



Implementasi *Deep Learning* untuk Sistem Keamanan Data Pribadi Menggunakan Pengenalan Wajah dengan Metode *Eigenface* Berbasis Android

Lintang Bagas Adrianto ¹, Mohammad Iwan Wahyuddin ², Wina Winarsih ³

^{1,2,3} Program Studi Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional.

article info

Article history:

Received 11 October 2020

Received in revised form

30 November 2020

Accepted 1 December 2020

Available *online* January 2021

DOI:

<https://doi.org/10.35870/jtik.v5i1.201>

Keywords:

Face recognition, Deep Learning, Android, Eigenface.

Kata Kunci:

Pengenalan Wajah; Deep Learning; Android; Eigenface.

abstract

The development of technology in security systems combined with facial recognition, of course, makes every protected data safe. Many methods can be combined with a security system, one of which is the eigenface method, which is part of facial recognition. In this study, a personal data security system was built using Android-based deep learning. Based on the results of tests carried out on three devices with different Android versions, it is known, if on Android 8.1 (Oreo) the maximum distance is ± 40 cm, on Android 9.0 (Pie) the maximum distance is ± 50 cm, and on the Android version, 10.0 (Q) the maximum distance for facial object recognition is ± 60 cm. From the test results, it is known that by using the eigenface method, the farther the face is from the camera, the face cannot be detected. The implementation of this system is expected to protect personal data safely.

abstrak

Berkembangnya teknologi pada sistem keamanan yang dipadukan dengan pengenalan wajah, tentunya membuat setiap data yang dilindungi menjadi aman. Banyak metode yang dapat digabungkan dengan sistem keamanan, salah satunya metode eigenface, yang merupakan bagian dari pengenalan wajah. Pada penelitian ini dibangun sebuah sistem keamanan data pribadi menggunakan deep learning berbasis android. Berdasarkan hasil pengujian yang dilakukan pada tiga perangkat dengan versi android yang berbeda, maka diketahui, jika pada android 8.1 (Oreo) jarak maksimalnya ± 40 cm, pada android 9.0 (Pie) jarak maksimalnya ± 50 cm, dan pada versi android 10.0 (Q) jarak maksimal untuk pengenalan objek wajah, yaitu ± 60 cm. Dari hasil pengujian tersebut diketahui dengan metode eigenface, semakin jauh jarak wajah dari kamera, maka wajah tersebut tidak dapat terdeteksi. Implementasi dari sistem ini diharapkan dapat melindungi data pribadi dengan aman.

*Corresponding author. Email: lintangbagas20@gmail.com ¹.

© E-ISSN: 2580-1643.

Copyright © 2021. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan Riset) (<http://creativecommons.org/licenses/by/4.0/>).

1. Latar Belakang

Pengenalan wajah adalah merupakan teknologi komputer yang dapat digunakan untuk menentukan ukuran pada wajah, pendeteksian wajah, lokasi wajah, dan pengabaian latar belakang (*background*) citra, yang dimana kemudian dilakukan pengidentifikasian pada citra suatu wajah. Seiring berkembangnya teknologi, pengenalan wajah akhirnya digunakan dalam perangkat sistem android yang diterapkan untuk sistem keamanan data. Pengenalan atau pendeteksian wajah pada perangkat sistem android (*smartphone*) dilatarbelakangi dengan dibutuhkannya sebuah sistem yang dapat mengidentifikasi dan memverifikasi dengan lebih akurat dan aman dibandingkan hanya dengan password. Kesulitan dalam melakukan pengenalan atau pendeteksian wajah muncul dikarenakan kekompleksan dari kondisi dan bentuk wajah, yaitu seperti kualitas data gambar yang ditangkap, kemudian dari segi warna, pencahayaan, posisi data gambar yang tertangkap, maupun dalam hal perubahan geometrinya [1]. Tujuan pada penelitian ini adalah agar dapat membuat sistem keamanan data pribadi menggunakan pengenalan wajah dengan metode *eigenface* untuk melindungi berbagai macam data pribadi.

