

IMPLEMENTATION OF EIGENFACE ALGORITHM FOR IDENTIFICATION OF ANOPHELES IN SMARTPHONES

Alvianus Dengen*¹

¹Teknik Elektro, Fakultas Teknik, Universitas Teknologi Sulawesi, Indonesia
Email: alvianusdengen@utsmakassar.ac.id

(Naskah masuk: 10 Februari 2022, Revisi: 15 Februari 2022, diterbitkan: 25 Februari 2022)

Abstract

The development of smartphones as tools used in research provides many benefits in several fields, including education, health, and agriculture. Digital image research developed in mobile apps can help us choose the best decision so that the results are by what has been planned. One of the developments in entomology is entomology research that uses digital imagery. In the dataset activity, researchers used the anopheles's wings type as much as 22 image data. Fifteen types of Anopheles were used by the researcher with 100 experiments as test data. Image training and identification have the same process, taking pictures from a gallery or directly using the camera from an Android device. Preprocessing step in applications includes 1) converting image pixels to 100x100 pixels, then this pixel size becomes the standard of use in applications. 2) Convert an image to a binary using a threshold algorithm; this event is commonly called the Image Binaryization Process. Images derived from the threshold process are converted into one-dimensional vectors. Training Image and image identification using the Eigenface algorithm. The basic principle of the Eigenface algorithm is to quote the unique information of each image and then compare it to the image in the dataset. Identification with anopheles applications results in a good accuracy value with a success rate of 94.29%, with FMR value = 4.62% and FNMR = 2.78%.

Keywords: Android, Anopheles, Application, Eigenface Algorithm, Identification, Training Image.

IMPLEMENTASI ALGORITMA EIGENFACE UNTUK IDENTIFIKASI ANOPHELES DI SMARTPHONE

Abstrak

Pengembangan *smartphone* sebagai *tools* yang digunakan dalam penelitian memberikan banyak manfaat di beberapa bidang, termasuk pendidikan, kesehatan, dan pertanian. Penelitian citra digital yang dikembangkan dalam aplikasi seluler dapat membantu kita memilih keputusan terbaik sehingga hasilnya sesuai seperti apa yang sudah direncanakan. Salah satu perkembangan dalam bidang Entomologi adalah penelitian yang menggunakan citra digital. Dalam penelitian ini, saya dari anopheles yang digunakan untuk *dataset* sebanyak 22 jenis citra, untuk data uji penulis menggunakan 15 jenis citra dengan 100 kali percobaan. *Training Image* dan Identifikasi memiliki proses yang sama, mengambil gambar dari galeri atau langsung menggunakan kamera dari perangkat Android. *Preprocessing* dalam aplikasi meliputi: 1) mengubah *pixel* gambar menjadi 100x100 *pixel*, kemudian ukuran *pixel* ini menjadi standar penggunaan dalam aplikasi. 2) Konversi gambar ke biner menggunakan algoritma *threshold*, peristiwa ini biasa disebut Proses Binerisasi citra. Citra yang berasal dari proses *threshold*, diubah menjadi vektor satu dimensi. *Training Image* dan identifikasi citra memakai algoritma *Eigenface*. Prinsip utama algoritma *Eigenface* adalah menarik informasi yang unik dari setiap citra dan kemudian membandingkannya dengan citra dalam dataset. Identifikasi dengan aplikasi anopheles menghasilkan nilai akurasi yang baik dengan tingkat keberhasilan 94,29% dengan nilai FMR = 4,62% dan FNMR = 2,78%.

Kata kunci: Algoritma Eigenface, Android, Anopheles, Identifikasi, Training Image.

1. PENDAHULUAN

Anopheles merupakan salah satu genus nyamuk yang bisa membawa parasit *Protozoa Plasmodium* yang disebar oleh nyamuk *Anopheles* (*An.*) betina kepada manusia (CDC, 2017). Nyamuk *Anopheles* yang berperan sebagai vektor malaria di Indonesia

sekitar 22 spesies (Bonne, 1953). Identifikasi *Anopheles* sangatlah penting dilakukan karena spesies dari *Anopheles* memiliki perbedaan kapasitas vektorial sehingga proses identifikasi nyamuk *Anopheles* merupakan langkah penting untuk

mengenal karakteristik - karakteristik yang dimiliki oleh berbagai jenis *Anopheles*. [6][20][32]

Teknologi aplikasi melalui telepon pintar sudah mampu memberikan tampilan informasi yang menarik, selain itu kelebihan telepon pintar juga adalah fleksibel yang bisa digunakan di mana saja. [11] Sistem pada Identifikasi nyamuk *Anopheles* ini tidak mudah dilakukan mengingat citra masukannya cukup rumit serta ukurannya sangat kecil. Dalam penelitian ini, data masukan yang dipakai adalah sayap dari *Anopheles* yang mempunyai ciri-ciri tertentu supaya proses Ekstraksi ciri berfungsi memperjelas ciri-ciri sehingga mudah dalam melakukan identifikasi. [6][20][32]

Pengenalan Objek dan grafika dilakukan beberapa pendekatan untuk pemrograman *mobile* berdasarkan pada citra tanpa penggunaan model tiga dimensi, Pendekatan yang sesuai adalah menggunakan algoritma *Eigenface*. [7] Algoritma ini cukup sederhana, citra diekstraksi kemudian informasi yang terkandung dalam citra tersebut yang berupa sekumpulan variasi-variasi penting dalam citra untuk digunakan sebagai informasi untuk mengkodekan dan membandingkan citra nyamuk *Anopheles*. [10][15] Untuk mendukung penelitian ini, aplikasi komputer yang digunakan adalah Android Studio baik untuk perancangan tatap mukanya dan penulisan algoritma *Eigenface*. Pada Penelitian ini ada dua proses yang dibuat, *Training Image* dan Identifikasi. Data masukan dengan format JPG (.jpg), Citra ini melewati proses preprocessing untuk kebutuhan aplikasi kemudian citra tersebut akan di konversi dari format RGB (*red green blue*) kemudian mengatur derajat keabuan dengan algoritma *Threshold* sehingga memudahkan dalam proses *training image* dan identifikasi. Dataset sebagai data pembanding adalah hasil dari proses *Training Image* yang disimpan ke dalam bentuk XML (.xml), jenis *Anopheles* disimpan ke jenis.xml dan nilai *biner* dari citra hasil proses *training image* disimpan ke dalam *biner.xml*. Tujuan dari penelitian ini adalah mengidentifikasi *Anopheles* dengan bantuan aplikasi

