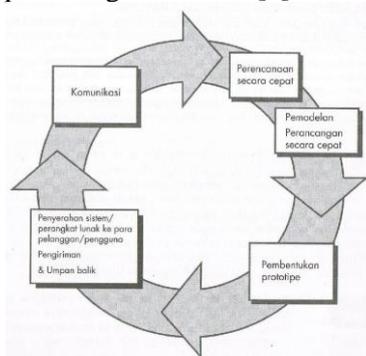
Setelah melakukan observasi, penulis melanjutkan dengan metode wawancara. Wawancara dilakukan secara langsung dengan Dekan Fakultas Ilmu Komputer Universitas Kuningan untuk memperoleh informasi yang relevan mengenai sistem presensi dan kebutuhan fakultas. Wawancara kualitatif terjadi ketika seorang peneliti mengajukan pertanyaan umum terbuka kepada satu atau lebih partisipan dan mencatat jawaban mereka [3].

c. Studi Pustaka

Pada tahap studi pustaka, penulis mengumpulkan data dan informasi dari berbagai sumber literatur, seperti buku, jurnal, dan e-book, yang berkaitan dengan topik penelitian [4]. Fokus studi pustaka meliputi Algoritma *Convolutional Neural Network* (CNN), teknologi *Face Recognition*, serta sistem presensi kehadiran. Studi pustaka ini dilakukan untuk memperoleh landasan teoritis dan referensi yang mendukung perancangan dan pengembangan aplikasi presensi berbasis pengenalan wajah.

2.2. Metode Pengembangan Perangkat Lunak

Model prototipe dapat digunakan untuk menjembatani kesenjangan pemahaman antara pelanggan dan pengembang perangkat lunak terkait aspek teknis. Model ini membantu memperjelas spesifikasi kebutuhan yang diinginkan oleh pelanggan, sehingga pengembang dapat merancang solusi yang sesuai. Dengan demikian, dapat disimpulkan bahwa model prototipe merupakan salah satu model pengembangan perangkat lunak yang memfasilitasi komunikasi dan kolaborasi antara pengembang dan objek penelitian. Proses ini meliputi beberapa tahapan utama, yaitu mendengarkan kebutuhan pelanggan atau melakukan analisis kebutuhan, membuat rancangan (*mockup*), dan melakukan pengujian terhadap rancangan tersebut.[5]



Gambar 1. Ilustrasi Model Pengembangan *Prototype* [6]

Berikut adalah tahapan-tahapan dalam metode prototipe :

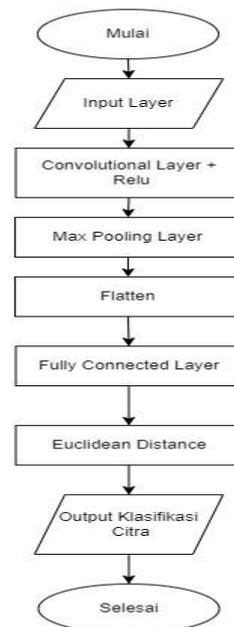
1. **Komunikasi dan Pengumpulan Data Awal**
Tahap ini melibatkan analisis kebutuhan pengguna melalui komunikasi intensif untuk memahami secara mendalam apa yang diharapkan oleh pengguna.
2. **Desain Cepat (*Quick Design*)**
Pada tahap ini, desain awal dibuat secara umum sebagai dasar untuk pengembangan lebih lanjut. Desain ini bersifat sementara dan akan disempurnakan pada tahap berikutnya.
3. **Pembentukan Prototipe**
Prototipe awal dibangun berdasarkan desain cepat. Prototipe ini kemudian diuji dan disempurnakan untuk memastikan bahwa fitur dan fungsionalitasnya sesuai dengan kebutuhan pengguna.
4. **Evaluasi Prototipe**

Prototipe yang telah dibentuk dievaluasi oleh pengguna dan tim pengembang. Hasil evaluasi digunakan untuk memperhalus analisis kebutuhan dan mengidentifikasi area yang perlu diperbaiki.