Pada penelitian terdahulu dilakukan penelitian terkait implementasi deep learning untuk sistem keamanan data pribadi menggunakan pengenalan wajah dengan mengimplementasikan *eigenface* untuk pengenalan wajah untuk melakukan tes untuk mendeteksi wajah dan mengenali citra wajah dengan bantuan opencv pada smartphone android [1]. Penelitian terkait juga dilakukan dengan pembahasan tentang pengenalan wajah dengan metode *eigenface*, namun secara real-time, yang bertujuan untuk melakukan uji coba sistem apakah penggunaan dan penerapan *eigenface* dapat melakukan pengenalan dan pendeteksian wajah secara real-time hingga empat data wajah secara langsung dengan kamera [2]. Penelitian serupa juga dilaksanakan dengan dilakukannya perancangan dan pengimplementasian pengenalan wajah dengan *eigenface* dengan tingkat keberhasilan pengenalan wajah hingga 80% dikondisi yang berbeda-beda (jarak objek, pencahayaan wajah, posisi wajah, atribut pada wajah, dan mimik wajah) [3].

Metode *eigenface* juga digunakan untuk pengenalan wajah, dimana *eigenface* digunakan untuk melakukan proses identifikasi bagi wajah yang menggunakan aksesoris (kacamata dan kerudung) dengan tingkat keberhasilan hingga 85% [4].

Penelitian serupa juga dilakukan dengan diimplementasikannya pengenalan wajah dengan Principal Component Analysis (PCA) dan *eigenface* dan dari penelitian tersebut, maka dapat diketahui bahwa pada sistem yang telah dibuat tersebut berhasil mengenali wajah manusia dan bekerja dengan lebih baik dalam berbagai kondisi orientasi wajah [5].

Selanjutnya, penelitian terkait juga dilakukan dengan menggunakan metode *eigenface* dan Principal Component Analysis (PCA) yang kemudian dilakukan pengujian hingga 100 kali pengujian untuk mendeteksi wajah dan berhasil terdeteksi hingga 94 kali terdeteksi dan 2 kali salah mengenali dan 4 tidak terdeteksi, jadi dapat disimpulkan bahwa keberhasilan dari akurasi wajah pada penelitian ini cukup tinggi, mencapai 94% [6]. Penelitian serupa terkait hal tersebut juga dilakukan dengan menggunakan *eigenface*, dimana dapat diketahui, bahwa dari penelitian yang dilakukan dari pengujian dengan database gambar wajah yang berisi 190 data gambar dari 38 orang berbeda (5 data gambar per orang) dapat diketahui, bahwa untuk pengenalan wajahnya cukup hanya dengan mengambil 10% data *eigenfacenya* dengan nilai tertingginya, dan jika jarak minimum antara citra uji dan citra lain adalah nol, citra uji sepenuhnya cocok dengan citra dari basis pelatihan, kemudian jika jaraknya lebih besar dari nol tetapi kurang dari ambang tertentu, itu adalah orang yang dikenal dengan ekspresi wajah lain, jika tidak maka itu adalah orang yang tidak dikenal. [7].

Kemudian penelitian juga dilakukan dengan menggunakan metode *eigenface* pada sistem password laptop, dimana dari penelitian ini dapat diketahui bahwa proses pengenalan dan pendeteksian wajah untuk membuka password laptop sangatlah sensitif, dikarenakan harus menyesuaikan dengan intensitas cahaya, jarak dan posisi wajah yang berbeda beda, dan dari sistem ini dapat diketahui bahwa sistem ini dapat lebih cepat untuk membuka password dibandingkan menggunakan keyboard [8].

Penelitian terkait menggunakan *eigenface* juga dilakukan dengan diterapkannya *eigenface* pada database yang sangat besar yang terdiri dari 3040 gambar dengan detail kacamata, jenggot, kumis, dan lainnya, dimana persentase hasil simulasi tingkat keberhasilannya hingga 94,74% [9].

Penelitian serupa terkait metode *eigenface* juga dilakukan bersamaan dengan PCA dan dapat ditarik kesimpulan dari penelitian ini bahwa pengujian yang dilakukan dengan berbagai subjek di lingkungan yang berbeda menunjukkan, *eigenface* masih mempunyai keterbatasan terhadap intensitas cahaya, ukuran, dan orientasi kepala, akan tetapi metode ini memperlihatkan klasifikasi wajah yang baik (tingkat keberhasilan lebih dari 85%) [10].

Dari beberapa penelitian yang sebelumnya telah dikemukakan, maka penulis melakukan penelitian dan memaparkan tahap-tahap yang akan dilakukan pada pembuatan aplikasi tersebut. Pengenalan wajah dipilih karena dapat mengenali objek wajah dengan baik hanya dengan menggunakan kamera perangkat smartphone sebagai penampil objek agar objek wajah dapat muncul. Pembuatan aplikasi ini dibuat menggunakan perangkat lunak android studio dan dengan pengimplementasian deep learning dengan metode *eigenface*.

