INFOMATEK: Jurnal Informatika, Manajemen dan Teknologi
Volume 24 Nomor 1 Juni 2022

PENGEMBANGAN APLIKASI IDENTIFIKASI ANOPHELES BERBASIS MOBILE

Alvianus Dengen*

Program Studi Teknik Elektro, Universitas Teknologi Sulawesi

Abstrak: Pengembangan *smartphone* sebagai *tools* yang digunakan dalam penelitian memberikan banyak manfaat dibeberapa 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, jenis 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: Aplikasi, Android, Anopheles, Training Image, Identifikasi, Algoritma Eigenface

I. PENDAHULUAN

Anopheles merupakan salah satu genus nyamuk yang bisa membawa parasit *Protozoa Plasmodium* yang disebar oleh nyamuk *Anopheles* (*An.*) betina kepada manusia (Martineau et.al, 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*.

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. 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 *Abdomen* dan sayap dari *Anopheles* yang mempunyai ciri-ciri tertentu supaya proses Ekstraksi ciri berfungsi memperjelas ciri-ciri sehingga mudah dalam melakukan identifikasi.

*) alvianusdengen@utsmakassar.ac.id

Diterima: 20 November 2021
Direvisi: 6 Juni 2022
Disetujui: 16 Juni 2022
DOI: 10.23969/infomatek.v24i1.4712

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*. 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*. 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.

II. METODOLOGI

Anopheles merupakan salah vector penyakit malaria yang mempunyai ciri khusus dari setiap jenisnya. Identifikasi morfologi dilakukan pada nyamuk betina berkaitan dengan perannya sebagai vektor malaria. Bagian yang diamati saat identifikasi morfologi nyamuk dewasa *Anopheles* meliputi ada tidaknya bercak kaki, proboscis, palpus, dan sayap. Variasi hasil identifikasi morfologi pada spesies sibling *Anopheles* dapat dipengaruhi oleh perbedaan habitat maupun perbedaan geografi (Harwin, 1969; Reid, 1968). Variasi dapat berupa perbedaan ukuran, warna, kaetotaksi, perilaku dan lainnya sehingga menyebabkan adanya perbedaan morfologi. Pada penelitian ini penulis akan menjadikan abdomen dan sayap anopheles sebagai data yang akan dipakai untuk Training image dan digunakan sebagai citra masukan untuk Identifikasi.

Citra yang juga biasa disebut gambar adalah informasi yang berbentuk visual. Proses Penangkapan citra berasal dari kekuatan sinar yang direfleksikan oleh objek. Pada saat cahaya menyinari sebuah objek, maka objek akan merefleksikan kembali cahaya tersebut (Cahyaningsih, 2010), (Manisha and Balasubramanian, 2015). Alat-alat pengindra optik bisa menangkap refleksi tersebut, contohnya mata manusia, kamera, mesin *scanner*, dan lainnya. Contohnya jika alat optik merekam pantulan cahaya tersebut dengan menggunakan mesin digital, maka citra digital yang akan dihasilkan. Pada citra digital, kesinambungan intensitas cahaya akan dikuantisasi sesuai resolusi dengan alat perekam.

Pola bisa didefinisikan sebagai suatu entitas yang dapat diidentifikasi dan diberi nama lewat ciri-cirinya (*feature*), ciri tersebut akan menjadi pembanding antara pola satu dengan pola yang lain (Choi et.al, 2011). Yang termasuk ciri yang baik suatu citra yakni citra yang mempunyai daya pembeda yang tinggi, di mana pola dikelompokkan berdasarkan cirinya bisa dilakukan dengan keakuratan yang tinggi. Ciri-ciri yang ada pada pola bisa didapatkan melalui hasil pengukuran titik suatu objek uji. Terkhusus pola dalam suatu

citra, ciri-ciri dari polanya bersumber dari informasi. Contohnya seperti kontur seperti garis, elips, dan lingkaran.