5. **Perbaiki Prototipe**
Berdasarkan hasil evaluasi, prototipe diperbaiki dan disempurnakan. Tahap ini bertujuan untuk menghasilkan tipe yang lebih mendekati produk akhir yang diinginkan.
6. **Produksi Akhir**
Setelah prototipe disetujui dan dianggap memenuhi kebutuhan pengguna, produk akhir diproduksi secara menyeluruh dan siap digunakan oleh pengguna.

2.3. Metode Pemecahan Masalah

Metode CNN (*Convolutional Neural Network*) adalah diantara metode pembelajaran mesin yang menggunakan *neural network* dan biasa digunakan untuk mengolah data citra. Data citra yang digunakan sebagai data masukan, biasa memiliki ukuran Panjang \times Lebar \times channel warna.



Gambar 2. Flowchart Metode *Convolutional Neural Network* [5]

1. **Input Layer**
Pada CNN, lapisan input berfungsi sebagai representasi dari data masukan atau gambar yang akan diproses oleh jaringan. Lapisan ini menerima gambar atau data input dalam bentuk matriks piksel untuk diolah lebih lanjut.
2. **Convolutional Layer**

Lapisan ini berperan utama dalam mengekstraksi fitur-fitur penting dari data input, yang biasanya berupa gambar. Secara prinsip, lapisan konvolusi melakukan operasi konvolusi pada data input menggunakan satu atau lebih filter (*kernel*) untuk menghasilkan peta fitur. Untuk keperluan perhitungan, penulis memutuskan untuk membatasi ukuran input citra dari 224x224 piksel menjadi 6x6 piksel. Citra tersebut diambil nilai komponen warna Red, Green, dan Blue (RGB), dan berikut adalah nilai dari masing-masing komponen tersebut.

Chanel Red

0	0	0	0	0	0
0	199	200	200	199	0
0	199	200	200	199	0
0	199	199	199	199	0
0	198	199	199	198	0
0	0	0	0	0	0

Chanel Green

0	0	0	0	0	0
0	148	149	149	148	0
0	148	149	149	148	0
0	148	148	148	148	0
0	147	148	148	147	0
0	0	0	0	0	0

Chanel Blue

0	0	0	0	0	0
0	131	132	132	131	0
0	131	132	132	131	0
0	131	131	131	131	0
0	130	131	131	130	0
0	0	0	0	0	0

Gambar 3. Matriks *Chanel Red, Green, dan Blue*

Filter dengan ukuran 3x3 yang akan digunakan dalam proses konvolusi dapat dilihat pada Gambar 4.

1	0	-1
1	0	-1
1	0	-1

Gambar 4. Filter 3x3

Selanjutnya, dilakukan proses perhitungan konvolusi. Proses ini melibatkan perkalian antara kernel berukuran 3x3 dengan matriks channel Red berukuran 6x6 secara bertahap hingga mencapai akhir matriks. Tahapan perhitungan ini dapat dilihat dari area yang telah ditandai.

a. *Chanel Red*

0	0	0	0	0	0
0	199	200	200	199	0
0	199	200	200	199	0
0	199	199	199	199	0
0	198	199	199	198	0
0	0	0	0	0	0

1	0	-1
1	0	-1
1	0	-1

Proses 1:
 $(1x0) + (0x0) + (-1x0) + (1x0) + (0x199) + (-1x200) + (1x0) + (0x199) + (-1x200) = -400$

Proses 2
 $(1x0) + (0x0) + (-1x0) + (1x199) + (0x200) + (-1x200) + (1x199) + (0x200) + (-1x200) = -2$

Proses 3
 $(1x0) + (0x0) + (-1x0) + (1x200) + (0x200) + (-1x199) + (1x200) + (0x200) + (-1x199) = 2$

Proses 4
 $(1x0) + (0x0) + (-1x0) + (1x200) + (0x199) + (-1x0) + (1x200) + (0x199) + (-1x0) = 400$

·
·
·

Proses 16
 $(1x199) + (0x199) + (-1x0) + (1x199) + (0x198) + (-1x0) + (1x0) + (0x0) + (-1x0) = 398$

Hasil :

-400	-2	2	400
-599	-2	2	599
-599	-2	2	599
-398	-1	1	398

b. Chanel Green

0	0	0	0	0	0
0	148	149	149	148	0
0	148	149	149	148	0
0	148	148	148	148	0
0	147	148	148	147	0
0	0	0	0	0	0