2. Metode Penelitian

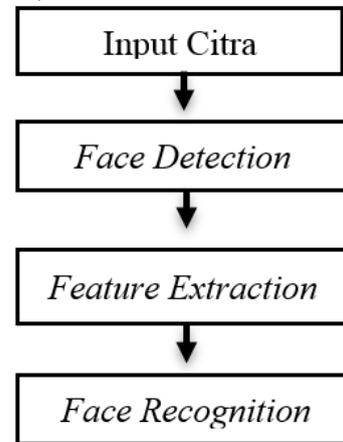
Deep Learning (Pembelajaran Dalam)

Adalah merupakan salah satu pembelajaran mesin (*machine learning*) yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data, dengan fungsi transformasi non-linear yang berlapis dan mendalam bahkan bisa mencapai hingga ratusan lapisan. Algoritma pada *deep learning* dapat digunakan untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-supervised learning*), yang dapat digunakan juga diberbagai aplikasi, seperti pengenalan wajah, citra maupun suara, mengklasifikasi teks, dan lain - lain.

Pengenalan Wajah

Pengenalan wajah (*Face Recognition*) adalah merupakan sebuah proses identifikasi atau verifikasi sebuah citra data wajah menggunakan algoritma komputasi dan dibandingkan dengan

citra data wajah yang ada. Pengenalan dan pendeteksian wajah juga merupakan salah satu teknologi biometrik yang telah digunakan diberbagai sistem keamanan seperti pengenalan retina mata, iris mata, dan pengenalan sidik jari. Pengenalan wajah juga bagian dari aplikasi visi komputer, yang dapat melakukan identifikasi dan verifikasi data hanya dari gambar digital atau *frame* video, hanya dengan cara membandingkan fitur wajah dari data gambar, dengan data – data wajah pada *database*. Berikut merupakan blok diagram dari pengenalan wajah.



Gambar 1. Blok Diagram Pengenalan Wajah.

Dari blok diagram pengenalan wajah pada gambar 1. dapat diketahui bahwa pada pengenalan wajah, awalnya adalah merupakan data masukan (*input*) citra, bisa berupa wajah dan lainnya. Selanjutnya, dilakukan pendeteksian wajah untuk mendeteksi adanya objek, yang kemudian masuk ke ekstraksi fitur yang berupa fitur untuk mengenali ciri – ciri baik pada citra maupun sekitarnya, dan terakhir barulah didapatkan hasil pengenalan wajah (wajah terdeteksi).

Eigenface

Eigenface adalah merupakan sekumpulan vektor *eigen* yang digunakan dalam masalah *computer vision* untuk pengenalan wajah manusia dan mengacu pada pendekatan berbasis penampilan untuk pengenalan wajah yang berusaha menangkap variasi dalam kumpulan gambar wajah dan menggunakan informasi tersebut untuk menyandingkan dan membandingkan gambar wajah individu secara holistik [5]. *Eigenface* itu sendiri cukup sederhana, dimana data latih direpresentasikan dengan vektor *flat* (gabungan vektor) dan digabungkan menjadi matriks tunggal, dan lalu *eigenface* dari masing - masing citra diekstraksi

dan disimpan pada *database*, dimana selanjutnya *test image* yang telah masuk didefenisikan juga *eigenface*nya dan dilakukan perbandingan dengan *eigenface* dari data wajah pada *database*.

Berikut merupakan tahapan perhitungan *eigenface*.

- a) Pertama adalah data disiapkan dengan membuat himpunan S yang terdiri dari semua data *training image* ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$).

$$S = \{ \Gamma_1, \Gamma_2, \dots, \Gamma_M \}$$

- b) Kemudian diambil nilai tengahnya atau *mean* (Ψ).

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

- c) Kemudian dicarilah selisih (Φ) antara *training image* (Γ_i) dengan nilai tengah (Ψ).

$$\Phi_i = \Gamma_i - \Psi$$

- d) Kemudian, barulah menghitung nilai matriks kovarian (C).

