

Aplikasi Presensi Pegawai Dengan Metode Principal Component Analysis (PCA) (Studi Kasus Badan Pusat Statistik Kabupaten Bandung)

Ade Yuliana M.T¹, Evi Samsudin²

Program Studi Teknik Informatika- Politeknik TEDC Bandung

Jl. Politeknik-Pesantren KM2 Cibabat Cimahi Utara – Cimahi Jawa Barat - Indonesia

yulianaad@poltektedc.ac.id, evisamsudin44@gmail.com

Abstrak

Presensi pegawai merupakan aspek penting dalam menjaga kinerja dan efisiensi organisasi, terutama pada lembaga seperti Badan Pusat Statistik (BPS) yang berperan dalam mengumpulkan dan menganalisis data statistik. Dalam upaya untuk mengidentifikasi dan menganalisis pola presensi pegawai di BPS Kabupaten Bandung masalah umum yang sering timbul dalam pengelolaan presensi pegawai dan mengurangi penghindaran presensi, kecurangan presensi, keterlambatan, dan kesalahan administratif. Penghindaran presensi dapat merugikan organisasi dalam hal produktivitas dan kedisiplinan. Kecurangan presensi dapat merusak integritas data presensi dan merugikan pegawai lainnya. Keterlambatan dan kesalahan administratif dapat mengganggu efisiensi operasional. Oleh karena itu, diperlukan aplikasi presensi pegawai dengan metode *Principal Component Analysis* (PCA) dengan Algoritma *Eigenface* yang mengidentifikasi pegawai secara akurat dan sulit dipalsukan. Aplikasi presensi pegawai merupakan aplikasi yang digunakan untuk mempermudah proses saat melakukan presensi. Aplikasi ini berbasis *website* dengan menggunakan *framework Laravel*, *MySQL* sebagai basis data. Berdasarkan hasil dari tahap pengembangan dengan tahapan dalam analisis kebutuhan menggunakan diagram UML, desain hingga tahapan testing dengan metode *black box testing* dan *User Acceptance Test* (UAT) untuk mengetahui tingkat kegunaan aplikasi kepada pengguna. Dalam pelaksanaan hasil pengujian/testing menggunakan metode *User Acceptance Test* dengan Interpretasi skor hasil didapat persentase 87,8%.

Kata kunci: Presensi Pegawai, *Framework Laravel*, *Principal Component Analysis* (PCA), Algoritma *Eigenface*, BPS Kabupaten Bandung.

Abstract

Employee presence is an important aspect in maintaining organizational performance and efficiency, especially in institutions such as the Central Statistics Agency (BPS) whose role is in collecting and analyzing statistical data. In an effort to identify and analyze employee attendance patterns at BPS Bandung Regency, common problems that often arise in managing employee attendance and reducing attendance avoidance, attendance fraud, delays, and administrative errors. Attendance avoidance can hurt an organization in terms of productivity and discipline. Attendance fraud can damage the integrity of attendance data and harm other employees. Administrative delays and errors can interfere with operational efficiency. Therefore, it is necessary to apply employee presence using the Principal Component Analysis (PCA) method which identifies employees accurately and is difficult to falsify. The employee attendance application is software that is used to simplify the process when making attendance. This website-based

application uses the Laravel framework, MySQL as the database. Based on the results of the development stage with requirements analysis stages using UML diagrams, design to testing stages using the black box testing method and User Acceptance Test (UAT) to determine the level of application usability to users. In implementing the test results using the User Acceptance Test method with interpretation of the results, the percentage obtained was 87.8%.

Keywords: *Employee Presence, Laravel Framework, Principal Component Analysis (PCA), Algoritma Eigenface, BPS Bandung Regency.*

I. PENDAHULUAN

Presensi merupakan elemen fundamental dalam menunjang efektivitas operasional instansi. Sistem presensi yang berkualitas tinggi diharapkan menghasilkan data yang akurat dan maksimal, serta sejalan dengan target yang telah ditetapkan. Catatan kehadiran pegawai merupakan dokumen waktu kedatangan dan kepulangan pegawai di perusahaan atau instansi, baik dalam bentuk daftar hadir manual maupun kartu yang diisi dengan mesin pencatat waktu [1].

Perkembangan sistem waktu dan presensi telah berkembang dengan hadirnya teknologi seperti komputer dan peralatan lainnya. Sistem presensi menggunakan kertas, komputer, sidik jari, pemindai mata dan sekarang banyak perangkat seperti Android. Adanya teknologi ini memungkinkan beberapa perusahaan atau institusi untuk memperbaharui sistemnya dengan menggunakan teknologi ini karena lebih mudah dioperasikan dan presensi dapat dilakukan dimana saja sehingga tidak ada waktu yang terbuang percuma.

BPS merupakan singkatan dari Badan Pusat Statistik adalah lembaga pemerintah non departemen yang bertanggung jawab langsung kepada Presiden [2]. Awalnya BPS menggunakan *Fingerprint Electronic Presence*, namun pada tahun 2020 BPS mengubah *Fingerprint Presence* menggunakan aplikasi Kaizala dan meluncurkan *Work From Home* (WFH). BPS menggunakan Kaizala sebagai media presensi, yaitu dengan menggunakan *request* presensi di *workgroup*. Presensi ini dilakukan secara otomatis oleh *admin* BPS RI dan dikirimkan pada jam kerja normal dan dapat juga dilakukan oleh *admin* grup.

Berdasarkan observasi meskipun sistem presensi tersebut sudah diterapkan dari tahun 2020 sampai sekarang, namun pada kenyataannya terdapat beberapa permasalahan,

setidaknya, nama perusahaan dan nama negara tempat penulis diantaranya pada saat mengambil data kehadiran *admin* harus mengirimkan permintaan kehadiran ke pesan grup karena akan tertumpuk pesan lainnya. Daftar kehadiran harus direkap setiap selesai melakukan presensi. Kekurangan lain *form* presensi dapat dibagikan oleh siapa saja tidak ada hak akses. Setiap yang memiliki *form* presensi dapat melakukan presensi. Kemudian saat melakukan presensi, pegawai dapat mengisi presensi kapan saja dan dimana saja pada lokasi yang tidak menentu, baik dalam jangkauan tempat kerja maupun di luar lokasi tempat kerja.