Proses 1

$$(1x0) + (0x0) + (-1x0) + (1x0) + (0x148) + (-1x149) + (1x0) + (0x148) + (-1x149) = -298$$

Proses 2

$$(1x0) + (0x0) + (-1x0) + (1x148) + (0x149) + (-1x149) + (1x148) + (0x149) + (-1x149) = -2$$

Proses 3

$$(1x0) + (0x0) + (-1x0) + (1x149) + (0x149) + (-1x148) + (1x149) + (0x149) + (-1x148) = 2$$

Proses 4

$$(1x0) + (0x0) + (-1x0) + (1x149) + (0x148) + (-1x0) + (1x149) + (0x148) + (-1x0) = 298$$

·
·
·

Proses 16

$$(1x148) + (0x148) + (-1x0) + (1x148) + (0x147) + (-1x0) + (1x0) + (0x0) + (-1x0) = 296$$

Hasil :

-298	-2	2	298
-446	-2	2	446
-445	-2	2	445
-296	-1	1	296

c. Chanel Blue

0	0	0	0	0	0
0	131	132	132	131	0
0	131	132	132	131	0
0	131	131	131	131	0
0	130	131	131	130	0
0	0	0	0	0	0

Proses 1

$$(1x0) + (0x0) + (-1x0) + (1x0) + (0x131) + (-1x132) + (1x0) + (0x131) + (-1x132) = -264$$

Proses 2

$$(1x0) + (0x0) + (-1x0) + (1x131) + (0x132) + (-1x132) + (1x131) + (0x132) + (-1x132) = -2$$

Proses 3

$$(1x0) + (0x0) + (-1x0) + (1x132) + (0x132) + (-1x131) + (1x132) + (0x132) + (-1x131) = 12$$

Proses 4

$$(1x0) + (0x0) + (-1x0) + (1x132) + (0x131) + (-1x0) + (1x132) + (0x131) + (-1x0) = 264$$

·
·
·

Proses 16

$$(1x131) + (0x131) + (-1x0) + (1x131) + (0x130) + (-1x0) + (1x0) + (0x0) + (-1x0) = 262$$

Hasil :

-264	-2	2	264
-395	-2	2	395
-394	-2	2	394
-262	-1	1	262

Setelah proses konvolusi dilakukan pada setiap layer (Red, Green, dan Blue), hasilnya kemudian dijumlahkan untuk memperoleh output akhir:

-962	-6	6	962
-1440	-6	6	1440
-1438	-6	6	1438
-956	-3	3	956

3.

3. Aktifasi ReLu

Dengan menggunakan fungsi aktivasi ReLU, setiap nilai negatif akan diubah menjadi 0, sehingga menghasilkan output konvolusi sebagai berikut:

0	0	6	962
0	0	6	1440
0	0	6	1438
0	0	3	956

4. Max Pooling

Matriks hasil konvolusi tersebut akan menjadi input untuk *pooling layer* dengan menerapkan metode *max pooling* menggunakan *kernel size 2x2*. Proses *max pooling* dapat dilihat pada ilustrasi berikut:

0	0	6	962
0	0	6	1440
0	0	6	1438
0	0	3	956

0	0	6	962
0	0	6	1440
0	0	6	1438
0	0	3	956

0	0	6	962
0	0	6	1440
0	0	6	1438
0	0	3	956

0	0	6	962
0	0	6	1440
0	0	6	1438
0	0	3	956

Hasil :

0	1440
0	1438

5. Flatten Layer

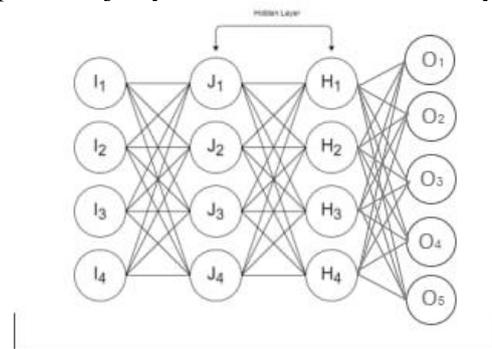
Selanjutnya, dilakukan proses *flatten layer*, yaitu tahapan mengubah matriks menjadi bentuk satu dimensi. Hasil proses ini dapat dilihat pada matriks berikut:

0	1440
0	1438

0
1440
0
1438

6. Fully Connected Layer

Output dari tahapan ini adalah vektor satu dimensi yang akan digunakan sebagai input untuk tahapan selanjutnya. Berikut adalah ilustrasinya:



Gambar 5. Fully connected layer

Nilai output dari proses *flatten* akan diproses lebih lanjut melalui *fully connected layer*. Berikut adalah contoh perhitungannya:

$$J1 = (0 \cdot 0,1) + (1440 \cdot 0,1) + (0 \cdot 0,1) + (1438 \cdot 0,1) = 0 + 144 + 0 + 143,8 = 287,3$$

$$J2 = (0 \cdot 0,4) + (1440 \cdot 0,4) + (0 \cdot 0,4) + (1438 \cdot 0,4) = 0 + 576 + 0 + 575,2 = 1151,2$$

$$J3 = (0 \cdot 0,5) + (1440 \cdot 0,5) + (0 \cdot 0,5) + (1438 \cdot 0,5) = 0 + 720 + 0 + 719 = 1439$$

$$J4 = (0 \cdot 0,3) + (1440 \cdot 0,3) + (0 \cdot 0,3) + (1438 \cdot 0,3) = 0 + 432 + 0 + 431,4 = 863,4$$

Hidden layer pertama diilustrasikan dengan neuron J1, J2, J3, dan J4. Setiap neuron tersebut dikalikan dengan *weight* yang berbeda, menghasilkan nilai J1 = 287,3; J2 = 1151,2; J3 = 1439; dan J4 = 863,4. Selanjutnya, setiap neuron dari J1, J2, J3, dan J4 akan dikalikan lagi dengan *weight* yang berbeda untuk menghasilkan nilai H1, H2, H3, dan H4 sebagai *hidden layer* berikutnya.

$$H1 = (287,3 \cdot 0,2) + (1151,2 \cdot 0,2) + (1439 \cdot 0,2) + (863,4 \cdot 0,2) = 57,46 + 230,24 + 287,8 + 172,68 = 748,18$$

$$H2 = (287,3 \cdot 0,3) + (1151,2 \cdot 0,3) + (1439 \cdot 0,3) + (863,4 \cdot 0,3) = 86,19 + 345,36 + 431,7 + 259,02 = 1122,27$$

$$\begin{aligned}
 \mathbf{H3} &= (287,3 \cdot 0,4) + (1575 \cdot 0,4) + (1007 \cdot 0,4) + (863,4 \cdot 0,4) \\
 &= 114,92 + 630 + 402,8 + 345,36 = 1493,08 \\
 \mathbf{H4} &= (287,3 \cdot 0,5) + (1575 \cdot 0,5) + (1007 \cdot 0,5) + (863,4 \cdot 0,5) \\
 &= 143,65 + 787,5 + 503,5 + 431,7 = 1866,35
 \end{aligned}$$

Setiap neuron pada H1, H2, H3, dan H4 akan dikalikan dengan *weight* yang berbeda-beda untuk menghasilkan nilai O1, O2, O3, O4, dan O5 sebagai *hidden layer* berikutnya.

$$\begin{aligned}
 \mathbf{O1} &= (745,68 \cdot 0,1) + (1119,99 \cdot 0,1) + (1493,08 \cdot 0,1) + (1866,35 \cdot 0,1) = 74,568 + 111,999 + 149,308 + 186,635 = 522,528 \\
 \mathbf{O2} &= (745,68 \cdot 0,3) + (1119,99 \cdot 0,3) + (1493,08 \cdot 0,3) + (1866,35 \cdot 0,3) = 223,758 + 335,997 + 447,924 + 559,905 = 1567,584 \\
 \mathbf{O3} &= (745,68 \cdot 0,4) + (1119,99 \cdot 0,4) + (1493,08 \cdot 0,4) + (1866,35 \cdot 0,4) = 298,272 + 447,996 + 597,232 + 746,54 = 2090,04 \\
 \mathbf{O4} &= (745,68 \cdot 0,2) + (1119,99 \cdot 0,2) + (1493,08 \cdot 0,2) + (1866,35 \cdot 0,2) = 149,136 + 223,998 + 298,796 + 373,27 = 1045,2 \\
 \mathbf{O5} &= (745,68 \cdot 0,5) + (1119,99 \cdot 0,5) + (1493,08 \cdot 0,5) + (1866,35 \cdot 0,5) = 372,84 + 559,995 + 746,54 + 933,175 = 2612,55
 \end{aligned}$$