2. METODE PENELITIAN

Beberapa penelitian terkait memuat tentang Identifikasi dengan pengenalan pola menggunakan metode *Eigenfaces*. Proses identifikasi Proses identifikasi wajah adalah yang paling populer, penulis disini mencoba mengambil beberapa penelitian terkait yang memuat pengenalan wajah/objek dengan pendekatan metode *Eigenfaces*, antara lain:

Menurut Penelitian Müge Çarıkçı a, Figen Özen yang berjudul “A Face Recognition Ssystem Based on Eigenface Method” metode yang digunakan adalah metode *Eigenface*. Strategi dari metode *Eigenfaces* yang digunakan dalam penelitian ini adalah ekstraksi dari fitur karakteristik wajah yang mewakili wajah tersebut sebagai

kombinasi linear yang disebut “*Eigenfaces*” yang merupakan hasil ekstraksi. [7]

Suwanto Sanjaya dan Fadhilah Syafria yang berjudul “Sistem Pakar Identifikasi Nyamuk menggunakan Pohon Keputusan (Studi Kasus: Nyamuk *Anopheles* Betina asal oriental di Indoensia) Metode yang digunakan Pohon Keputusan karena buku kunci Identifikasi nyamuk memiliki aturan-aturan seperti pohon keputusan. [32]

Menurut penelitian yang dilakukan Guillaume Dave, Xing Chao, Kishore Sriadibhatla dengan judul “Face Recognition on Mobile Phones” Penulis menyelidiki berbagai algoritma untuk pengenalan wajah pada ponsel. Langkah pertama dalam sistem pengenalan wajah adalah deteksi wajah. Penulis menggunakan algoritma segmentasi warna dan template matching untuk deteksi wajah, eigen dan fisher wajah untuk pengenalan wajah. Implementasi Algoritma menggunakan MATLAB dan kemudian diimplementasikan pada ponsel DROID. Kemudian Penulis membuat tradeoff antara akurasi dan kompleksitas komputasi algoritma terutama karena kami menerapkan sistem pengenalan wajah pada ponsel dengan kemampuan hardware yang terbatas. [11]

Penelitian yang dilakukan oleh Maria De Marsico, Chiara Galdi, Michele Nappi, Daniel Riccio dengan Judul “FIRME: Face and Iris Recognition for Mobile Engagement” menjelaskan bahwa FIRME adalah aplikasi pengenalan Wajah dan Iris berbasis multimodel yang didesain untuk aplikasi mobile, desain dan implementasi FIRME mengandalkan arsitektur amodular. Alur kerjanya merupakan Paket terpisah dan diganti. Yang pertama menangani akuisisi citra, dari titik ini, kemudian cabang berbeda melakukan deteksi, segmentasi, ekstraksi fitur, dan mencocokkan untuk wajah dan iris secara terpisah, untuk wajah dilakukan antispoofing juga dilakukan setelah segmentasi kemudian dari dua cabang berbeda menyatu. FIRME dapat melakukan reidentifikasi dan pemilihan sampel terbaik. [9]

Penelitian dengan Judul “XFace: A Face Recognition system for Android Mobile Phones” yang ditulis oleh Jiawei Hu, Liangrui Peng, dan Li Zheng menjelaskan Aplikasi Xface diusulkan pendekatan pengenalan wajah termasuk deteksi wajah, deteksi mata, preprocessing untuk ROI (*Region of Interest*), LBP (*Local Binary Pattern*) ekstraksi fitur, Fitur Pengurangan dimensi berdasarkan PCA (*Principal Component Analysis*) dan classifier jarak minimum. Penulis dalam penelitian kemudian membandingkan hasil dengan menggunakan Metode *Eigenface* dan *Fisherfaces*. [13]

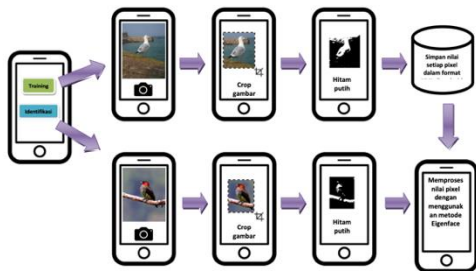
Pengembangan perangkat lunak menurut Suryan (2014), menggunakan model sekuensial linear, sebuah pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial yang mulai dari *system level* terus maju ke analisis, desain,

implementasi dan pengujian.[31] Model ini melingkupi aktivitas sebagai berikut:

1. *Feasibility*
 Pada tahap ini membangun syarat dari semua elemen sistem dalam aplikasi dan mengalokasikan beberapa subset dari kebutuhan aplikasi.
2. *Requirement (Analisis Kebutuhan)*
 Proses pada tahap ini yakni mengumpulkan kebutuhan sistem secara lengkap kemudian dianalisis dan didefinisikan sesuai dengan kebutuhan sistem yang harus dipenuhi oleh aplikasi yang akan dibangun. Tahap ini dikerjakan secara lengkap untuk hasil desain yang lengkap.
3. *Design*
 Pada tahap desain aplikasi, merupakan proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda yakni data, arsitektur software, representasi interface, dan detail Algoritma prosedural.
4. *Implementasi*
 Pada tahap ini desain yang sudah ada kemudian diterjemahkan kedalam kode-kode dengan menggunakan bahasa yang sudah ditentukan. Program yang telah dibangun langsung diuji baik secara unit.
5. *Test*
 Proses pengujian berfokus pada logika internal software, penyatuan unit-unit program kemudian diuji secara keseluruhan (*System Testing*).