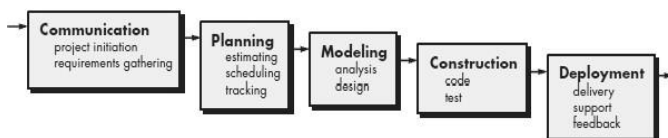
Metode *Principal Component Analysis* (PCA) sering digunakan dalam analisis data dan pengolahan gambar karena kemampuannya untuk mengurangi dimensi data dengan mempertahankan informasi yang paling penting. PCA mengidentifikasi komponen utama atau kombinasi linear dari fitur-fitur awal yang memiliki variansi tertinggi, sehingga mengurangi kompleksitas data tanpa mengurangi makna. Pada dasarnya metode PCA mengambil karakteristik wajah dengan mengekstrak informasi penting pada citra wajah. Fitur wajah digunakan untuk mendeteksi wajah dengan menghitung matriks vektor rata-rata dan data matrik dari *database* citra wajah, karakteristik wajah akan menjadi dasar untuk menghitung jarak dari wajah yang mewakili nilai bobot individu yang mewakili satu atau lebih dari citra wajah [3].

Pada umumnya, ada dua tahap urutan pencocokan wajah. Langkah pertama, yaitu pengenalan wajah, dimana program citra wajah pada gambar yang diambil oleh kamera untuk memastikan wajah seseorang tertangkap. Langkah kedua, untuk pendeteksian wajah (*face detection*).

Setelahnya, hasil data dari *face detection* yang terdeteksi dibandingkan dengan data yang tersimpan di *database*. Berdasarkan permasalahan diatas, maka peneliti membuat suatu aplikasi yang berjudul “Aplikasi Presensi Pegawai dengan Metode *Principal Component Analysis* (PCA) Di Badan Pusat Statistik Kabupaten Bandung”, diharapkan adanya aplikasi presensi dapat mempermudah untuk melakukan presensi yang akurat, efisien dan pengolahan data presensi.

II. METODE PENELITIAN

Metodologi penelitian pada kajian tugas akhir ini adalah *Metode Waterfall*. *Metode Waterfall* menawarkan pendekatan yang terstruktur dan terencana untuk pengembangan perangkat lunak. Model ini cocok untuk proyek dengan persyaratan yang jelas dan stabil, serta tim yang berpengalaman [4], tahapan dari metodologi ini dapat dilihat pada Gbr. 1 dibawah ini.



Gbr. 1 *Metode Waterfall*

1. *Communication*

Communication atau komunikasi yaitu untuk mengetahui kondisi sistem yang telah berjalan dan kebutuhan sistem yang diperlukan kemudian. Dalam pengumpulan data pada tahap ini melalui wawancara secara langsung dan melakukan studi literatur. Pada proses analisis ini penulis mengumpulkan data melalui wawancara secara langsung kepada kepala BPS Kabupaten Bandung, TU Kasubag dilakukan pada tanggal 12 desember 2022 sampai dengan 12 januari 2023 di instansi, mempelajari aplikasi yang digunakan saat ini dan mempelajari beragam literatur yang sesuai dengan penelitian selain itu, dilengkapi dengan mempelajari dokumen yang berlatu.

2. *Planning*

Tahapan *Planning* merupakan kelanjutan dari Tahap Komunikasi (Analisis Kebutuhan). Pada tahap ini, dokumen perencanaan untuk proses pengembangan perangkat lunak disusun. Dokumen ini memuat data-data terkait proses pembuatan sistem, termasuk langkah-langkah pengerjaan. Dengan cara pengumpulan data sebagai berikut:

Data Primer Pada data primer peneliti mengumpulkan data melalui wawancara secara langsung kepada kepala BPS Kabupaten Bandung, TU Kasubag dilakukan pada tanggal 12 desember 2022 sampai dengan 12 januari 2023 di instansi, mempelajari aplikasi yang digunakan.

Data Sekunder Pada data sekunder peneliti mengumpulkan data dengan mempelajari beragam literatur yang sesuai dengan penelitian dan mempelajari data-data yang sudah ada.

3. *Modeling (analysis dan design)*

Tahapan *modeling* atau tahap pemodelan, untuk sistem dapat dibuat dengan baik maka diperlukan perancangan sistem yang spesifik dan terperinci pada arsitektur sistem, penulis menerapkan Metode PCA (*Principal Component Analysis*) menggunakan Algoritma *Eigenface* kemudian dilanjutkan merancang *design* sistem dengan menggunakan *use case* diagram, *use case description*, activity diagram, dan perancangan antar muka.

4. *Construction (code dan test)*

Dengan memperoleh rancangan desain sistem, maka selanjutnya melakukan eksekusi desain sistem dengan bahasa pemrograman *framework Laravel*, dan *java script*. Kemudian dilanjutkan *test* atau pengujian aplikasi, pada tahapan pengujian aplikasi yang sudah dibangun akan melewati proses evaluasi aplikasi yang telah dikembangkan akan melalui serangkaian pengujian untuk memastikan kesesuaiannya dengan kebutuhan dan bebas dari kesalahan. Jika sistem tidak memenuhi ekspektasi pengguna, maka akan dilakukan perbaikan dan pengujian ulang. Pengujian yang akan dilakukan meliputi pengujian internal (*black box testing*) dan pengujian eksternal (pengujian oleh pengguna).

5. *Deployment*

Tahap *Deployment* adalah tahapan verifikasi atau final dimana klien (instansi) atau pengguna menguji aplikasi

tersebut telah sesuai dengan yang disetujui. Lalu, pembuatan perangkat lunak harus ada pemeliharaan secara berkala.

1) Analisis Sistem yang Sedang Berjalan

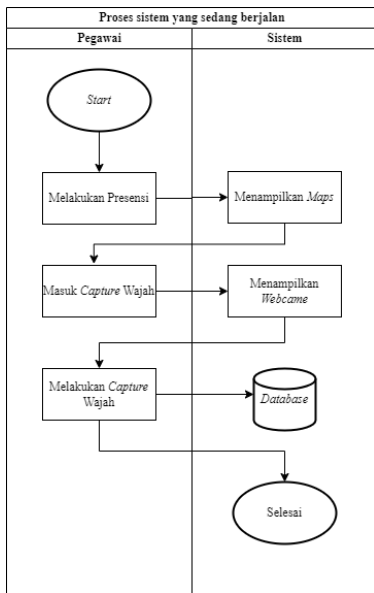
Analisis terhadap sistem yang dijalankan oleh BPS Kabupaten Bandung berdasarkan observasi penulis mulai melakukan pada hari Senin tanggal 12 Desember 2022 bersama Kepala BPS Kabupaten Bandung dan Kepala Bagian Kantor Pusat menemukan bahwa sistem yang berjalan saat ini terdapat kekurangan dalam hal kehadiran, berikut adalah masalah kehadiran kantor BPS Kabupaten Bandung:

1. Admin merekap data presensi kehadiran setiap melakukan presensi
2. Form presensi dapat dibagikan, jadi siapa saja yang memiliki form presensi dapat melakukan presensi.