Preprocessing merupakan tahapan di mana data masukan aplikasi akan diproses untuk kebutuhan aplikasi. Gambar akan diubah dahulu dimensinya ke ukuran 100x100 *pixel*. Ukuran tersebut selanjutnya digunakan sebagai ukuran standard untuk pemrosesan data, baik data training maupun data gambar yang akan diidentifikasi.

Tahapan konversi citra asli ke citra biner disebut binerisasi citra, gambar berwarna di konversi ke biner hitam putih dilakukan dengan proses *Threshold*. *Threshold* merupakan batas yang menentukan apakah sebuah pixel akan diubah ke hitam atau putih dalam proses binerisasi. Gambar yang akan diproses dimuat terlebih dahulu di memori kemudian dibaca tinggi dan lebarnya ke dalam variabel. Nilai *threshold* yang ditentukan disimpan ke dalam variabel. Setiap nilai *pixel* diproses didalam sebuah perulangan sesuai koordinat *pixel* yang didapatkan lewat nilai tinggi dan lebar dari gambar. Nilai *pixel* tersebut yang awalnya berupa nilai bertipe integer dikonversi ke nilai RGB (*red-green-blue*). Nilai RGB bertipe *byte* yang memiliki nilai dari 0 sampai 255. Dari nilai *pixel* yang bertipe integer akan didapat nilai RGB dimana setiap nilainya tidak lebih dari 0 sampai 255. Ketiga nilai tersebut kemudian dijumlahkan dan dibagi tiga untuk mendapatkan rata-rata nilai tersebut.

Citra hasil dari proses *threshold* selanjutnya diubah ke vektor satu dimensi, hal ini bisa mempermudah menghitung jarak dari masing-masing citra yang akan dipakai dalam proses pengenalan. Contohnya dalam *training image* terdapat citra dengan ukuran 10x10*pixel* maka citra mempunyai *flatvector* ukuran 1x10. Dalam aplikasi *Anopheles* hasil dari proses binerisasi tersebut menghasilkan sebuah flatvector berupa string yang hanya berisi nilai biner 0 dan 1.

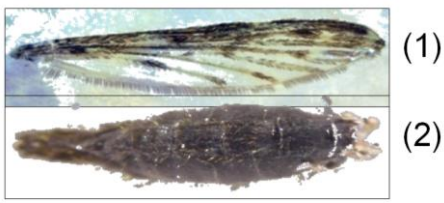
Secara keselurahan aplikasi menggunakan Algoritma *Eigenface*, baik pada saat memasukkan data ke *dataset* maupun proses Identifikasi. *Decoding* dilakukan dengan cara menghitung lalu disajikan melalui sebuah matriks yang memiliki ukuran besar, proses ini terdapat dalam metode *eigenface*. Dalam Algoritma *Eigenface* terdapat pendekatan *eigenvalue* dan *eigenvector*, melalui pendekatan ini, selanjutnya dilakukan pengkategorian guna mengenali bagian tubuh nyamuk *anopheles* yang diidentifikasi kemudian dibandingkan dengan objek yang sudah tersimpan dalam *dataset*. Prinsip dasar dalam pengenalan citra adalah melalui pengambilan informasi yang bersifat unik dari sebuah citra kemudian dipakai proses *encode* dan selanjutnya membandingkannya melalui hasil *decode* yang terlebih dahulu dilakukan.

Perangkat lunak (*software*) dan perangkat Keras (*Hardware*) adalah perangkat atau alat yang digunakan dalam penelitian kali ini adalah. Perangkat lunak meliputi *Android Studio* yang sesuai untuk perancangan tatap muka dari aplikasi dan untuk membuat algoritma Eigenface (Collins et.al, 2011), (Hu et.al, 2015). Untuk sistem operasi dari telepon pintar yang digunakan, penulisan menggunakan *Android marshmallow* sistem operasi *Android* generasi ke 13. Perangkat keras untuk *android studio* menggunakan Laptop dengan spesifikasi *processor* 2.7 GHz intel Core i5, Memory 8 GB MHz DDR3, Intel Iris Graphics 6100 1536 MB. Spesifikasi telepon pintar yang digunakan size 6.0 Inches, kapasitas 128 GB, RAM 4 GB, Kamera

belakang 48 MP, Kamera depan 16 MP. Tambahan *Wifi Microschope* dan APK untuk diinstal di telepon pintar sebagai tambahan lensa (Kaur, 2015).