2.1. Perancangan Aplikasi

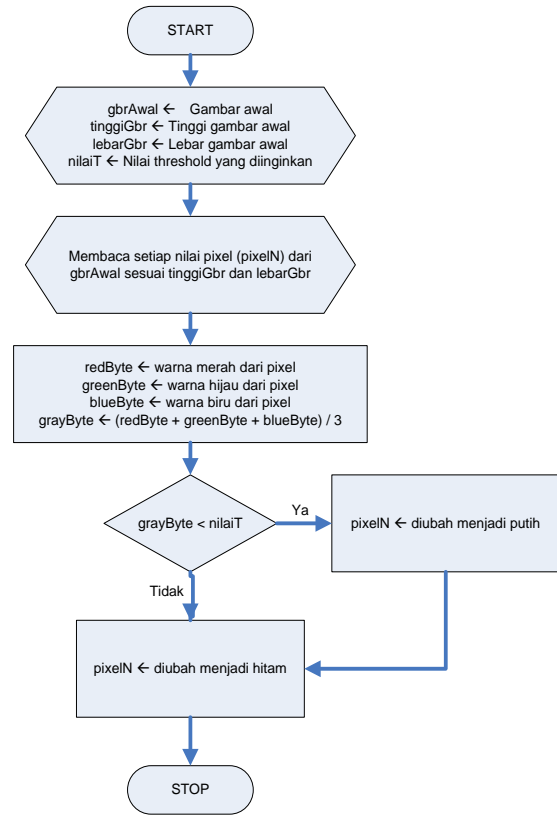
Dalam perancangan aplikasi Anopheles ini secara garis besarnya sebagai berikut:



Gambar 1. Rancangan Aplikasi Anopheles

Aplikasi Anopheles terdiri dari dua proses yang akan dirancang, Training Image dan Identifikasi. Pada kedua proses ini, melalui tahap *preprocessing*. *Preprocessing* adalah tahap dimana data masukan diproses sesuai kebutuhan aplikasi dimana dimensi gambar masukkan diubah menjadi 100x100 pixel. Tahap selanjutnya adalah proses Binerisasi citra, Proses ini mengubah citra masukan berwarna menjadi gambar biner yang berwarna hitam putih

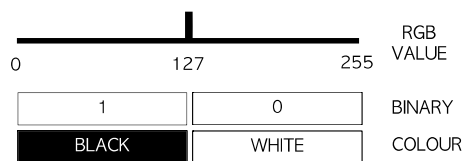
dengan proses menggunakan *Threshold*. Proses *Threshold* ini digambar dalam flowchart dibawah ini:



Gambar 2. Flowchart *Threshold*

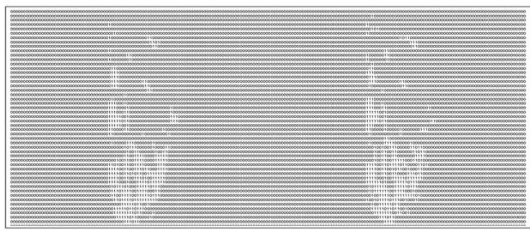
Threshold merupakan batas yang menentukan pixel akan diubah menjadi hitam atau putih dalam binerisasi. [31] Citra masukan yang akan diproses dimuat terlebih dahulu di *memory* kemudian dibaca tinggi dan lebarnya ke dalam variabel. Nilai *threshold* yang ditentukan disimpan ke dalam variabel. Nilai pixel yang awalnya bertipe integer diubah menjadi RGB (*red-green-blue*). Nilai RGB bertipe byte yang memiliki nilai dari 0 sampai 255. Nilai pixel yang bertipe integer akan didapat nilai RGB dimana setiap nilainya tidak lebih dari 0 sampai 255, ketiga nilai ini kemudian dijumlahkan dan dibagi tiga untuk mendapatkan nilai rata-ratanya.

```
R <- 0xFFand nilaiPixel
G <- (0xFF00and nilaiPixel) DIV 256
B <- (0xFF0000and nilaiPixel) DIV 65536
```



Gambar 3. Nilai *threshold* untuk menentukan nilai ambang batas

Hasil dari proses *threshold* akan menghasilkan sebuah flatvector berupa string yang hanya berisi nilai biner 0 dan 1, seperti gambar berikut:

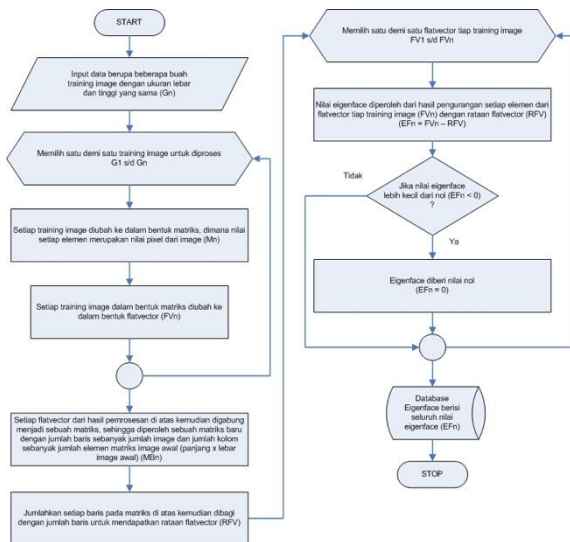


Gambar 4. Contoh Flatvector

2.2. Algoritma Eigenface

Implementasi secara keseluruhan algoritma Eigenface sederhana, Citra hasil training direpresentasikan menjadi flatvector (gabungan vector) dan digabung secara bersama-sama menjadi sebuah matriks tunggal. Nilai Eigenface masing-masing citra kemudian di ekstraksi dan disimpan dalam file temporary atau database. Test Image yang diidentifikasi kemudian nilai eigenface-nya dibandingkan dengan nilai eigenface dalam database atau file temporary. [7][10][15] [23][24][25][28]

Algoritma Eigenface pada proses Training Image digambarkan dalam flowchart dibawah ini:



Gambar 5. Flowchart Pembentukan Training Image dengan Algoritma Eigenface

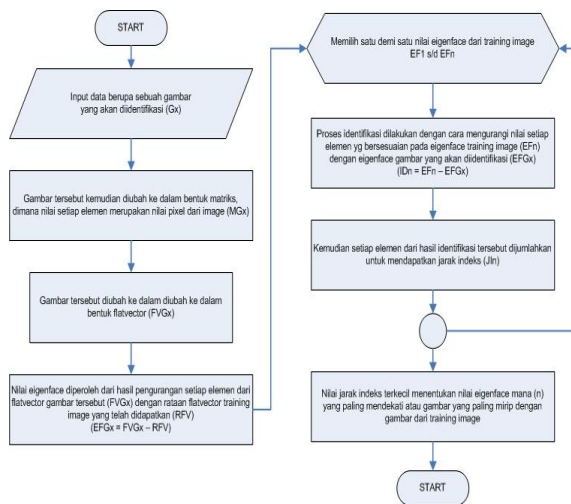
Proses pembentukan database Eigenface dijabarkan sebagai berikut:

1. Input data berupa beberapa buah training image dengan ukuran lebar dan tinggi yang sama (Gn)
2. Memilih satu demi satu training image untuk diproses G1 s/d Gn
3. Setiap training image diubah ke dalam bentuk matriks, dimana nilai setiap elemen merupakan nilai pixel dari image (Mn)
4. Setiap training image dalam bentuk matriks diubah ke dalam bentuk flatvector (FVn)
5. Setiap flatvector dari hasil pemrosesan di atas kemudian digabung menjadi sebuah matriks, sehingga diperoleh sebuah matriks baru dengan jumlah baris sebanyak jumlah image dan

jumlah kolom sebanyak jumlah elemen matriks image awal (panjang x lebar image awal) (MBn)

6. Jumlahkan setiap baris pada matriks di atas kemudian dibagi dengan jumlah baris untuk mendapatkan rata-ran flatvector (RFV)
7. Memilih satu demi satu flatvector tiap training image FV1 s/d FVn
8. Nilai eigenface diperoleh dari hasil pengurangan setiap elemen dari flatvector tiap training image (FVn) dengan rata-ran flatvector (RFV) ($EF_n = FV_n - RFV$)
9. Jika nilai eigenface lebih kecil dari nol ($EF_n < 0$), maka Eigenface diberi nilai nol ($EF_n = 0$)
10. Simpan nilai eigenface ke dalam database (EFn)

Proses Identifikasi dengan algoritma eigenface dijabarkan dalam flowchart dibawah ini:



Gambar 6. Flowchart Identifikasi dengan Algoritma Eigenface

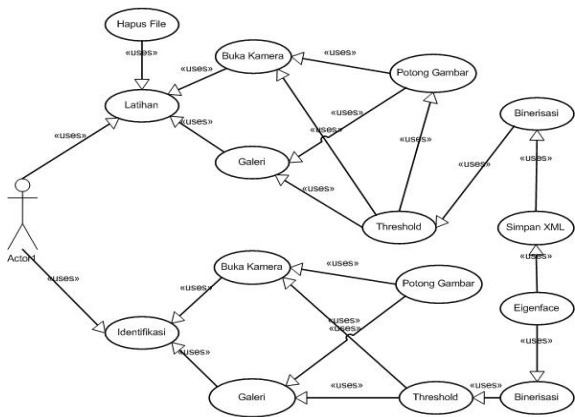
Penjelasan algoritma eigenface pada proses identifikasi sebagai berikut:

1. Input data berupa sebuah gambar yang akan diidentifikasi (Gx)
2. Gambar tersebut kemudian diubah ke dalam bentuk matriks, dimana nilai setiap elemen merupakan nilai pixel dari image (MGx)
3. Gambar tersebut diubah ke dalam bentuk flatvector (FVGx)
4. Nilai eigenface diperoleh dari hasil pengurangan setiap elemen dari flatvector gambar tersebut (FVGx) dengan rata-ran flatvector training image yang telah didapatkan (RFV) ($EFG_x = FVG_x - RFV$)
5. Memilih satu demi satu nilai eigenface dari training image EF1 s/d EFn
6. Proses identifikasi dilakukan dengan cara mengurangi nilai setiap elemen yg bersesuaian pada eigenface training image (EFn) dengan eigenface gambar yang akan diidentifikasi (EFGx) ($ID_n = EFn - EFG_x$)

7. Kemudian setiap elemen dari hasil identifikasi tersebut dijumlahkan untuk mendapatkan jarak indeks (JIn)
8. Nilai jarak indeks terkecil menentukan nilai eigenface mana (n) yang paling mendekati atau gambar yang paling mirip dengan gambar dari training image

2.3. Use Case Diagram

Diagram Use Case aplikasi anopheles ditunjukkan seperti gambar berikut:

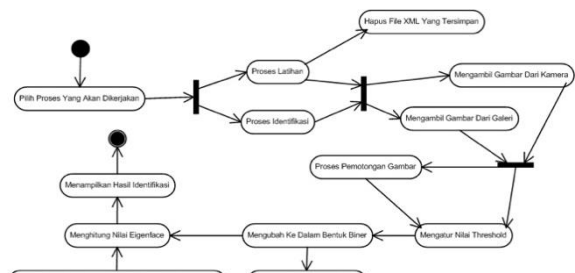


Gambar 7. Use Case Diagram

Diperlihatkan pada gambar di atas use case yang dirancang untuk pembuatan aplikasi “Penerapan Algoritma Eigenface pada aplikasi Identifikasi Anopheles”. Use case yang akan dirancang terdapat menu Training Image/Latihan dan Identifikasi.

2.4. Activity Diagram

Diagram aktivitas ditunjukkan seperti gambar berikut:



Gambar 8. Use Case Diagram

Activity diagram menjelaskan alur penggunaan sistem pada aplikasi dari user. Dari activity diagram ini dijelaskan bahwa aplikasi maka akan ada tampilan dua menu utama/ Pada activity ini pengguna memilih diantara dua menu utama, jika memilih Latihan/Training Image maka proses akan berakhir dengan menyimpan data kedalam database. Jika memilih Identifikasi, proses akhir berakhir dimana image masukan akan diidentifikasi.