Berikut ini analisa alur proses presensi pegawai yang masih berjalan di BPS Kabupaten Bandung:

1. Pegawai Membuka Aplikasi Kaizala.
2. Pegawai Masuk Ke grup chat dan melakukan Presensi
3. System menampilkan Maps sesuai lokasi
4. System menampilkan Face Reconition

Alur sistem yang berjalan dapat dilihat pada flowmap dalam Gbr. 2 Sistem presensi yang sedang berjalan.



Gbr. 2 Flowmap Sistem yang Berjalan

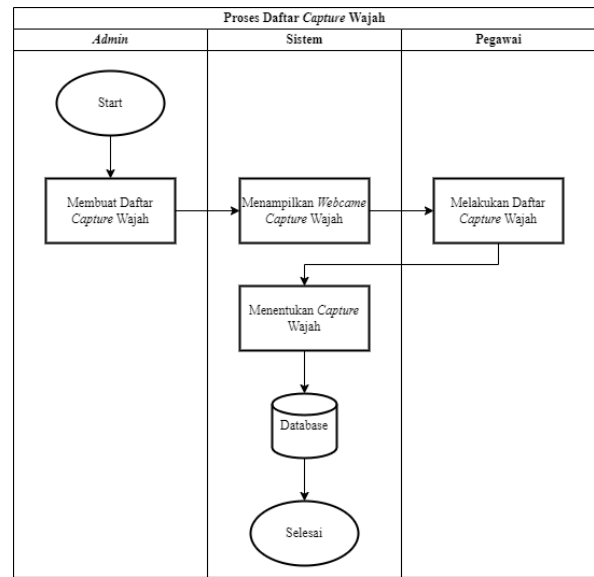
2) Analisis Sistem yang Akan Dibangun

Sistem presensi pegawai yang akan dikembangkan diharapkan dapat mengatasi berbagai permasalahan yang terjadi dalam pengambilan data pegawai. Pada sistem yang akan dikembangkan terdapat beberapa fitur yang akan mempermudah proses pengelolaan data presensi pegawai, diantaranya:

1. Admin/Operator membuat daftar capture wajah.

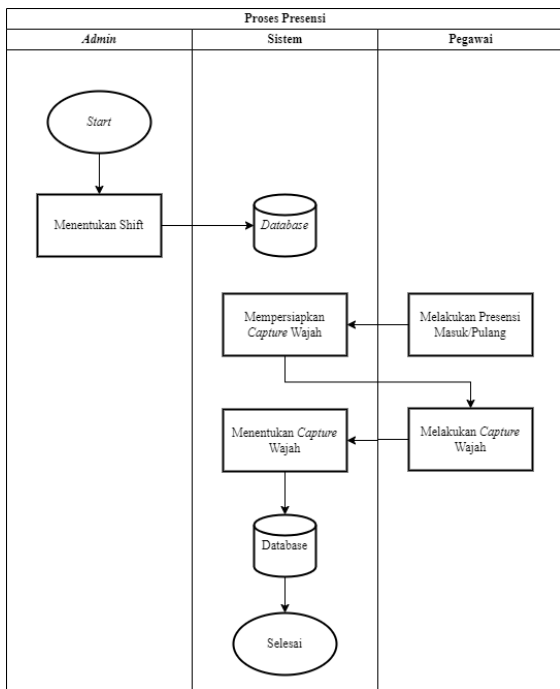
2. Pegawai melakukan capture daftar wajah yang telah di buatkan admin sampai terdeteksi oleh sistem dan tersimpan kedalam database.
3. Admin/operator membuat shift kerja pegawai.
4. Pegawai melakukan presensi yang telah di tentukan admin dengan ketentuan waktu dan lokasi presensi.
5. Sistem akan menentukan verifikasi wajah jika terdeteksi benar maka akan selesai presensi dan jika tidak tidak dapat melakukan presensi.

Alur proses dibagi menjadi 2 (dua) bagian, yaitu (1) tahapan capture wajah dan disimpan kedalam database dideskripsikan dalam Gbr.3 Flowmap proses Daftar Capture Wajah.



Gbr. 3 Flowmap proses Daftar Capture Wajah

Alur proses (2) yaitu proses presensi pegawai dan penyimpanan data dapat dilihat pada Gbr. 4 Flowmap proses Presensi.



Gbr. 4 Flowmap Alur Presensi

Dalam menyelesaikan fitur awal yaitu membuat daftar capture wajah, diterapkan Penerapan Metode PCA (*Principal Component Analysis*) menggunakan Algoritma *Eigenface*.

1) *Eigenface*

Eigenface salah satu metode pengenalan wajah berbasis *Principal Component Analysis* (PCA), menawarkan kemudahan implementasi. Pada metode ini, langkah awal adalah melakukan pemrosesan awal citra untuk menghasilkan gambar yang lebih optimal. [5]. Algoritma *face detection* pada *Eigenface* ada beberapa tahapan, yaitu [6]:

1. Menyusun *Flatvector* Matriks Citra.
2. Menghitung Nilai Tengah atau *Mean* (ψ).
3. Menghitung Selisih antara *Training Image* dengan Nilai Tengah atau *Mean* (ψ).
4. Menghitung Nilai *Matriks Kovarian*.
5. Menghitung Nilai *Eigenvalue* dan *Eigenvector*.
6. Mencari Nilai *Eigenface*.
7. Proses Identifikasi.

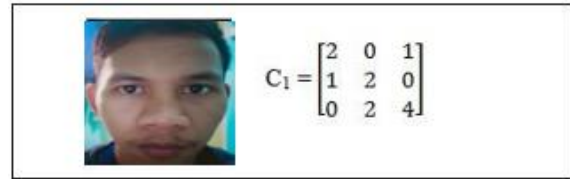
2) *Principal Component analysis*

Principal Component Analysis (PCA) merupakan suatu teknik untuk menemukan pola-pola tersembunyi dalam data dan kemudian menyajikannya dalam bentuk yang lebih sederhana untuk mempermudah visualisasi perbedaan dan kesamaan antar pola tersebut. [7] dalam [5].