Tahap berikutnya adalah proses klasifikasi menggunakan perhitungan *euclidean distance*. Metode ini digunakan untuk membandingkan vektor fitur wajah yang dihasilkan dari *fully connected layer* dengan vektor fitur wajah yang tersimpan dalam database.

7. Euclidean Distance

Tahap akhir dalam model CNN adalah klasifikasi menggunakan *Euclidean Distance* yang merupakan metode untuk mengukur jarak antara dua titik dalam ruang vektor. Semakin dekat dan mirip dua titik tersebut, semakin kecil nilai jaraknya.

Berikut adalah rumus *Euclidean Distance*:

$$d = \sqrt{(q_1 - p_1)^2 + \dots + (q_n - p_n)^2}$$

Keterangan:

- (q_1, p_1) = koordinat atau nilai dari dimensi ke-*i* pada vektor.
- n = jumlah dimensi atau panjang vektor.
- d = jarak *Euclidean Distance*.

Contoh perhitungan :

Citra 1	Citra 2
522,528	525,390
1567,584	1569,520
2090,04	2093,01
1045,2	1047,4
2612,55	2618,42

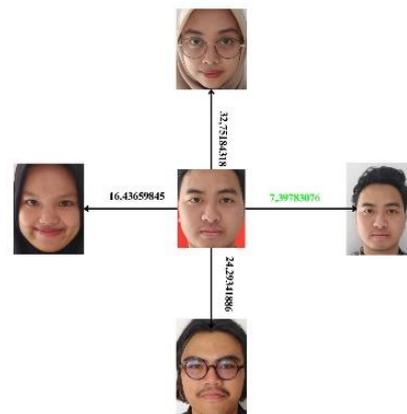
p = Citra 1 nilai vektor fitur wajah dari hasil proses *fully connected layer*

q = Citra 2 nilai vektor fitur wajah yang tersimpan di database

$$\begin{aligned}
 d &= \sqrt{[(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2 + (q_4 - p_4)^2]} \\
 d &= \sqrt{[(525,390 - 522,528)^2 + (1569,520 - 1567,584)^2 + (2093,01 - 2090,04)^2 + (1047,4 - 1045,2)^2 + (2618,42 - 2612,55)^2]} \\
 d &= \sqrt{[(2,862)^2 + (1,936)^2 + (2,97)^2 + (2,2)^2 + (5,87)^2]} \\
 d &= \sqrt{54,7279} \\
 d &= 7,39783076
 \end{aligned}$$

Hasil perhitungan *euclidean distance* yang diperoleh adalah **7,39783076**. Nilai ini kemudian dibandingkan dengan nilai fitur wajah dari citra lain yang telah tersimpan dalam database.

Berikut adalah contoh visualisasi citra wajah input yang dihitung jaraknya dengan citra wajah dalam database. Angka yang ditandai dengan warna hijau menunjukkan jarak terdekat dibandingkan dengan citra wajah lainnya yang ada dalam database.



Gambar 6. Contoh perbandingan *euclidean distance*

3. HASIL DAN PEMBAHASAN

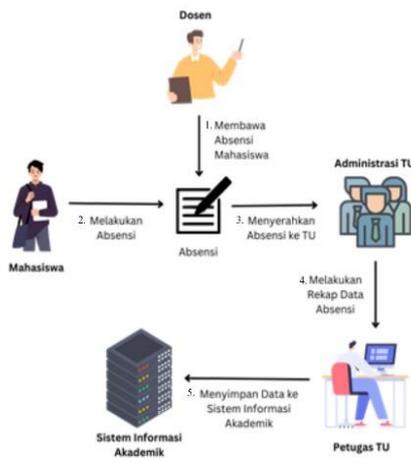
3.1 Perancangan

Model Rich Picture adalah teknik visual dalam perancangan sistem yang digunakan

untuk menggambarkan hubungan, proses, dan permasalahan dalam suatu sistem secara intuitif. Dengan simbol, gambar, dan teks, *Rich Picture* membantu memahami kompleksitas sistem, mengidentifikasi aktor utama, serta menemukan solusi sebelum masuk ke tahap perancangan teknis yang lebih rinci.