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T$$

$$L = A^T A \quad L = \Phi_m^T \Phi_n$$

- e) Setelah itu, dihitunglah *eigenvalue* (λ) dan *eigenvector* (v) dari matriks kovarian (C).

$$C \times v_i = \lambda_i \times v_i$$

- f) Selanjutnya, jika *eigenvector* (v) telah diperoleh, maka selanjutnya *eigenface* (μ) sudah dapat dicari dengan:

$$\mu_i = \sum_{k=1}^M v_{ik} \Phi_k$$

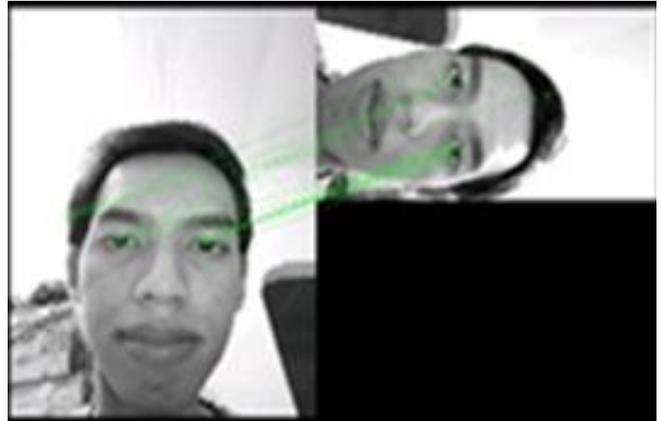
$l = 1, \dots, M$

3. Analisis dan Pembahasan

Penerapan Metode Eigenface

Pada proses awalnya, wajah baru diubah menjadi *eigenface*, kemudian dimasukkan ke dalam penyimpanan. Lalu, pada gambar masukan dilakukan perbandingan dengan gambar rata-rata dan perbedaan tersebut dikalikan dengan masing-masing vektor *eigen* dari matriks L . Setiap nilai tentunya memiliki bobot dan akan disimpan pada vektor Ω .

$$\omega_k = u_k^T (\Gamma - \Psi) \Omega^T = [\omega_1, \omega_2, \dots, \omega_m]$$



Gambar 2. Penambahan Data *Training Image*.

Selanjutnya ditentukan kelas wajah mana saja yang memberikan gambaran terbaik untuk citra masukannya, hal ini dilakukan dengan meminimalkan jarak *euclidean*.

$$\epsilon_k = \| \Omega - \Omega_k \|^2$$

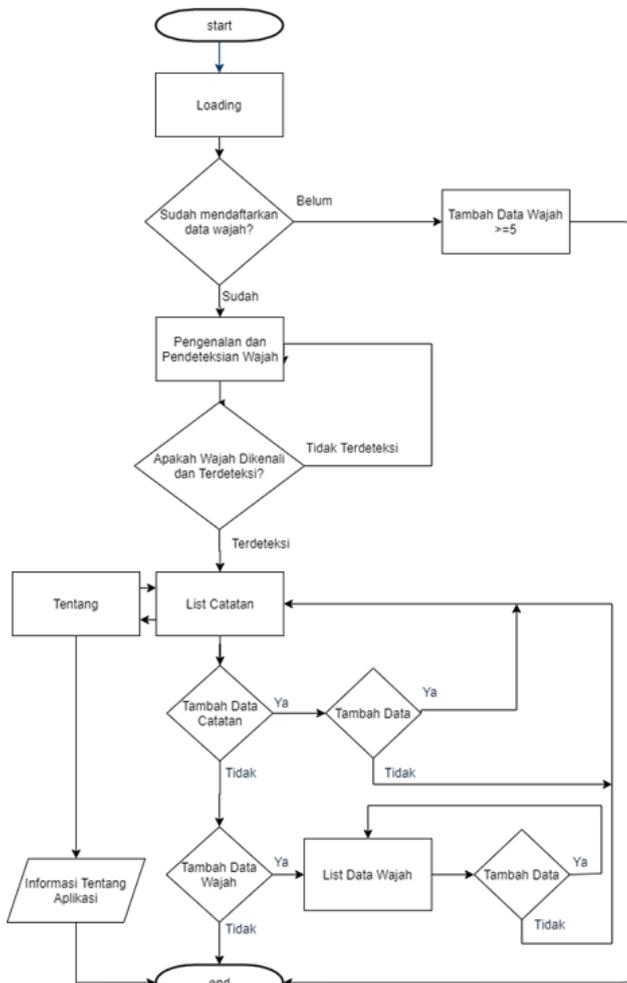
Kemudian, pada proses wajah input dilakukan pertimbangan untuk penggolongan kelas. Jika hasil ϵ_k di bawah ambang atau *threshold* $\theta\epsilon$ maka, citra merupakan wajah yang dapat dikenali, tetapi jika perbedaan berada di atas *threshold* yang diberikan, tetapi di bawah *threshold* yang kedua, maka citra merupakan wajah yang tak dikenali. Jika citra masukan bukan bagian dari kedua *threshold* di atas maka citra merupakan bukan wajah.