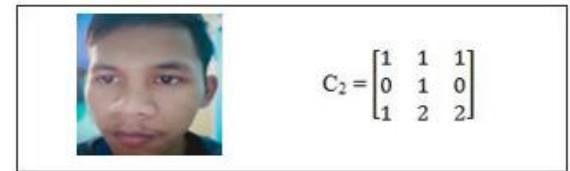
Adapun Langkah-langkah dalam penerapannya adalah sebagai berikut:

1. Tahap awal dalam menghitung nilai eigen sebuah foto adalah proses normalisasi. Dalam proses ini, format gambar diubah dari RGB menjadi skala abu-abu

(*grayscale*) terlebih dahulu. Selanjutnya, gambar *grayscale* tersebut diubah menjadi bentuk matriks. Adapun gambar foto tersebut dapat dilihat pada Gbr. 5 dan 6 adalah sebagai berikut:



Gbr. 5 Citra wajah 1



Gbr. 6 Citra Wajah 2

Disusunnya suatu himpunan *S* *matriks* terdiri dari *training image* keseluruhan.

$$S = \{\Gamma_1, \Gamma_2, \dots, \Gamma_M\} \dots\dots\dots (1)$$

Keterangan:

S = Himp. (set) yang berisi elemen-elemen $\Gamma_1, \Gamma_2, \dots, \Gamma_M$.
 $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ = Elemen yang membentuk himpunan S.

2. Setelah mendapatkan himpunan matriks, langkah selanjutnya adalah menghitung nilai rata-ratanya. Caranya, jumlahkan nilai matriks wajah 1 dan wajah 2, lalu bagi hasilnya dengan jumlah data wajah yang ada di database. Dalam penelitian ini, terdapat dua data wajah di database.

$$\Psi = \frac{1}{2} \sum_{n=1}^m \Gamma_n \dots\dots\dots (ii)$$

Keterangan:

Ψ = Nilai dari Ψ yang ingin dihitung.
M = Jumlah total *elemen* himpunan (jumlah dari $\Gamma_1, \Gamma_2, \dots, \Gamma_M$).
 Γ_n = Nilai *elemen* ke-n dalam himpunan.

$$\Psi = \frac{1}{2} \left(\begin{pmatrix} 2 & 0 & 1 \\ 2 & 2 & 0 \\ 0 & 2 & 4 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{pmatrix} \right)$$

Maka $\Psi = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{pmatrix}$

3. Menghitung Selisih data *Training* gambar (*Image*)

$$\Phi_i = \Gamma_i - \Psi \dots\dots\dots (iii)$$

Keterangan:

Φ_i = Nilai dari Φ_i yang ingin dihitung.
 Γ_i = Nilai *elemen* ke-i dalam himpunan.
 Ψ = Nilai rata-rata (*average*) dari semua elemen dalam himpunan ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$).

$$\begin{aligned} \Phi_i = \Gamma_i - \Psi &= \begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} 0 &= \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} - \begin{vmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{vmatrix} \\ 0 &= \det \begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \end{aligned}$$

Maka *eigenvalue* yang dihasilkan adalah $\lambda_1 = 3, \lambda_2 = 1, \lambda_3 = 3$

4. Menghitung Nilai *Matriks Kovarian*

Nilai *matriks kovarian* (C) digunakan untuk menghitung *eigenvalue* (λ) dan *Eigenvector* (v).

$$\begin{aligned} C &= \frac{1}{M} * \sum (\Phi_n - \Phi_T)^2 \\ L &= A^T A \quad L = \Phi_m^T \Phi_n \end{aligned} \quad (iv)$$

Keterangan:

- C = Nilai dari C yang ingin dihitung.
- M = Jumlah total *elemen* himpunan (jumlah $\Gamma_1, \Gamma_2, \dots, \Gamma_M$).
- Φ_n = Nilai dari *n*-nth *elemen* himpunan, dimana $n = 1, 2, \dots, M$.
- Φ_T = Nilai dari total *elemen* dalam himpunan ($\Phi_1 + \Phi_2 + \dots + \Phi_M$).
- L: Nilai dari *matriks* L yang ingin dihitung.
- A = *Matriks* data dengan dimensi $M \times N$, di mana M adalah jumlah *elemen* dalam himpunan dan N adalah dimensi atau *fitur* dari setiap *elemen*.
- Φ_m = Nilai dari *elemen* ke-*m* himpunan, dimana $m = 1, 2, \dots, M$.
- Φ_n = Nilai dari *elemen* ke-*n* himpunan, di mana $n = 1, 2, \dots, M$.

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Maka $L = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

5. Tahap selanjutnya adalah mencari nilai *eigenvalue* (λ) dan *eigenvector* (v) dari *matriks* kovarian (C).

$$C \times v_i = \lambda_i \times v_i \dots \dots \dots (v)$$

Keterangan:

- C= *Matriks* yang terlibat dalam analisis *eigenvalues* dan *eigenvectors*.
- v_i = *Vektor eigen* (*eigenvector*) yang merupakan *vektor* dari *matriks* C.
- λ_i = Nilai *eigen* (*eigenvalue*) yang berhubungan dengan *vektor eigen* v_i

Cari nilai nilai *eigenvalue* (λ) dan *eigen-vector* (v).

$$\begin{aligned} L \times v &= \lambda \times v \\ L \times v &= \lambda I \times v \\ L - \lambda I &= 0 \text{ atau } \lambda I - L = 0 \end{aligned}$$

Eigenvector (v) diperoleh dengan memasukkan nilai *eigenvalue* (λ) ke dalam persamaan $\lambda I - L v = 0$. *Eigenvektor* dari setiap *eigenvalue* dihitung berdasarkan kolom *eigenvalue* masing-masing dan kemudian digabungkan kembali menjadi satu *matriks*.

a. Untuk $\lambda_1 = 3$, maka:

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v1 adalah $= \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$

b. Untuk $\lambda_2 = 1$, maka:

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v2 adalah $= \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$

c. Untuk $\lambda_3 = 3$, maka:

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v3 adalah $= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Setelah didapat *eigenvector* v_1, v_2 , dan v_3 , maka *eigenvector* yang dihasilkan dari *matriks*.