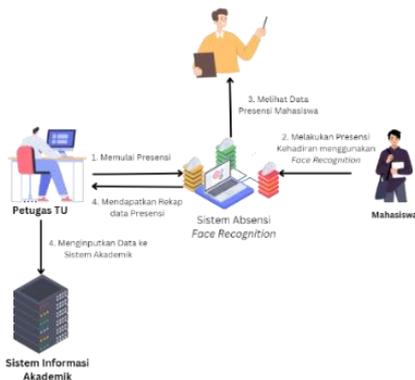
Dibawah ini digambarkan analisis sistem berjalan dan sistem usulan, use case serta diagram aktifitas dan class diagram (diwakili satu kegiatan):

1. Analisis Sistem Yang Berjalan



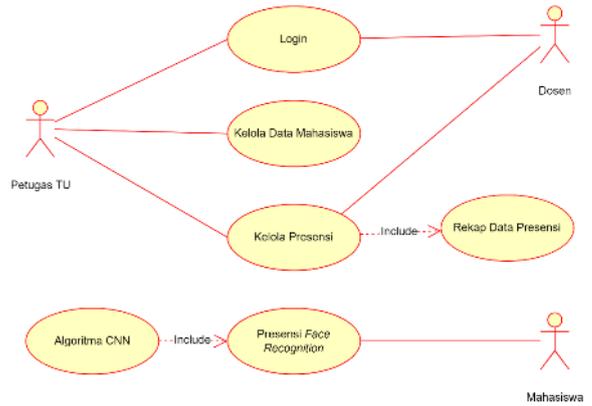
Gambar 7. Rich Picture system berjalan.

2. Analisis Sistem Yang Diusulkan



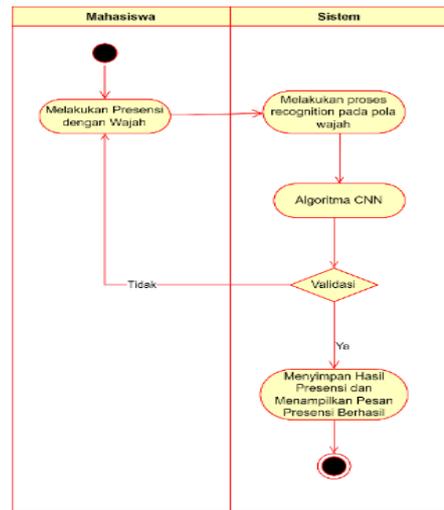
Gambar 8. Rich Picture Sistem usulan

3. Use Case Diagram



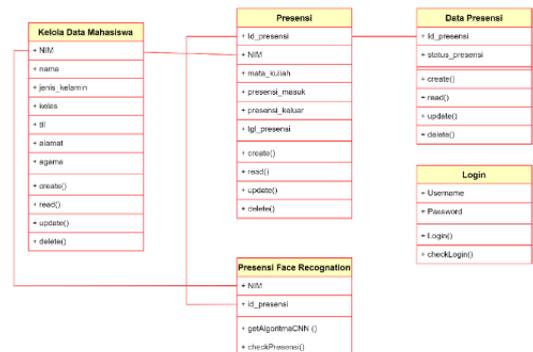
Gambar 9. Use Case Diagram

4. Diagram Aktifitas Presensi Face Recognition



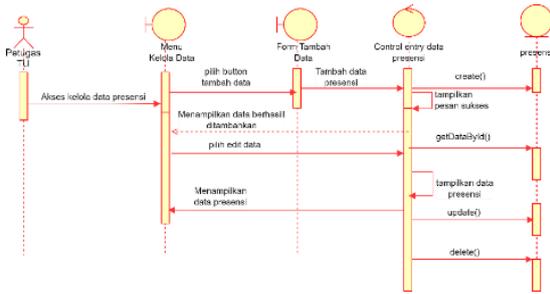
Gambar 10. Diagram Aktifitas Presensi Face Recognition