Subyek dari pada penelitian ini adalah merancangan aplikasi identifikasi *anopheles* berbasis *mobile*. Data yang dipakai dalam penelitian ini menggunakan citra 22 jenis abdomen dan sayap dari *Anopheles* dengan format JPG yang semuanya didapatkan dari Laboratorium Entomologi Universitas Hasanuddin. Citra tersebut kemudian diolah menjadi dataset untuk aplikasi. Identifikasi citra *Anopheles* dilakukan menggunakan telepon pintar yang sudah diinstal aplikasi yang telah dibuat. Jika aplikasi menangkap citra, aplikasi akan membandingkan nilai *eigenface* dataset dengan citra masukan yang akan di identifikasi, maka aplikasi akan menampilkan presentasi kemiripan dan nama *Anopheles* yang mendekati dengan dataset.

Data yang digunakan berasal dari laboratorium entomologi Universitas Hasanuddin. Data yang dapat dikumpulkan berupa gambar-gambar nyamuk Anopheles dari beberapa jenis seperti *Barbirostris, Barbumbrosus, Farauti, Flavirostris, Hyrcanus, Kochi, Indefinitus, Koliensis, Longirostris, Maculatus, Pseudobarbirostris, 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. *Abdomen* dan sayap nyamuk *Anopheles* menjadi data *training* yang dianggap paling mudah dilakukan proses identifikasi. Berikut adalah beberapa contoh gambar yang dipilih:



Gambar 1. Contoh Gambar (1) Sayap An. Barbirostis (2) Abdomen An. Punculatus

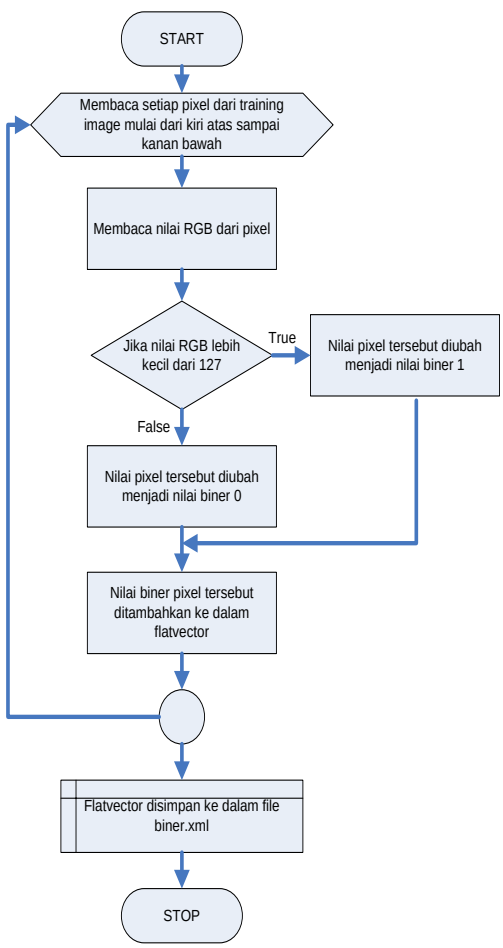
Kesulitan yang dialami selama proses pengambilan data adalah tubuh nyamuk yang terlalu kecil sehingga membutuhkan perangkat keras seperti mikroskop untuk mengambil gambar tersebut. Terdapat pula banyak bagian tubuh nyamuk yang telah hancur karena telah terlalu lama disimpan.