2.5. Sumber Data

Data yang digunakan berasal dari laboratorium entomologi Universitas Hasanuddin. Data yang dapat dikumpulkan berupa gambar-gambar nyamuk Anopheles dari beberapa jenis seperti *Barbistrostris*, *Barbumbrosus*, *Farauti*, *Flavistrostris*, *Hyrceanus*, *Kochi*, *Indefinitus*, *Koliensis*, *Longirostris*, *Maculatus*, *Pseudobarbistrostris*, *Punculatus*, *Subpictus*, *Sundaicus*, *Tesselatus*, dan *Vagus*. Gambar nyamuk yang dimasukkan sebagai data training berupa bagian-bagian tubuh dari nyamuk yang dianggap tidak cacat dan bukan merupakan bagian tubuh yang dapat digerakkan seperti tungkai kaki dikarenakan posisi dari gambar juga sangat mempengaruhi hasil identifikasi. Sayap nyamuk Anopheles menjadi data training yang dianggap paling mudah dilakukan proses identifikasi dan sebagai data set uji untuk proses identifikasi.



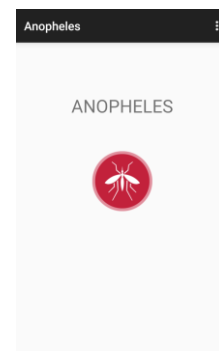
Gambar 9. Contoh Sayap Anopheles

3. HASIL DAN PEMBAHASAN

3.1. Perancangan Aplikasi

3.1.1 Tampilan Splash Screen

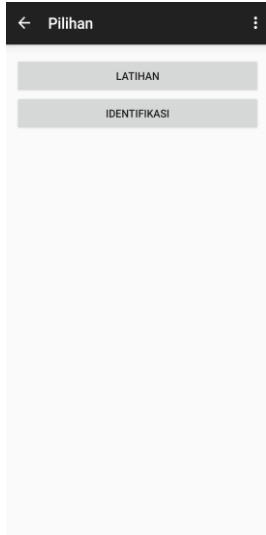
Ketika mengetuk icon aplikasi yang nantinya akan muncul pertama adalah splash screen. Splash screen dalam aplikasi Anopheles ini berupa logo yang nantinya ketika diketuk akan masuk kedalam menu utama, tampilan splash screen seperti pada gambar berikut.



Gambar 10. Tampilan Splash Screen

3.1.2 Tampilan Menu Utama

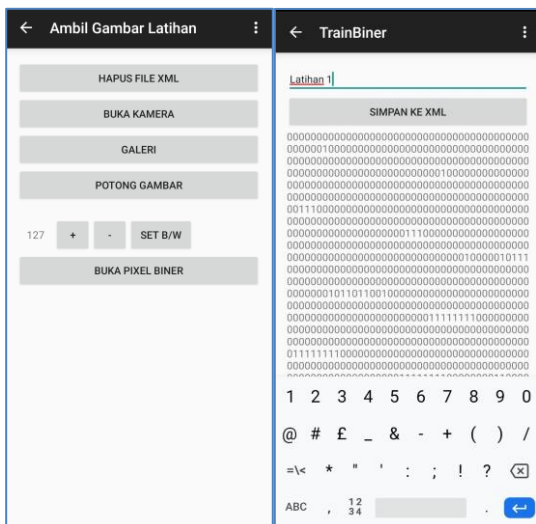
Selanjutnya adalah menu utama. Dalam aplikasi Anopheles yang akan telah dirancang, terdapat 2 menu utama yakni Menu Latihan/Training Image dan Menu Identifikasi., seperti yang ditunjukkan pada gambar 11.



Gambar 11. Tampilan Menu Utama

3.1.3 Tampilan Menu Latihan/Training Image

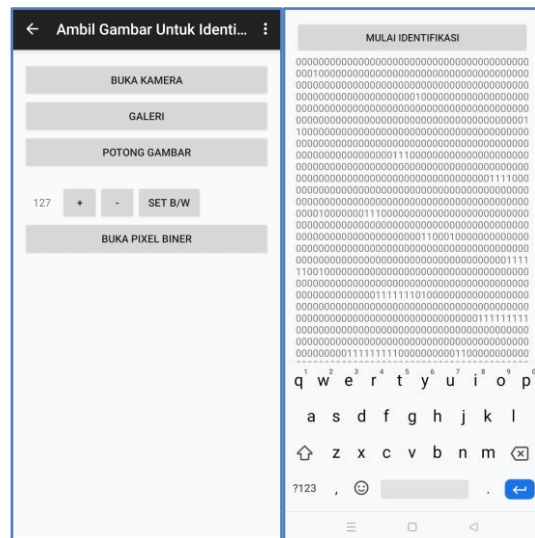
Dari layer utama, ketika tombol Latihan yang dipilih. Maka akan muncul layer Latihan yang digunakan untuk pengambilan citra masukan kemudian disimpan kedalam dataset .xml yang berisi dua jenis .xml, biner.xml menyimpan nilai-nilai biner dari gambar uji dan nama.xml menyimpan nama-nama gambar training image. Dari form latihan, ketika tombol Buka Pixel Biner dipilih maka akan muncul layer Train biner yang terdiri dari Text View untuk menulis nama jenis Anopheles, tombol Simpal Ke XML serta menampilkan flatvector berbentuk string.



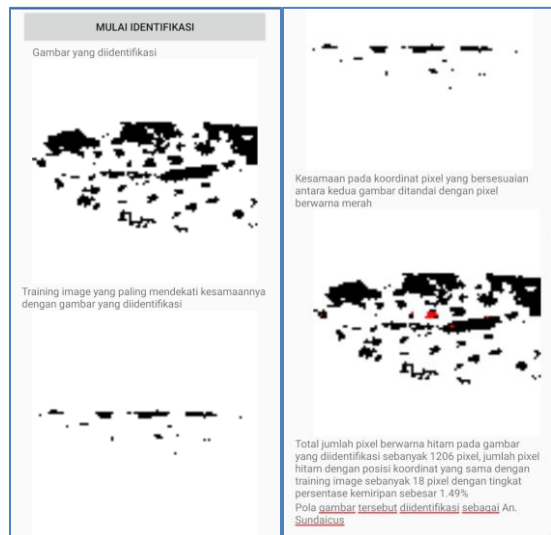
Gambar 12. Tampilan Latihan/Training Image & Layer Train Biner

3.1.4 Tampilan Menu Identifikasi

Pada menu identifikasi terdiri dari 3 layer, ketika menekan tombol identifikasi, layer pertama yang muncul adalah layer untuk memproses data masuk untuk identifikasi. Ketika menekan tombol Buka Pixel Biner, maka muncul layer kedua, layer ini berisi nilai flatvector berupa string biner dan tombol Mulai Identifikasi.