5. Class Diagram

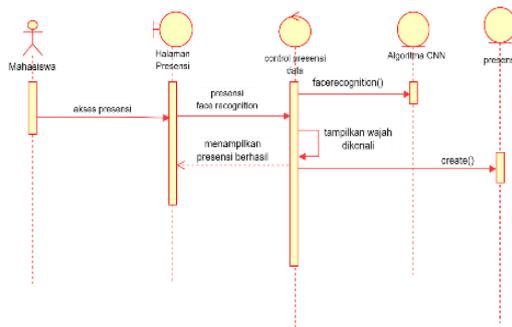


Gambar 10. Class Diagram

6. Sequence Diagram Kelola Presensi



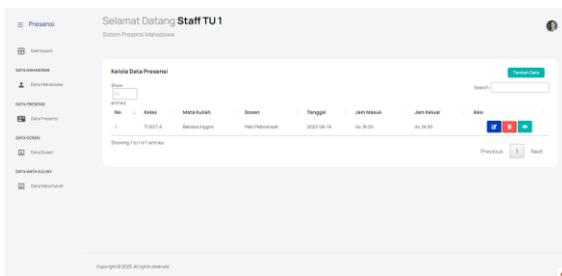
Gambar 11. Sequence Diagram Kelola Presensi



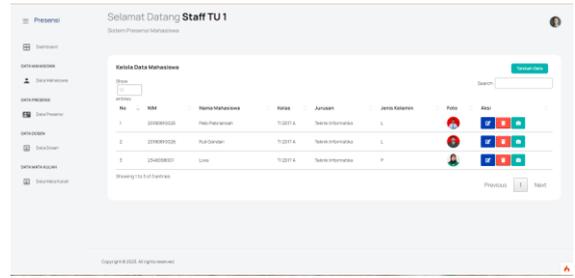
Gambar 13. Sequence Diagram Presensi Face Recognition

3.2 Implementasi

Implementasi adalah tahap merealisasikan rancangan sistem ke dalam lingkungan nyata. Proses ini mencakup instalasi, konfigurasi, pengujian, serta pelatihan pengguna untuk memastikan sistem berjalan sesuai kebutuhan. Di bawah ini adalah tampilan dari aplikasi yang telah diimplementasikan.



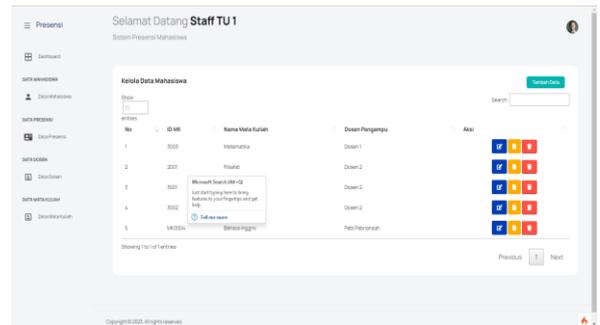
Gambar 14. Implementasi Antarmuka Data Presensi



Gambar 15. Implementasi Antarmuka Data Mahasiswa



Gambar 16. Implementasi Antarmuka Data Dosen



Gambar 17. Implementasi Antarmuka Data Matakuliah

3.3 Pengujian Pengujian K-Fold Cross Validation

Pada skenario pengujian, digunakan metode *K-Folds Cross Validation*, yaitu teknik untuk mengevaluasi keakuratan model yang dibangun berdasarkan dataset tertentu. Pengujian ini menggunakan nilai $K=4$, yang menunjukkan jumlah bagian dataset. Dataset dibagi menjadi 4 bagian, dan pengujian dilakukan sebanyak 4 kali. Pada setiap iterasi, satu bagian dataset digunakan sebagai data uji, sedangkan tiga bagian lainnya digunakan sebagai data latih, dan proses ini dilakukan secara bergantian..