$$L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

6. Menghitung nilai *Eigenface*

Langkah berikutnya, dengan *eigenvector* (v) yang diperoleh, maka nilai *eigenface* (μ) dapat dicari dengan: Cari nilai *eigenface* (μ):

$$\mu_i = \sum_{k=1}^m v_{ik} \phi_k \dots \dots \dots (vi)$$

Keterangan:

- μ_i = parameter yang ingin dihitung untuk *entitas* /objek ke-*i*.
- $\sum (\Sigma)$ = simbol penjumlahan yang menunjukkan bahwa kita akan menjumlahkan semua *elemen* yang memenuhi kondisi tertentu.
- $k=1$ = Nilai k akan bervariasi dari 1 hingga m, di mana m adalah suatu bilangan bulat yang menggambarkan jumlah *elemen* yang dijumlahkan.
- v_{ik} = Variabel atau nilai yang terkait dengan *entitas* ke-*i* dan memiliki *indeks* k.

ϕ_k = Bobot yang terkait dengan nilai vik dan memiliki indeks k.

$$1. \mu_1 = v \times \phi_1$$

$$= \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0 & -1 & 0 \\ -2 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$2. \mu_2 = v \times \phi_2$$

$$= \begin{bmatrix} 1 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

a. Proses Identifikasi Wajah

Dalam proses identifikasi, kita akan mencocokkan wajah baru dengan wajah-wajah yang tersimpan dalam basis data untuk mencari kecocokan. Sederhananya, kita mencari tahu apakah wajah baru tersebut identik dengan salah satu wajah yang telah terdaftar di *database*.

Pengenalan wajah baru yang masuk (*test face*), prosedur yang dilakukan sama pada data wajah yang ada di *database*, untuk mendapatkan nilai *eigenface* dari wajah baru.

$$\mu_{new} = v \times \Gamma_{new} - \Psi$$

$$\Omega = \mu_1, \mu_2, \dots, \mu_M \tag{vii}$$

Keterangan:

μ_{new} = Hasil dari perhitungan yang ingin dihasilkan.

v = nilai yang akan dikalikan dengan Γ_{new} .

Γ_{new} = Elemen lain yang mungkin memiliki arti khusus dalam konteks tertentu.

Ψ = Konstanta yang akan dikurangkan dari hasil perkalian v dan Γ_{new} .

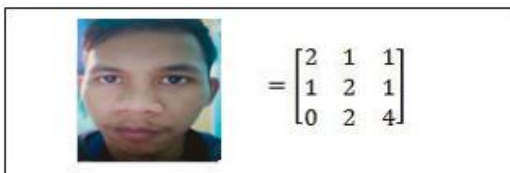
Ω = vektor Ω , yang merupakan suatu kumpulan nilai atau elemen.

$\mu_1, \mu_2, \dots, \mu_M$ = Nilai-nilai *individual* yang membentuk vektor Ω . μ_1 adalah nilai pertama, μ_2 adalah nilai kedua, dan seterusnya hingga μ_M yang merupakan nilai ke-M.

Pertama cari selisih (ϕ) antara *test face* dengan nilai tengah (Ψ). nilai *matriks test face* dari koordinat.

$$\phi_{new} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 2 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$



Gbr. 7 Citra Wajah baru (test face)

Kemudian, dengan adanya selisih (Φ) antara *tesface* untuk nilai tengah (Ψ) diketahui, maka nilai *eigenface* dapat dicari.

$$\mu_{new} = v \times \phi_{new}$$

$$\mu_{new} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mu_{new} = \begin{bmatrix} 0 & 0 & -1 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

Digunakannya *euclidean distance* untuk menghasilkan selisih terkecil antara *eigenface training image* (Γ_i) pada *database* dengan *eigenface test face*, maka jumlahkan *matriks* dari masing-masing *euclidean distance*.

$$\epsilon_k = \Omega - \Omega_k \tag{viii}$$

Keterangan:

ϵ_k = vektor ϵ_k , yang merupakan vektor hasil perbedaan atau selisih antara Ω dan Ω_k .

Ω = vektor Ω yg merupakan suatu kumpulan nilai atau elemen.

Ω_k = vektor Ω_k yang juga merupakan suatu kumpulan nilai atau elemen.

$$\epsilon_k = \begin{bmatrix} 0 & 0 & -1 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 & 0 \\ -2 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \sqrt{0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2}$$

$$= \sqrt{4}$$

$$= 2$$

$$\epsilon_k = \Omega - \Omega_{new}$$

$$= \begin{bmatrix} 0 & 0 & -1 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 & 0 \\ -2 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \sqrt{0^2 + 1^2 + 1^2 + 4^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2}$$

$$= \sqrt{8}$$

$$= 2.828$$

Berdasarkan hasil perhitungan, jarak *eigenface* citra wajah satu dengan *testface* menunjukkan nilai terkecil dibandingkan dengan wajah lainnya. Hal ini mengindikasikan bahwa *testface* memiliki kemiripan yang lebih tinggi dengan *face training* dibandingkan dengan wajah lain yang diuji. Oleh karena itu, dapat disimpulkan bahwa *testface* paling identik dengan *face training*.

A. Perancangan

Perancangan dibagi menjadi dua (2), yaitu sistem dan antarmuka.

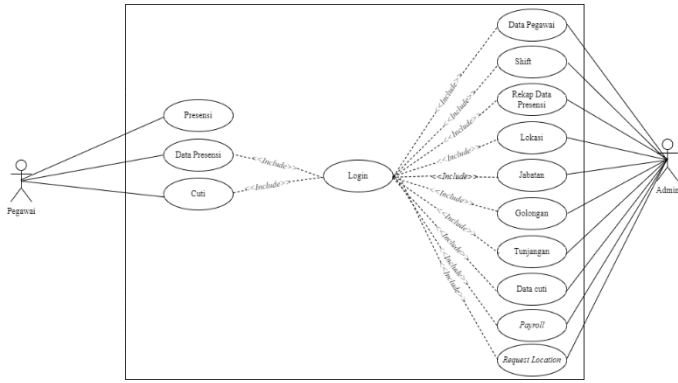
1) Perancangan Sistem

Perancangan sistem diperlukan untuk gambaran pada aplikasi presensi yang nantinya akan dibangun supaya terarah dan sesuai dengan kebutuhan. Aplikasi presensi ini dirancang

menggunakan *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan perancangan *database*.

a. *Use Case Diagram*

Use Case Diagram merupakan model yang menggambarkan bagaimana tingkah laku sistem informasi. *Use case diagram* menjelaskan tentang fitur yang ada pada aplikasi, yang bisa digunakan oleh aktor. Adapun aktivitas yang dapat dilakukan oleh masing-masing aktor bisa dilihat pada Gbr. 8 dibawah ini.