Sebelum diproses lebih lanjut, gambar tersebut diubah dahulu dimensinya menjadi berukuran 100x100 *pixel*. Ukuran tersebut selanjutnya digunakan sebagai ukuran standard untuk pemrosesan data, baik data *training* maupun data gambar yang akan diidentifikasi. Gambar tersebut kemudian dipilih sebagai *training image* yang akan menghasilkan gambar dalam bentuk susunan digit biner 1 dan 0. Di mana digit biner 1 mewakili area gambar yang berwarna *pixel* hitam, sedangkan digit biner 0 mewakili area gambar yang berwarna *pixel* putih.

Proses perubahan dari gambar berwarna menjadi gambar biner yang berwarna hitam putih dituliskan dalam *pseudocode* berikut:

```
pic ←
bacaFileGambar("gambarnyamuk.jpg")
for x ← 0 to lebarGambar(pic)
  for y ← 0 to tinggiGambar(pic)
    p ← bacaPixel(x,y)
    r ← red(p)
    g ← green(p)
    b ← blue(p)
```

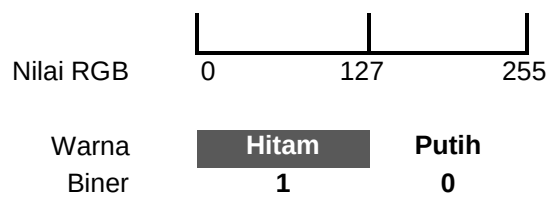
```
    if(r < 127 and g < 127 and b <
127)
      string.append("1")
    else
      string.append("0")
    end if
  next j
next i
flatVector ← string
```



Gambar 2. Pseudecode dan Flowchart dari Proses Binerisasi Citra

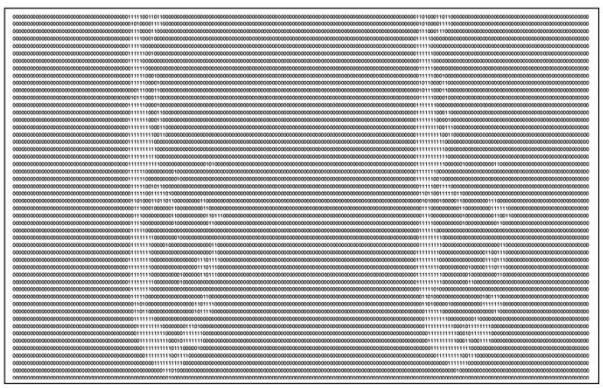
Gambar digital merupakan *array* dua dimensi atau dikenal dengan istilah matriks. Setiap elemen dari matriks merupakan *picture element* (*pixel*) yang menyusun gambar tersebut. Setiap *pixel* berisi nilai integer dari 0 sampai dengan 16.777.215. Nilai integer tersebut kemudian diubah ke dalam format warna *Red-Green-Blue* (RGB) agar dapat diketahui komposisi warna dari setiap *pixel*. Pengubahan dari nilai integer ke RGB tersebut dilakukan lewat proses yang dituliskan dalam *pseudocode* berikut:

```
R ← 0xFF and nilaiPixel
G ← (0xFF00 and nilaiPixel) DIV 256
B ← (0xFF0000 and nilaiPixel) DIV 65536
```