Gambar 13. Tampilan Menu Identifikasi, Layer Pertama, Layer Kedua.



Gambar 14. Tampilan Menu Identifikasi, Layer Ketiga

3.1.5 Hasil Pengujian Aplikasi

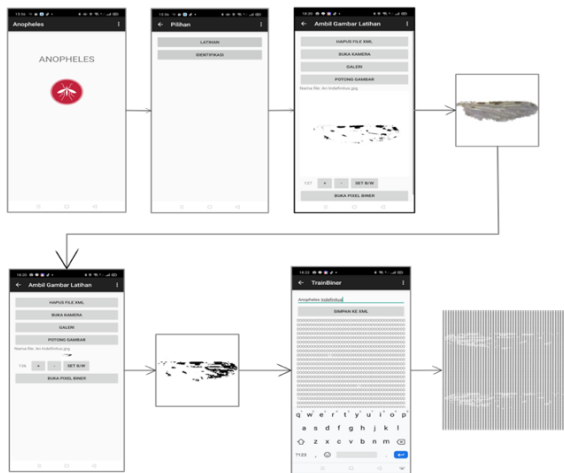
Hasil pengujian aplikasi dilakukan dengan memasukkan citra (*image*) yang akan dijadikan database aplikasi. Citra ini berasal dari laboratorium Entomologi Universitas Hasanuddin yang terdiri dari bagian sayap da abdomen, Pada aplikasi ini, dua menu dirancang yakni:

- Training Imgae, citra masukkan akan diolah sesuai prosedur yang dirancang dalam aplikasi sehingga sistem mempunyai pengetahuan

- Identifikasi, citra masukan berupa sayap dan abdomen yang penulis potong sesuai aturan peneliti dari laboratorium Entomologi. Citra ini kemudian akan diambil gambarnya dan dijadikan data identifikasi

3.1.5.1 Pengujian Proses Training Image di Hp Android

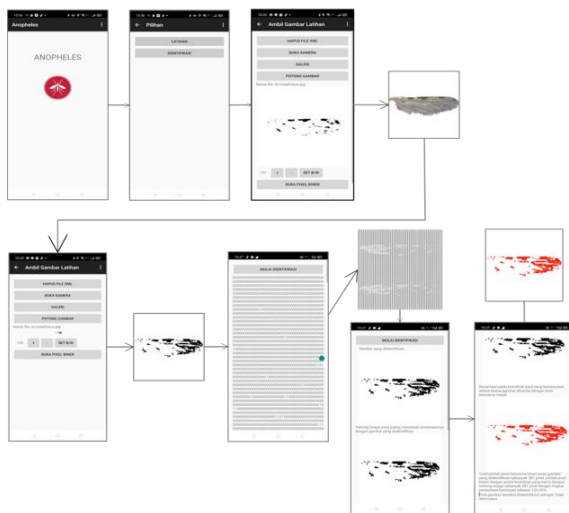
Untuk pengujian proses Training Image pada aplikasi identifikasi *Anopheles* dapat dilihat pada gambar 15. Proses training image melalui preprocessing sampai simpan data ke database.



Gambar 15. Proses Training Image pada aplikasi Android

3.1.5.2 Pengujian Proses Identifikasi di Hp Android

Proses identifikasi pada aplikasi *Anopheles* dimulai dari preprocessing, merubah citra ke citra biner, pengaturan nilai *threshold* dalam aplikasi, kemudian menampilkan *flatvector*. Proses Identifikasi bisa dilihat pada gambar 16 berikut.



Gambar 16. Proses Identifikasi

Hasil identifikasi dimana citra masukan dibandingkan dengan citra yang sudah ada dalam

database kemudian hasil perbandingan bisa dilihat pada gambar 14 hasil identifikasi terdapat garis-garis merah yang merupakan vector yang menunjukkan titik-tik tersebut merupakan kesamaan citra antara citra masukan dan citra didalam database, keterangan jumlah pixel yang sesuai dan jumlah presentasi kemiripan terdapat dibawah gambar hasil identifikasi.

3.1.6 Hasil Percobaan

Tingkat keberhasilan sistem pada aplikasi untuk mengidentifikasi nyamuk *Anopheles* dengan menggunakan metode *Eigenface* diperoleh dengan perhitungan *False Match Rate (FMR)* dan *False Non Match Rate (FNMR)*. [14][18] FMR adalah suatu probabilitas sampel yang berasal dari luar sistem tetap cocok dengan acuan data yang ada dalam *dataset*. FNMR adalah probabilitas sampel yang sama dari *dataset* tetap tidak cocok dengan acuan yang terdaftar pada *dataset*.

FMR dan FNMR adalah suatu fungsi dari nilai *threshold*. Jika nilai *threshold* diatas ambang batas (127), membuat sistem lebih bisa toleran terhadap variasi input aplikasi maka FNMR naik. Sebaliknya, jika nilai dibawah ambang batas atau lebih kecil, akan membuat sistem lebih baik dalam identifikasi tetapi bisa menolak data masukan yang sama dengan yang ada didalam *dataset*.

$$FMR = \frac{\text{Jumlah Kesalahan Cocok}}{\text{Jumlah Keseluruhan Proses Pencocokkan}} \times 100\% \quad (1)$$

$$FNMR = \frac{\text{Jumlah Kesalahan Tidak Cocok}}{\text{Jumlah Keseluruhan Proses Pencocokkan}} \times 100\% \quad (2)$$

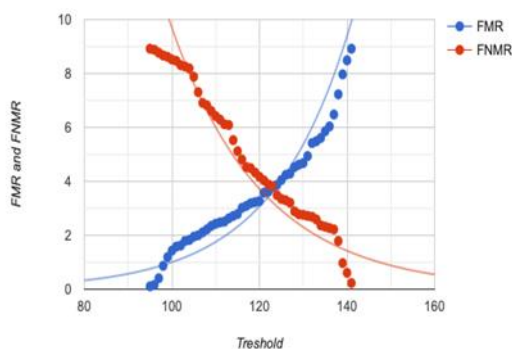
Pada persamaan (a) dan (b) memperlihatkan laju hasil perhitungan dalam eksperimen tentang kesalahan sistem dalam aplikasi menerima data masukan sebagai data uji yang tidak ada dalam *dataset* dan menolak data masukan sama dengan yang ada di dalam *dataset*. Dari hasil perhitungan tersebut akan memberikan hasil akurasi pengujian.