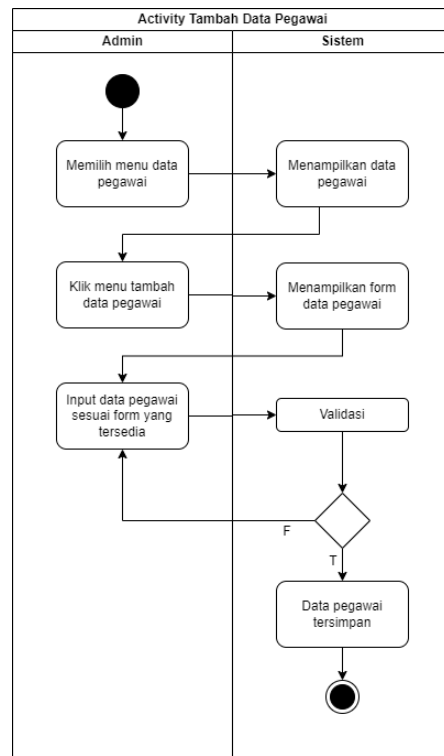


Gbr. 8 Diagram Use Case

b. *Activity Diagram*

Activity Diagram berfungsi untuk memvisualisasikan logika proses bisnis dan alur kerja dalam banyak kasus [8].

Salah satu *Activity diagram* dalam system ini adalah *activity* tambah data pegawai, yang menjelaskan aktivitas *admin* dalam menambah data pegawai, pertama *admin* masuk ke menu data pegawai, lalu sistem akan menampilkan data pegawai, memilih menu tambah data pegawai dan *system* akan menampilkan halaman *form* data tambah pegawai, kemudian *admin* mengisi data pegawai sesuai dengan *form* yang tersedia pada halaman tambah data pegawai. Untuk lebih jelasnya dapat dilihat pada Gbr. 9 berikut ini.

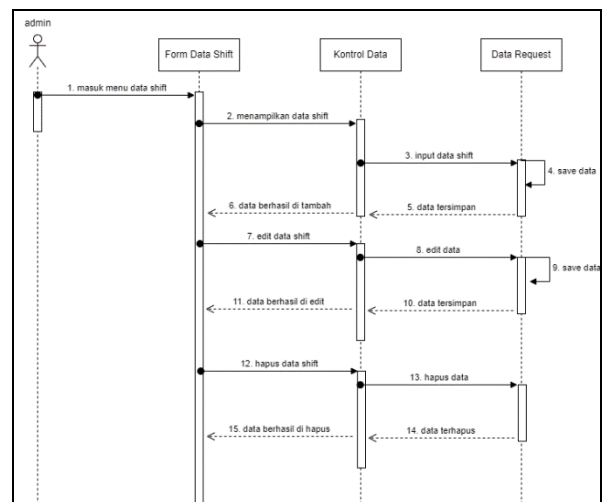


Gbr. 9 Activity diagram tambah data pegawai

c. *Sequence Diagram*

Sequence diagram memvisualisasikan perilaku objek dalam sebuah skenario penggunaan. Diagram ini menggambarkan garis hidup objek, pesan yang dikirimkan dan diterima antar objek, serta urutan waktu terjadinya interaksi tersebut. Berikut merupakan salah satu *sequence diagram* dari aplikasi presensi pegawai BPS Kabupaten Bandung.

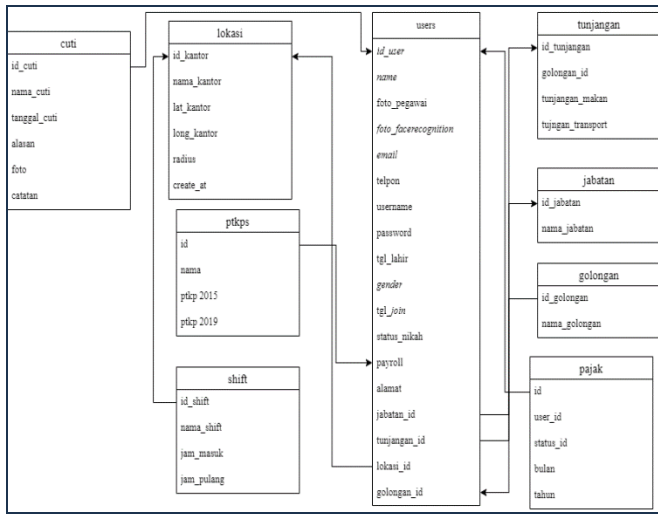
Gbr. 7 merupakan *Sequence* Kelola data shift pada aplikasi presensi pegawai di BPS Kabupaten Bandung adalah *actor* (*admin*) dapat menambah, mengedit dan menghapus shift kerja.



Gbr. 10 Sequence Diagram Kelola Data Shift

d. *Diagram Kelas*

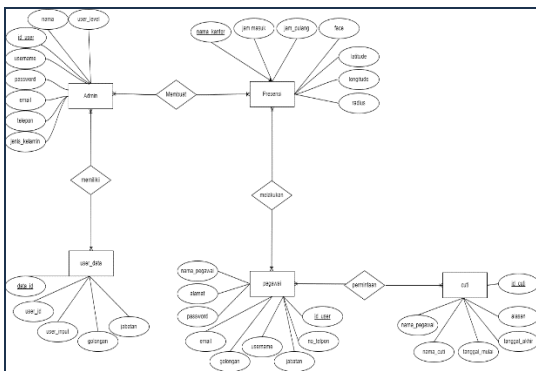
Class diagram yaitu proses yang memvisualisasikan struktur sistem dari kelas-kelas yang akan dibuat. Diagram kelas memiliki nama kelas, atribut, metode atau kinerjanya. Gbr. 11 adalah *class diagram* Aplikasi presensi pegawai.