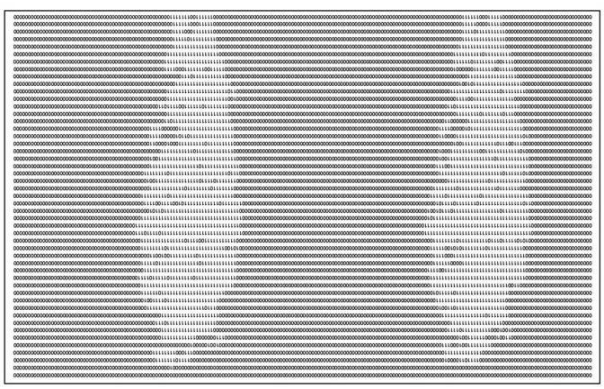


Gambar 3. Nilai Threshold untuk menentukan batas hitam dan putih

Setelah didapatkan nilai RGB dari pixel tersebut maka selanjutnya adalah menentukan nilai dari *threshold* sebagai ambang batas penentu. Nilai *threshold* diperlukan untuk memastikan apakah sebuah pixel itu nantinya dikategorikan ke dalam warna hitam atau warna putih. Angka 127 merupakan angka yang cukup baik sebagai nilai *threshold* karena merupakan nilai tengah dari angka 0 sampai 255. Setiap pixel yang memiliki nilai elemen RGB di bawah 127 akan dikategorikan berwarna hitam dan memiliki nilai biner 1, sedangkan *pixel* yang memiliki nilai elemen RGB 127 ke atas akan dikategorikan berwarna putih dan memiliki nilai biner 0.



(1)



(2)

Gambar 4. Contoh Flatvector (1) An. Barbirostris, (2) Abdomen An. Puculatus

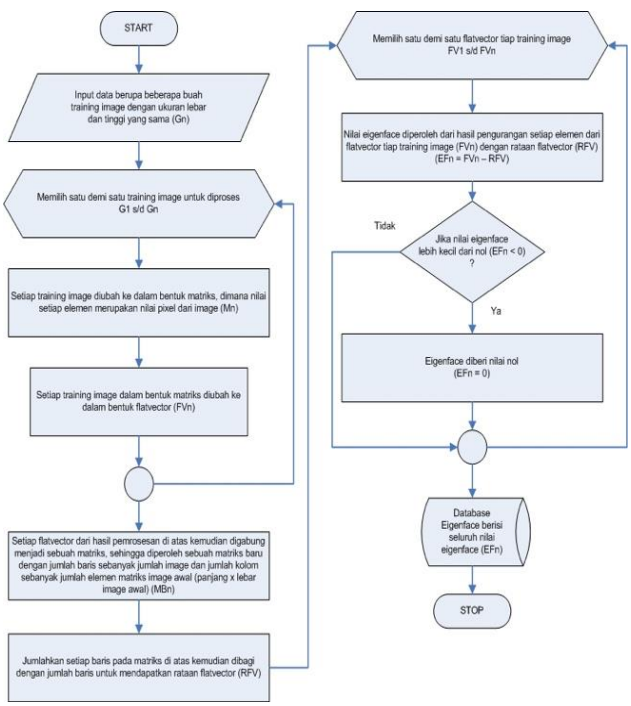
Data *flatvector* berikut nama dari nyamuk tersebut kemudian disimpan ke dalam file XML. Proses tersebut dilakukan terhadap semua data gambar yang dijadikan sebagai *training image*. Hasilnya berupa file XML yang berisi serangkaian data f*latvector* dari tiap training image nyamuk *Anopheles*. File XML tersebut nantinya digunakan di dalam proses *eigenface* untuk mengidentifikasi gambar nyamuk lain. Proses mengubah citra digital ke citra biner dan disimpan ke dalam file biner.xml dan nama jenis nyamuk disimpan ke dalam file jenis.xml dapat dilihat pada pseudecode berikut:

```
SET str
pic ←
bacaFileGambar("trainingImage.jpg")
for i ← 0 to lebarGambar(pic)
  for j ← 0 to tinggiGambar(pic)
    p ← bacaPixel(i,j)
    r ← red(p)
    g ← green(p)
    b ← blue(p)
    if(r < 127 and g < 127 and b < 127)
      str.append("1")
    else
      str.append("0")
    end if
   next j
next i
tulisFileTeks("biner.xml", str)
INPUT(namaNyamuk)
tulisFileTeks("jenis.xml", namaNyamuk)
```