Pada proses perhitungan hasil percobaan dengan menggunakan FMR dan FNMR, percobaan dilakukan dengan menggunakan 15 sampel sayap *Anopheles*. Data masukan atau gambar berupa sayap dari nyamuk *anopheles* diambil langsung dari aplikasi *Anopheles*.

Pengujian dilakukan dengan *threshold* terendah 95 dan tertinggi 141. Penulis menggunakan 22 jenis citra nyamuk *anopheles* yang didapatkan dari Laboratorium Entomologi untuk dijadikan *dataset* dan 15 sampel citra uji dan 100 kali percobaan. Hasil pengujian menunjukkan dengan *threshold* 129 merupakan hasil terbaik dengan presentasi akurasinya adalah 94,29%. FMR dalam sistem sama dengan 4,62% dan FNMR sama dengan 2,78%. Sistem memiliki tingkat akurasi yang ditolak sekitar 95,38% dan tingkat akurasi yang diterima sekitar 97,22%.

Tabel 1. Hasil Pengujian

No.	THRESHOLD	FMR (%)	FNMR (%)	ACCURACY
1	95	0,11	8,92	73,11
2	96	0,16	8,88	73,56
3	97	0,41	8,77	74,02
4	98	0,87	8,67	74,21
5	99	1,19	8,62	74,70
6	100	1,43	8,52	75,09
7	101	1,58	8,47	75,56
8	102	1,62	8,31	76,96
9	103	1,79	8,26	77,92
10	104	1,83	8,19	78,53
11	105	1,95	7,88	79,01
12	106	2,00	7,31	79,20
13	107	2,11	6,91	80,16
14	108	2,22	6,82	81,47
15	109	2,35	6,61	82,15
16	110	2,43	6,42	82,96
17	111	2,48	6,29	83,21
18	112	2,51	6,12	83,64
19	113	2,63	6,09	84,92
20	114	2,72	5,53	85,12
21	115	2,79	5,12	85,91
22	116	3,02	4,81	86,32
23	117	3,09	4,53	87,50
24	118	3,19	4,49	88,56
25	119	3,23	4,33	89,23
26	120	3,27	4,17	90,88
27	121	3,58	4,04	91,48
28	122	3,62	3,89	91,93
29	123	3,79	3,82	92,19
30	124	3,87	3,49	93,14
31	125	4,05	3,35	93,89
32	126	4,24	3,30	94,18
33	127	4,29	3,21	94,01
34	128	4,53	2,89	94,06
35	129	4,62	2,78	94,29
36	130	4,68	2,76	93,88
37	131	4,94	2,72	92,81
38	132	5,42	2,69	91,28
39	133	5,49	2,59	91,95
40	134	5,61	2,37	90,78
41	135	5,86	2,32	90,11
42	136	6,04	2,27	89,88
43	137	6,48	2,22	87,67
44	138	7,23	1,79	86,79
45	139	7,97	0,97	85,81
46	140	8,49	0,61	84,33
47	141	8,92	0,23	83,85



Gambar 17. Grafik hasil Pengujian dengan FMR dan FNMR

4. KESIMPULAN

Dalam jurnal ini telah dilakukan penelitian dan perancangan aplikasi identifikasi anopheles berbasis android dengan menggunakan metode Eigenface. Aplikasi anopheles dibangun dalam pemrograman

java untuk android dan diterapkan pada platform android. Pada percobaan sistem identifikasi anopheles ini diperoleh hasil dengan akurasi ditolak sekitar 95,38% dan tingkat akurasi diterima sekitar 97,22%. Pada hasil percobaan dengan melihat tingkat akurasi yang dicapai bahwa aplikasi anopheles dengan metode Eigenface berbasis android ini menghasilkan nilai akurasi yang baik dengan tingkat keberhasilan 94,29% dengan nilai FMR = 4,62% dan FNMR = 2,78%.

Pada aplikasi anopheles memiliki dua menu, Training Image dan Identifikasi. Kedua menu tersebut memiliki proses yang sama yakni mulai dari Import Image dari gallery atau capture image langsung dari kamera smartphone kemudian memasuki tahap preprocessing mulai dari resize image, pengaturan threshold, kemudian merubah image ke biner. Proses akhir dari Training image adalah simpan image masukan yang sudah dikelola ke dalam dataset sedangkan Identifikasi menampilkan hasil proses Identifikasi beserta gambar pembandingan dari dataset.

DAFTAR PUSTAKA

- [1] R. I. Afida, "Deteksi Dini Osteoporosis Dengan Neuro-Fuzzy System Melalui Anatomic Index Dari Citra Dental Panoramic Radiograph Pada Area Tulang Mandible," UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG, 2013.
- [2] R. N. Aini and A. Riyantomo, "Aplikasi Pembelajaran Bahasa Inggris Bersama 'Transpofun' Berbasis Android," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 68–72, 2019.
- [3] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, no. 1, pp. 152–160, 2018.
- [4] R. H. Ariesdianto, Z. E. Fitri, A. Madjid, and A. M. N. Imron, "Identifikasi Penyakit Daun Jeruk Siam Menggunakan K-Nearest Neighbor," *J. Ilmu Komput. dan Inform.*, vol. 1, no. 2, pp. 133–140, 2021.
- [5] F. A. Bachtiar *et al.*, "Sistem Pendukung Keputusan Seleksi Penentuan Penerima Beasiswa Dengan Metode Simple Additive Weighting (SAW)," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 68–72, Jan. 2019.
- [6] Biomedis Balitbang, *Peta Anopheles di Tanah Papua*. Kementerian Kesehatan R.I., Badan Penelitian dan Pengembangan Kesehatan, Balai Penelitian dan Pengembangan Biomedis Papua, 2015.