Gbr. 11 *Class diagram*

e. *Entity Relationship Diagram (ERD)*

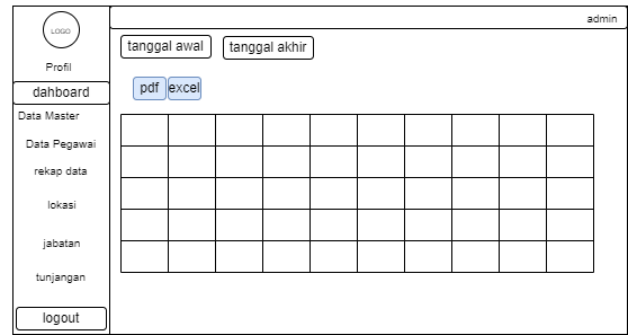
Entity relationship diagram sistem akses presensi pegawai BPS Kabupaten Bandung terdapat pada Gbr. 12.



Gbr. 12 *Entity Relationship Diagram*

2) *Perancangan Antarmuka*

Perancangan Antarmuka merupakan proses penting dalam menciptakan sistem yang mudah digunakan dan menarik bagi penggunanya. Proses ini melibatkan deskripsi terperinci tentang tampilan dan interaksi yang akan dialami pengguna saat berinteraksi dengan sistem yaitu Gbr. 13 yang merupakan halaman laporan presensi yang berfungsi untuk membuat laporan presensi.



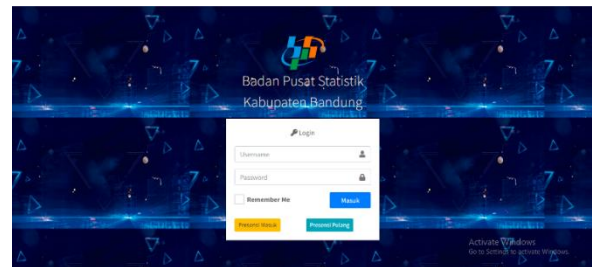
Gbr. 13 *Halaman Laporan Presensi*

B. *Implementasi*

Implementasi yaitu tahapan untuk mengimplementasikan semua perancangan menjadi suatu sistem melalui proses pemrograman. Tampilan beberapa implementasi pada sistem dimuat dalam Gbr. 14, Gbr.15 dan Gbr.16.

1) *Tampilan login*

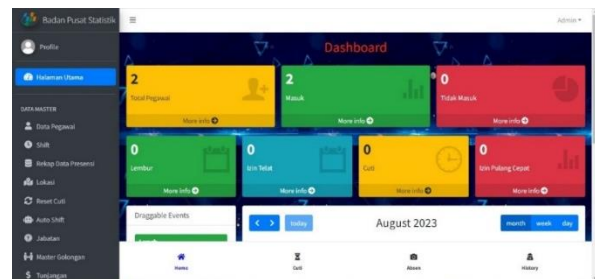
Pada Gbr. 14 dibawah ini menampilkan halaman *login* untuk semua *user* sebelum melakukan tahapan berikutnya, *user* diharuskan untuk *login* terlebih dahulu.



Gbr. 14 *Tampilan login*

2) *Tampilan dashboard*

Pada gambar dibawah ini menampilkan halaman *dashboard*, untuk lebih jelasnya dapat dilihat pada Gbr. 15 dibawah ini.



Gbr. 15 *Tampilan dashboard*

3) *Tampilan presensi pegawai*

Pada gambar dibawah ini menampilkan halaman presensi pegawai, untuk lebih jelasnya dapat dilihat pada Gbr. 16 dibawah ini.



Gbr. 16 Tampilan presensi.

C. Pengujian

Pengujian skenario perangkat lunak dilakukan oleh penulis ada dua (2) langkah pengujian yaitu pengujian *Black Box* dan *User Acceptance Test (UAT)*.

1) Pengujian *Black Box*

Pengujian *Black Box* merupakan pengujian perangkat lunak yang berfokus kepada fungsionalitas dari sistem yang sedang dibangun [10]. Pengujian *Black Box* dalam sistem ini dimuat pada TABEL I.

TABEL I
BLACK BOX APLIKASI PRESENSI

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Uji
1.	User melakukan Login	User dapat memasuki sistem sesuai dengan perannya masing-masing	Berhasil Hasil pengujian dapat dilihat pada Gbr. 14
2.	User Melakukan klik Dashboard	Menampilkan halaman utama/Dashboard	Berhasil Hasil pengujian dapat dilihat pada Gbr. 15
3	Pegawai melakukan presensi dengan menunjukkan wajah yang sudah didaftarkan ke <i>webcame</i>	Menampilkan hasil Wajah terdeteksi/ dikenali	Berhasil Hasil pengujian dapat dilihat pada Gbr. 16

2) Hasil Pengujian UAT

UAT merupakan suatu prosedur verifikasi untuk memvalidasi solusi yang diberikan oleh sistem sudah sesuai dengan kebutuhan penggunaannya. Secara umum, UAT diperuntukkan untuk *user* atau pengguna akhir yang memiliki fokus pada fungsionalitas dari sistem yang dibangun [11].

Penilaian menggunakan skala *likert* yang terdiri dari lima kategori, dapat dilihat pada TABEL II berikut.

TABEL II
SKALA LIKERT PENGUJIAN UAT

No	Nilai	Bobot Nilai
1	Sangat Baik (SB)	5
2	Baik (B)	4
3	Cukup (C)	3
4	Kurang (K)	2
5	Sangat Kurang (SK)	1

Rumus perhitungan skor sebagai berikut:

$$\text{Skor} = \sum_{i=1}^n \text{responen yang menjawab option } i \times i \dots (i)$$

$$\text{Rata-rata} = \frac{\text{skor}}{\text{jumlah responen}} \dots (ii)$$

$$\text{Skor tertinggi} = 5 \times \text{jumlah responen} \dots (iii)$$

$$\text{Skor terendah} = 1 \times \text{jumlah responen} \dots (vi)$$

Berdasarkan yang dibutuhkan pengguna dalam pengujian UAT dibuat dalam bentuk kuesioner yang terdiri dari tiga kelompok pertanyaan yaitu desain (3 pertanyaan), fitur (5 pertanyaan) dan Kepuasan pengguna (3 pertanyaan). Setelah dilakukan pengujian terhadap 15 responden. Hasil pengujian secara terinci terdapat pada TABEL III dan Tabel IV Nilai total rata-rata.