a. Proses *Training Image*

Proses *Training Image* dengan Algoritma *Eigenface* dimulai dari: (1) data masukan berupa beberapa buah *training image* dengan ukuran lebar dan tinggi yang sama (Gn), (2) memilih satu demi satu *training image* untuk diproses G1 sampai dengan Gn. (3) Setiap *training image* diubah ke dalam bentuk matriks, di mana nilai setiap elemen merupakan nilai pixel dari gambar (Mn). Setiap training image dalam bentuk matriks diubah ke dalam bentuk *flatvector* (FVn). (4) Setiap *flat vector* dari hasil pemrosesan di atas kemudian digabung sehingga menghasilkan matriks, sehingga diperoleh sebuah matriks baru memiliki jumlah yang barisnya sama banyak dengan jumlah gambar dan jumlah kolom sama banyaknya dengan jumlah elemen matriks gambar awal (panjang x lebar gambar awal) (MBn). (5) Jumlahkan setiap baris pada matriks di atas kemudian dibagi dengan jumlah baris untuk mendapatkan rataan *flat*

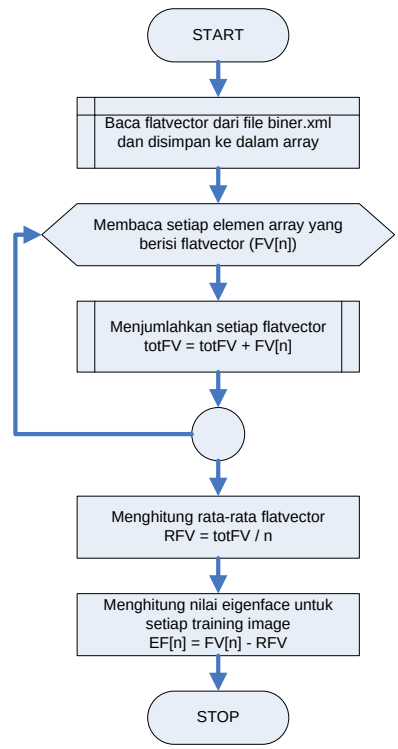
vector (RFV). (6) Memilih satu demi satu *flat vector* tiap training image FV1 s/d FVn. (7) Nilai *Eigenface* diperoleh dari hasil pengurangan setiap elemen dari *flat vector* tiap training image (FVn) dengan rataan *flat vector* (RFV) (EFn = FVn – RFV). (8) Jika nilai *Eigenface* lebih kecil dari nol (EFn < 0), maka *Eigenface* diberi nilai nol (EFn = 0). Simpan nilai *Eigenface* ke dalam database (EFn).



Gambar 5. Flowchart Training Image

Menghitung nilai *eigenfaces* secara keselurahan pada proses *training image* digambarkan dalam *flowchart* dan *pseudecode* berikut:

```
FV[] ← bacaFileTeks("biner.xml")
n ← FV[].length - 1
for i ← 0 to n
  totalFV ← addVector(FV[i],
totalFV)
next i
RFV ← avgFlatVector(totalFV, n)
for i ← 0 to n
  EF[i] ← subsVector(FV[i],
RFV[])
next
```



Gambar 6. *Flowchart* Hitung Nilai *Eigenfaces*

b. Proses Identifkasi

Proses kedua dalam aplikasi ini adalah proses Identifikasi, data masukan dalam proses Identifikasi data juga berupa bagian-bagian tubuh dari nyamuk *Anopheles*. (1) Input data berupa sebuah gambar yang akan diidentifikasi (Gx), (2) Gambar tersebut kemudian diubah ke dalam bentuk matriks, di mana nilai setiap elemen merupakan nilai pixel dari gambar (MGx). (3) Gambar tersebut diubah ke dalam diubah ke dalam bentuk *flat vector* (FVGx). (4) Nilai *Eigenface* diperoleh dari hasil pengurangan setiap elemen dari flatvector gambar tersebut (FVGx) dengan rataan *flatvector training image* yang telah didapatkan (RFV) (EFGx = FVGx – 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) (IDn = EFn – EFGx). (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*.