Gambar 3. Perbandingan Data Wajah.

Flowchart Aplikasi

Flowchart digunakan sebagai penggambaran penggunaan aplikasi yang dijabarkan melalui diagram alir. Berikut merupakan *flowchart* proses perangkat lunak secara keseluruhan pada aplikasi Notepriv.

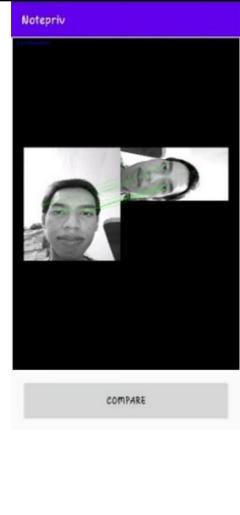


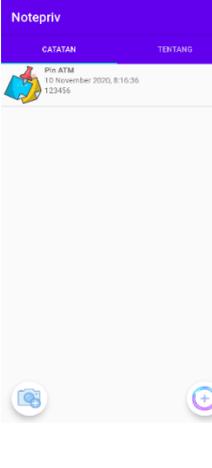
Gambar 4. Flowchart aplikasi Notepriv.

Dari flowchart aplikasi pada gambar 4. dapat diketahui saat aplikasi pertama kali dijalankan maka langsung ditampilkan halaman *loading* dan setelah itu barulah masuk ke halaman pendaftaran data wajah. Jika belum mempunyai data wajah, maka akan diarahkan ke halaman untuk pendaftaran wajah, namun jika sudah memasukkan data wajah, maka selanjutnya akan langsung diteruskan ke halaman pengenalan wajah. Pada halaman pengenalan wajah ini dilakukan pendeteksian citra wajah dengan menggunakan pengenalan atau pendeteksian yang merupakan bagian dari *deep learning* yang berguna untuk melakukan proses identifikasi atau verifikasi sebuah data citra wajah yang tidak diketahui dengan metode *eigenface*, lalu kemudian dibandingkan dengan data wajah yang sudah ada pada *database*. Jika dikenali, maka langsung masuk ke *list* catatan yang merupakan tempat untuk menyimpan data, dimana pada halaman tersebut dapat menambah ataupun menghapus data catatan. Selain itu pada halaman tersebut juga dapat melakukan penambahan ataupun mengubah data

wajah, serta terdapat halaman *about* untuk menjelaskan alasan untuk apa dan mengapa aplikasi dibuat.

Storyboard

No.	Bagian	Sisi
1		Tampilan awal ketika membuka aplikasi adalah halaman <i>loading</i> , yang merupakan halaman pertama yang akan ditampilkan sebelum pada akhirnya dilanjutkan ke halaman pendeteksian wajah, jika sudah ada data wajahnya. Namun jika belum akan diarahkan ke halaman tambah data wajah terlebih dahulu.
2		Kemudian, akan ditampilkan halaman untuk deteksi dan pengenalan wajah, dimana wajah yang dideteksi akan dibandingkan dengan data wajah yang sudah ada pada <i>database</i> aplikasi. Diproses ini, metode pengenalan wajah digunakan untuk mengenali objek citra wajah.