TABEL III
HASIL PENGUJIAN UAT

Huruf	Nilai	Desain			Fitur				
		D1	D2	D3	F1	F2	F3	F4	F5
SB	5	9	8	9	10	1	9	10	9
B	4	5	7	6	4	-	5	5	4
C	3	1	-	-	1	-	1	-	2
K	2	-	-	-	-	-	-	-	-
SK	1	-	-	-	-	-	-	-	-
Jumlah		15 Orang							
Skor		68	68	69	69	5	70	70	67
Rata-rata		4,5	4,5	4,6	4,6	5	4,6	4,6	4,4

TABEL IV
NILAI TOTAL RATA-RATA UAT

Huruf	Nilai	Kepuasan			Total Nilai
		K1	K2	K3	
SB	5	8	9	8	450
B	4	6	4	4	200
C	3	1	2	3	33
K	2	-	-	-	-
SK	1	-	-	-	-
Jumlah		15 Orang			
Skor		67	67	65	683
Rata-rata		4,4	4,6	4,3	91

Hasil pengujian *User Acceptance Test (UAT)* pada tabel di atas jumlah skor untuk pertanyaan yang diajukan kepada responden dengan perhitungan sebagai berikut:

$$\text{Skor tertinggi} : 5 \times 151 = 750$$

$$\text{Skor terendah} : 1 \times 151 = 151$$

$$\begin{aligned} &\text{Interprestasi skor hasil pengamatan} \\ &= (\text{jumlah nilai} / \text{skor tertinggi}) * 100\% \\ &= (683 / 750) * 100\% = 91\% \end{aligned}$$

Hasil dari pengujian UAT yang dilakukan terhadap 15 responden dengan 11 pertanyaan didapatkan hasil pengujian UAT 91% yang menunjukkan hasil yang sangat baik..

III. KESIMPULAN DAN SARAN

A. Kesimpulan

Diperoleh hasil yang berdasarkan dari proses penelitian, perancangan, dan implementasi hingga pengujian maka penulis dapat menarik kesimpulan sebagai berikut:

1. Aplikasi presensi pegawai berbasis *web* ini adalah media yang dapat digunakan untuk melakukan presensi harian dan dapat digunakan untuk menyampaikan informasi kepada pimpinan. Aplikasi ini dirancang penulis menggunakan sistem operasi *windows 10*, *xampp* sebagai servernya dan *framework laravel* sebagai bahasa *cripting* dan *Mysql* sebagai *database management system* (DBMS).
2. Perancangan metode *Principal Component Analysis* (PCA) telah menghasilkan pendekatan yang efektif untuk mengurangi dimensi data dengan mempertahankan informasi yang signifikan. Melalui analisis komponen utama, metode ini memungkinkan representasi data yang lebih ringkas dan memiliki potensi untuk meningkatkan kualitas analisis.
3. Implementasi aplikasi presensi telah menghasilkan solusi yang memudahkan pengelolaan dan pemantauan presensi dalam lingkungan kerja. Aplikasi ini dirancang dengan antarmuka pengguna yang intuitif, memungkinkan pengguna untuk melakukan pencatatan presensi dengan mudah. Aplikasi ini memberikan solusi yang efisien untuk masalah pemantauan presensi pegawai.
4. Hasil pengujian aplikasi presensi menunjukkan bahwa sistem berjalan sesuai dengan harapan dan memenuhi kebutuhan yang diinginkan. Pengujian fungsional menginformasi bahwa fitur-fitur seperti pencatatan presensi, penghitungan kehadiran, dan laporan statistik berfungsi dengan baik. Berdasarkan hasil pengujian yang telah dilakukan dengan pengujian *black box testing* dan *user acceptance testing* (UAT) dengan perhitungan interpretasi skor hasil dari pengujian UAT yang dilakukan memperlihatkan 91% telah terpenuhi.

B. Saran

Mengingat berbagai keterbatasan yang dimiliki dari penelitian yang telah penulis lakukan, penulis memberikan saran-saran untuk pengembangan dan meningkatkan aplikasi presensi pegawai sebagai media presensi:

1. Tingkat keamanan dan privasi, menjamin keamanan data dengan menerapkan *encryption* keamanan yang kuat.
2. Pada fitur *payroll* perhitungan gaji pegawai dan manajemen bayaran kepada pegawai berdasarkan data presensi serta transparansi dalam perhitungan gaji dan tunjangan.

REFERENSI

- [1] Kurnia, F. N. (2022). Rancang bangun sistem absensi di upt pengawasan mutu dan keamanan pangan. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 8(2), 169–179.
- [2] Badan Pusat Statistik. (2022). Sejarah Badan Pusat Statistik. 2022. <https://bandungkab.bps.go.id/menu/1/sejarah.html>
- [3] Sanusi, S., Alfaizun, A., & Mahyuddin, M. (2022). Sistem Pengenalan Citra Wajah Menggunakan Metode *Eigenface* dengan Visual Studio. *Jurnal Teknologi Informasi*, 1(1), 1–8. <https://doi.org/10.35308/v1i1.5432>
- [4] Tujni, B., & Hutrianto, H. (2020). Pengembangan Perangkat Lunak

- Monitoring Wellies Dengan Metode Waterfall Model. *Jurnal Ilmiah Matrik*, 22(1), 122–130. <https://doi.org/10.33557/jurnalmatrik.v22i1.862>.
- [5] Pratiwi, E.D & Harjoko, A Implementasi Pengenalan Wajah Menggunakan PCA (Principal Component Analysis). *IJEIS*, Vol.3, No.2, October 2013, pp. 175–184 ISSN: 2088-3714
- [6] Muliawan, R.M et. al Implementasi Pengenalan Wajah Dengan Metode Eigenface Pada Sistem Absensi. *Jurnal Coding, Sistem Komputer Untan* Volume 03, No. 1 (2015). Hal 41-50 ISSN : 2338-493X
- [8] Lim, Resmana., et. al 2002. Face Recognition Menggunakan Metode. Linear Discriminant Analysis (LDA). *Proceeding Komputer dan Sistem Intelijen*. Jakarta 21- 22 Agustus 2002
- [9] Haryanto, B. (2004). *Rekayasa Sistem Berorientasi Objek*. Bandung: Penerbit Informatika Bandung.
- [10] Rungta, K. (2019). *Software Testing. Learn software testing in 1 day: Definitive Guide to Learn Software Testing for Beginners*. Independently Published
- [11] Perry, W. E. (2006). *Effective Methods for Software Testing*. New York City: Wiley Publishing.