- [7] M. üg. Çarıkçı and F. Özen, "A Face Recognition System Based on Eigenfaces Method," *Procedia Technol.*, vol. 1, pp. 118–123, 2012.
- [8] K. Choi, K. A. Toh, and H. Byun, "Realtime training on mobile devices for face recognition applications," *Pattern Recognit.*, vol. 44, no. 2, pp. 386–400, Jan. 2011.
- [9] M. De Marsico, C. Galdi, M. Nappi, and D. Riccio, "FIRME: Face and iris recognition for mobile engagement," *Image Vis. Comput.*, vol. 32, no. 12, pp. 1161–1172, 2014.
- [10] J. Efendi, M. I. Zul, and W. Yunanto, "Real time face recognition using eigenface and viola-jones face detector," *Int. J. Informatics Vis.*, vol. 1, no. 1, pp. 16–22, 2017.
- [11] D. Guillaume, C. Xing, and S. Kishore, "Face Recognition in Mobile Phones," *ResearchGate*, vol. 24, no. 17, pp. 1–8, 2010.
- [12] Y. Guo, S. Han, Y. Li, C. Zhang, and Y. Bai, "K-Nearest Neighbor combined with guided filter for hyperspectral image classification," in *Procedia Computer Science*, 2018, vol. 129, pp. 159–165.
- [13] J. Hu, L. Peng, and L. Zheng, "XFace: A Face Recognition System for Android Mobile Phones," in *Proceedings - 3rd IEEE International Conference on Cyber-Physical Systems, Networks, and Applications, CPSNA 2015*, 2015, pp. 13–18.
- [14] V. Jalaja and G. S. G. N. Anjaneyulu, "Face recognition by using eigen face method," *Int. J. Sci. Technol. Res.*, vol. 9, no. 3, pp. 961–966, 2020.
- [15] A. JAMHARI, "A Perancangan Sistem Pengenalan Wajah Secara Real-Time pada CCTV dengan Metode Eigenface:," *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 2, no. 2, pp. 20–32, 2020.
- [16] A. JANNAT, U. H. ABU, N. TAHREEM, and M. TAYYABA, "a Review on Software Testing and Its Methodology," *i-manager's J. Softw. Eng.*, vol. 13, no. 3, p. 32, 2019.
- [17] A. Kaur, "An Essential Guide to Automated GUI Testing Of Android Mobile Applications," *Int. J. Comput. Tech.*, vol. 2, no. 6, pp. 8–12, 2015.
- [18] K. E. Kim *et al.*, "Hand grip pattern recognition for mobile user interfaces," *Proc. Natl. Conf. Artif. Intell.*, vol. 2, pp. 1789–1794, 2006.
- [19] Y. I. Kurniawan, A. Rahmawati, N. Chasanah, and A. Hanifa, "Application for determining the modality preference of student learning," in *Journal of Physics: Conference Series*, 2019, vol. 1367, no. 1, pp. 1–11.
- [20] S. Lestari, A. Adrial, and R. Rasyid, "Identifikasi Nyamuk Anopheles Sebagai Vektor Malaria dari Survei Larva di Kenagarian Sungai Pinang Kecamatan Koto XI Tarusan Kabupaten Pesisir Selatan," *J. Kesehat. Andalas*, vol. 5, no. 3, pp. 656–660, 2016.
- [21] Y. Li, C. Xia, and J. Lee, "Detection of small-sized insect pest in greenhouses based on multifractal analysis," *Optik (Stuttg.)*, vol. 126, no. 19, pp. 2138–2143, 2015.
- [22] C. H. Low, "NSL-KDD Dataset." 2015.
- [23] N. Maharani Raharja, M. Arief Fathansyah, and A. Nur Nazilah Chamim, "Vehicle Parking Security System with Face Recognition Detection Based on Eigenface Algorithm," *J. Robot. Control*, vol. 3, no. 1, pp. 78–85, 2021.
- [24] P. S. Ramadhan, S. Nurarif, M. Syahril, Y. Riani, and N. Gulo, "Teknologi Biometrik Menggunakan Algoritma Eigenface Biometric Technology Using Eigenface Algorithm," *J. Comput. Eng. Syst. Sci.*, vol. 7, no. January, pp. 43–54, 2022.
- [25] R. Rosnelly, M. S. Simanjuntak, A. Clinton Sitepu, M. Azhari, S. Kosasi, and Husen, "Face Recognition Using Eigenface Algorithm on Laptop Camera," in *2020 8th International Conference on Cyber and IT Service Management, CITSM 2020*, 2020, pp. 1–4.
- [26] A. B. Salat, "Pembuatan Alat Pendeteksi Kebisingan Untuk Budidaya Burung Love Bird Berbasis Arduino Dan Android Melalui Wifi," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 106–114, 2019.
- [27] S. Putra, I. Fitri, and S. Ningsih, "Absensi Pengenalan Wajah Menggunakan Menggunakan Algoritma M Σ ," *J. Appl. Informatics Comput.*, vol. 5, no. 1, pp. 21–27, 2021.
- [28] U. Sari and K. Budayawan, "Implementasi Metode Eigenface pada Sistem Absensi Wajah Berbasis PHP dan MySQL," *Voteteknika (Vocational Tek. Elektron. dan Inform.)*, vol. 9, no. 3, p. 111, 2021.
- [29] P. A. Sasmito, Ilhamsyah, and R. P. Sari, "Sistem Pendukung Keputusan Penerima Beasiswa Dengan Menerapkan Metode Simple Additive Weighting (SAW)," Universitas Muhammadiyah Surakarta, 2019.
- [30] E. A. Shams and A. Rizaner, "A novel support vector machine based intrusion detection system for mobile ad hoc networks," *Wirel. Networks*, vol. 24, no. 5,

pp. 1821–1829, 2018.

- [31] W. Suryan, *Software Quality Engineering: A Practitioner's Approach*, vol. 9781118592496. 2014.
- [32] Suwanto Sanjay dan FaFadhilah Syafria, "Sistem Pakar Identifikasi Nyamuk Menggunakan Pohon Keputusan (Studi kasus Nyamuk Anopheles betina asal Oriental di Indonesia)," *J. Sains*, vol. 11, no. 2, pp. 266–272, 2014.
- [33] M. Verma and B. Raman, "Center symmetric local binary co-occurrence pattern for texture, face and bio-medical image retrieval," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 224–236, 2015.