<p>3</p>		<p>Berikutnya, jika wajah tidak sama dengan data wajah pada <i>database</i>, maka aplikasi akan <i>close</i> sendiri dan kemudiah jika masuk lagi akan diarahkan ke halaman tambah data wajah. Pada halaman tersebut pengguna diminta mendaftarkan wajahnya sebanyak 5 atau lebih data wajah, yang nantinya data wajah pada <i>database</i> tersebut akan digunakan untuk data pembandingan pada saat masuk aplikasi.</p>
<p>4</p>		<p>Jika <i>button add</i> pada halaman <i>list</i> data wajah ditekan, maka akan menampilkan halaman tambah data wajah, dimana dapat mengisi wajah dengan foto menggunakan kamera <i>smartphone</i> atau mengambil foto dari galeri.</p>
<p>5</p>		<p>Jika sudah berhasil masuk ke aplikasi, maka akan ditampilkan halaman <i>list</i> data catatan, dimana dapat mengisi data pribadi dalam bentuk catatan (<i>note</i>) untuk data pribadi seperti pin ATM, <i>password</i>, dan <i>lainnya</i>. Terdapat juga dua <i>button</i>, yaitu untuk menambah catatan dan menambah atau mengganti data wajah.</p>
<p>6</p>		<p>Jika <i>button add</i> pada halaman <i>list</i> data catatan ditekan, maka akan ditampilkan halaman untuk menambah data catatan yang diinginkan, baik berupa pin ATM, <i>password</i>, nomor telepon, dan <i>lainnya</i>.</p>
<p>7</p>		<p>Bagian dari aplikasi yang berikutnya adalah <i>about</i>. Bagian ini menampilkan informasi singkat tujuan dari pembuatan aplikasi dari penelitian ini.</p>

Pengujian Aplikasi

Pada aplikasi ini, pengujian dilakukan dengan menggunakan berbagai macam versi android pada tiga buah *smartphone android*, yang kemudian selanjutnya dilakukan pengujian berdasarkan jarak dari kamera ke objek wajah, untuk menguji seberapa jauh jarak yang dapat dideteksi oleh aplikasi ketika dilakukan pendeteksian wajah.

Tabel 1. Versi Android dan Jenis Perangkat *Smartphone*

No.	Versi Android	Perangkat
1.	8.1 (Oreo)	Vivo Y95
2.	9.0 (Pie)	Asus Zenfone Max Pro M2
3.	10.0 (Q)	Oppo F7

Tabel 2. Pengenalan Objek Wajah Pada Jarak 10 Cm

Versi Android	Keterangan
8.1 (Oreo)	Terdeteksi
9.0 (Pie)	Terdeteksi
10.0 (Q)	Terdeteksi

Tabel 3. Pengenalan Objek Wajah Pada Jarak 20 Cm

Versi Android	Keterangan
8.1 (Oreo)	Terdeteksi
9.0 (Pie)	Terdeteksi
10.0 (Q)	Terdeteksi

Tabel 4. Pengenalan Objek Wajah Pada Jarak 30 Cm

Versi Android	Keterangan
8.1 (Oreo)	Terdeteksi
9.0 (Pie)	Terdeteksi
10.0 (Q)	Terdeteksi

Tabel 5. Pengenalan Objek Wajah Pada Jarak 40 Cm

Versi Android	Keterangan
8.1 (Oreo)	Terdeteksi
9.0 (Pie)	Terdeteksi
10.0 (Q)	Terdeteksi

Tabel 6. Pengenalan Objek Wajah Pada Jarak 50 Cm

Versi Android	Perangkat
8.1 (Oreo)	Tidak Terdeteksi
9.0 (Pie)	Terdeteksi
10.0 (Q)	Terdeteksi

Tabel 7. Pengenalan Objek Wajah Pada Jarak 60 Cm

Versi Android	Perangkat
8.1 (Oreo)	Tidak Terdeteksi
9.0 (Pie)	Tidak Terdeteksi
10.0 (Q)	Terdeteksi

Tabel 8. Pengenalan Objek Wajah Pada Jarak 70 Cm

Versi Android	Perangkat
8.1 (Oreo)	Tidak Terdeteksi
9.0 (Pie)	Tidak Terdeteksi
10.0 (Q)	Tidak Terdeteksi

Untuk pengujian aplikasi selanjutnya, dilakukan juga pengujian dengan menggunakan tiga perangkat android, yaitu 8.1 (oreo), 9.0 (pie), 10.0 (Q), maka dapat diketahui bahwa pada android versi 8.1 (oreo), jarak maksimal agar perangkat dapat membaca objek wajah, yaitu ± 40 Cm. Pada android versi 9.0 (pie), jarak maksimal agar perangkat dapat membaca objek wajah, yaitu ± 50 cm. Pada versi android 10.0 (Q), jarak maksimal agar perangkat dapat membaca objek wajah, yaitu ± 60 Cm.