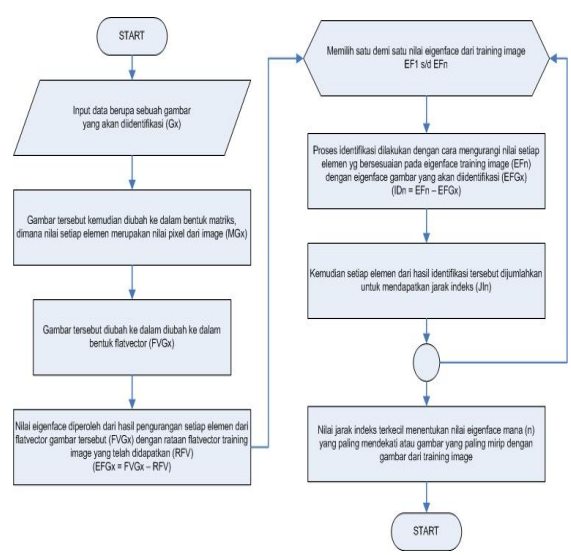
Proses Identifikasi dengan algoritma *Eigenface* dapat dilihat pada pseudecode dan flowchart berikut:

```
FV[] ← bacaFileTeks("biner.xml")
n ← FV[].length - 1
for i ← 0 to n
  totalFV ← addVector(FV[i],
totalFV)
next i
RFV ← avgFlatVector(totalFV, n)
for i ← 0 to n
  EF[i] ← subsVector(FV[i],
RFV[])
next
SET k ← 0
pic ←
bacaFileGambar("identifyImage.jpg"
)
for i ← 0 to n
  for j ← 0 to n
    p ← bacaPixel(i,j)
    r ← red(p)
    g ← green(p)
    b ← blue(p)
    if(r < 127 and g < 127 and b <
127)
      FVG[k] ← 1
    else
      FVG[k] ← 0
    end if
    k ← k + 1
  next j
next i
```

```
EFG[] ← subsVector(FVG[], RFV[])
for i ← 0 to n
  ID[i][] ← subsVector(EF[i],
EFG)
next i

for i ← 0 to n
  JI[i] ← arrSum(ID[i][])
next i

SET trainImg = -1
SET tmp = MAX_INTEGER
for i ← 0 to n
  if(JI[i] < tmp)
    tmp ← JI[i]
    trainImg ← i
  end if
next i
PRINT "Dianggap sama dengan
training image ke ", trainImg
namaNyamuk[] ←
bacaFileTeks("jenis.xml")
PRINT "Dengan jenis nyamuknya
adalah ", namaNyamuk[trainImg]
```



Gambar 7. Flowchart Proses Identifikasi

III. HASIL DAN PEMBAHASAN

Berisi data, analisis terhadap data dan pembahasannya. Hasil pengujian dari aplikasi dilakukan dengan memasukkan citra (*image*)

yang akan dijadikan *database* aplikasi. Citra ini berasal dari laboratorium Entomologi yang terdiri dari bagian abdomen dan sayap. Pada aplikasi ini, dua menu yang dirancang, yakni:

1. *Training image*, citra masukkan akan diolah sesuai prosedur yang dirancangan dalam aplikasi sehingga sistem mempunyai pengetahuan
2. Identifikasi, citra masukkan berupa abdomen dan sayap yang penulis potong sesuai aturan peneliti dari laboratorium entomologi. Citra ini kemudian akan diambil gambarnya dan dijadikan data identifikasi.

Pengujian selanjutnya dalam hal penggunaan, aplikasi *Anopheles* lebih fleksibel digunakan dibanding dengan identifikasi secara morfologi dimana aplikasi anopheles ini bisa digunakan asal data citra *anopheles* ada sehingga proses identifikasi bisa dilakukan kapanpun dan dimana saja.