Epoch : 5
Learning rate : 0.001
Drop out : 0,5

Tabel 1. Tabel Percobaan Learning Rate

Fold	Model Mahasiswa	
	Akurasi	Validation
1	60,85%	78%
2	84,05%	89,60%
3	91,45%	89,20%
4	93,10%	90,80%
5	95,20%	90,40%

Nilai rata-rata akurasi dari ke-5 fold menunjukkan seberapa efektif model dalam melakukan klasifikasi secara umum. Dalam mengklasifikasikan empat jenis daun, model ini menghasilkan akurasi dan validasi yang tinggi, dengan rata-rata akurasi pelatihan sebesar **95,20%** dan rata-rata validasi sebesar **90,40%**.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan pada mahasiswa kelas TI 2022 C Fakultas Ilmu Komputer Universitas Kuningan dengan menggunakan sistem **Presensi Kehadiran Mahasiswa Berbasis Face Recogniti dan Algoritma Convolutional Neural Network (CNN)**, dapat disimpulkan hal-hal berikut:

1. Aplikasi presensi kehadiran mahasiswa berbasis *face recognition* dengan algoritma CNN mampu mencegah terjadinya kecurangan dalam pencatatan kehadiran serta memudahkan dosen dalam mengevaluasi kehadiran mahasiswa.
2. Hasil pengujian menggunakan metode *K-Fold* menunjukkan bahwa aplikasi ini dapat melakukan pengenalan wajah dengan akurasi model sebesar **93,10%**, menandakan tingkat akurasi dan validasi yang tinggi.
3. Berdasarkan hasil penentuan *threshold*, disimpulkan bahwa nilai **0,3** merupakan ambang batas optimal untuk pencocokan wajah, di mana hanya wajah yang sama akan terdeteksi kecocokannya.

5. SARAN

Beberapa saran yang dapat dilakukan berdasarkan hasil penelitian ini untuk meningkatkan dan mengembangkan Aplikasi Presensi Mahasiswa

Berbasis *Face Recognition* dengan Algoritma *Convolutional Neural Network (CNN)* adalah sebagai berikut:

1. Disarankan untuk menambah jumlah dan variasi dataset wajah mahasiswa, termasuk kondisi pencahayaan, ekspresi wajah, dan sudut pengambilan gambar yang berbeda. Hal ini dapat meningkatkan akurasi dan kehandalan sistem dalam mengenali wajah secara lebih konsisten.
2. Perlu dilakukan eksperimen lebih lanjut untuk mengoptimalkan arsitektur model CNN, seperti penambahan lapisan (*layers*), pengaturan parameter, atau penggunaan teknik *transfer learning* dengan model pra-latih (*pre-trained model*) untuk meningkatkan performa sistem.
3. Aplikasi ini sebaiknya diuji coba pada skala yang lebih besar, seperti seluruh fakultas atau universitas, untuk mengevaluasi kinerja sistem dalam menangani jumlah pengguna yang lebih banyak dan beragam.
4. Pengembangan sistem presensi berbasis *real-time* dengan kemampuan deteksi wajah secara langsung (*real-time face detection*) dapat menjadi langkah lanjutan untuk meningkatkan efisiensi dan kenyamanan pengguna.

DAFTAR PUSTAKA

- [1] N. Lailatul Khoiriyah, F. Marisa, I. Dharma Wijaya, J. Teknik Informatika Jl Borobudur No, and J. Sukarno Hatta, "Rancang Bangun Sistem Presensi Online Berbasis Granted Validitas Data," 2018.
- [2] I. W. Suartika E. P, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, 2016.
- [3] J. W. Creswell and J. David Creswell, "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches," 2018.
- [4] Sugiyono, *Metode penelitian kuantitatif / Prof. Dr. Sugiyono*, 1st ed. Bandung : Alfabeta, 2018, 2018.
- [5] C. L. Nazalia, "Implementasi Algoritma Convolutional Neural Network untuk Deteksi Hama di Tanaman Sawi Hijau," *Institut Teknologi PLN Jakarta*, 2022.
- [6] R. S. Pressman and D. Bruce R. Maxim, *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 2014. [Online]. Available: <https://books.google.co.id/books?id=i8NmnAEA CAAJ>