Dari percobaan dan pengujian yang dilakukan pada aplikasi, dilakukan 20 kali percobaan pengenalan wajah dengan 1 orang yang sama dan dengan 5 data citra wajah yang ada pada *database*. Pengujian dilakukan dengan bermacam - macam bentuk ekspresi wajah dan didapat persentase keberhasilan, yaitu 80% atau dari 20 kali pengujian, maka didapat 16 citra wajah yang dapat dikenali oleh sistem, sedangkan 4 citra wajah tidak dikenali.

4. Kesimpulan

Berdasarkan penelitian, didapatkan kesimpulan aplikasi, yaitu:

- a) Pada penelitian ini menghasilkan sebuah aplikasi yang dimaksudkan untuk memberikan perlindungan untuk data – data pribadi bagi pengguna *smartphone android* menggunakan pengenalan wajah dengan *eigenface*.
- b) Aplikasi ini dibangun menggunakan *eigenface* yang merupakan metode pengenalan dan pendeteksian wajah yang berdasarkan pada algoritma *Principal Component Analysis* (PCA), dimana *eigenface* tersebut memanfaatkan citra wajah, yang kemudian diekstraksi dan disimpan dalam *database*, dan kemudian data latih (*test image*) yang ada didefinisikan juga nilai *eigenfacenya*, kemudian

dilakukan perbandingan dengan *eigenface* dari foto atau gambar dalam *database*.

- c) Dalam pengujian aplikasi dilakukan pada tiga perangkat android yaitu 8.1 (oreo), 9.0 (pie), 10.0 (Q). Pada android versi 8.1(oreo), jarak maksimal agar perangkat dapat membaca objek wajah, yaitu ± 40 Cm. Pada android versi 9.0 (pie), jarak maksimal agar perangkat dapat membaca objek wajah, yaitu ± 50 cm. Pada versi android 10.0 (Q), jarak maksimal agar perangkat dapat membaca objek wajah, yaitu ± 60 Cm.
- d) Berdasarkan aplikasi Notepriv, presentase keberhasilan rata – ratanya adalah sebesar 80% atau dari 20 kali percobaan, maka terdapat 16 citra wajah yang dapat dikenali oleh sistem, sementara 4 citra wajah tidak dikenali.

5. Daftar Pustaka

- [1] Derisma, D., 2016. Sistem Pengenalan Wajah Secara Realtime Berbasis Android Menggunakan Metode *Eigenface* Pada OpenCV. Jurnal Komputer Terapan, 2(2), pp.127-136.
- [2] Wirdiani, N.K.A., Lattifia, T., Supadma, I.K., Mahar, B.J.K., Taradhita, D.A.N. and Fahmi, A., 2019. Real-Time Face Recognition with *Eigenface* Method. International Journal of Image, Graphics and Signal Processing (IJIGSP), 11(11), pp.1-9.
- [3] Suprianto, D. and Hasanah, R.N., 2014. Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, *Eigenface* PCA & MySQL. Jurnal Eccis, 7(2), pp.179-184.
- [4] Wiryadinata, R., Istiyah, U., Fahrizal, R., Priswanto, P. and Wardoyo, S., 2017. Sistem Presensi Menggunakan Algoritme *Eigenface* dengan Deteksi Aksesoris dan Ekspresi Wajah. Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI), 6(2), pp.222-229.
- [5] Singh, A. and Kumar, S., 2012. Face recognition using pca and eigen face approach (Doctoral dissertation).
- [6] Zein, A., 2018, July. Pendeteksian Multi Wajah Dan Recognition Secara Real Time Menggunakan Metoda Principal Component Analysis (Pca) Dan *Eigenface*. In ESIT (Vol. 12, No. 1, pp. 1-7).
- [7] Slavković, M. and Jevtić, D., 2012. Face recognition using *eigenface* approach. Serbian Journal of electrical engineering, 9(1), pp.121-130.
- [8] Abdillah, Z.N., 2018. Implementasi Pengenalan Wajah Dengan Metode *Eigenface* Pada Sistem Password Laptop. JURIKOM (Jurnal Riset Komputer), 5(3), pp.278-283.
- [9] Çarıkcı, M. and Özen, F., 2012. A face recognition system based on *eigenfaces* method. Procedia Technology, 1, pp.118-123.
- [10] Lata, Y.V., Tungathurthi, C.K.B., Rao, H.R.M., Govardhan, A. and Reddy, L.P., 2009. Facial recognition using *eigenfaces* by PCA. International Journal of Recent Trends in Engineering, 1(1), p.587.