3.1 Hasil Pengujian Tahap *Training Image*

Tampilan aplikasi *Anopheles* dalam proses *training image* mulai dari proses *preprocessing* sampai ke simpan data biner ke XML bisa dilihat pada gambar berikut:

Gambar 8. Proses Training Image pada Aplikasi Anopheles

3.2 Hasil Pengujian Tahap Identifikasi

Proses Identifikasi pada aplikasi *Anopheles* dimulai dari *Preprocessing*, merubah citra ke citra biner, pengaturan nilai *threshold* dalam aplikasi, kemudian menampilkan *flatvector*. Hasil Identifikasi bisa dilihat digambar dibawah ini di mana citra masukan dibandingkan dengan citra yang sudah ada dalam *database* kemudian hasil perbandingan bisa dilihat pada gambar hasil identifikasi terdapat garis-garis merah yang merupakan *vector* yang menunjukkan titik-titik tersebut merupakan kesamaan ciri antara citra masukan dan citra yang terdapat di dalam *dataset*. tahap akhir terdapat keterangan jumlah *pixel* yang sesuai dan jumlah presentasi kemiripan.

Gambar 9. Proses Identifikasi pada Aplikasi Anopheles

3.3 Hasil Penelitian

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)*. 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*.

$$FMR = \frac{Jumlah\ Kesalahan\ Cocok}{Jumlah\ Keselurahan\ Proses\ Pencocokan}\ X\ 100\% \quad (1)$$

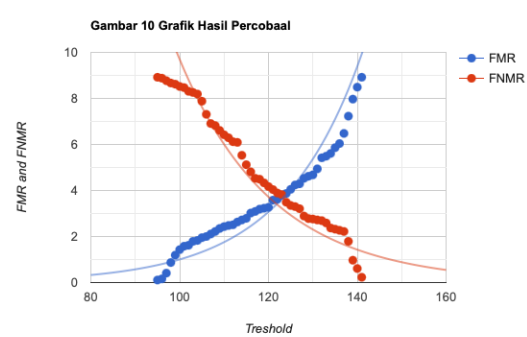
$$FMR = \frac{Jumlah\ Kesalahan\ Tidak\ Cocok}{Jumlah\ Keselurahan\ Proses\ Pencocokan}\ X\ 100\% \quad (2)$$

Pada persamaan (1) dan (2) 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*.

Tabel 1. Laporan Hasil Penelitian

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 10. Grafik Hasil Percobaan

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%.

V. 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 tingkat akurasi yang tinggi.

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 menampilakan hasil proses Identifikasi beserta gambar pembanding dari *dataset*.

DAFTAR PUSTAKA

Bonne-Webster, J. dan Swellengrebel, N.H. (1953). *The anopheline mosquitoes of the Indo- Australian Region*. J.H. de Bussy Amsterdam. 503 p.

Cahyaningsih, S. (2010). *Deteksi Osteoporosis dengan Thresholding Metode Otsu Pada Citra X-Ray Tulang Rahang*. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.

Choi, K., Toh, K. A., and Byun, H. (2011) Realtime Training on Mobile Devices for Face Recognition Application. *Pattern Recognition, 44*(2), 386-400

Collins, C., Galpin, M. D., and Kaeppler, M. (2011) *Android in Practice*, Shelter Island: Manning Publications Con..

Harwin, A.R. (1969). The concept of sibling spe-sies. *Ostrich : Journal of African Ornithology*, 40 : 27 -32

Hu, J., Peng, L., and Zheng, L. (2015). XFace: A Face Recognition System for Android Mobile Phones. *IEEE 3rd International on Cyber-Physical Systems, Networks, and Apllication*, pp. 13-18.

Kaur, A. (2015). An Essential guide to Automated GUI testing of Android Mobile Application. *International Journal of Computer Techniques, 2*(6), 8-12.

Manisha, V. and Balasubramanian, R. (2015). Center Symmetric local binary co-occurrence pattern for texture, face, and bio-medical image retrieval. *J. Vis. Comun. Image Represent, 32*, 224-236

Martineau, M., Conte, D., Raveaux, R., Arnault, I., Munier, D., Venturini, G. (2017). A survey on image-based insect classification. *Pattern Recognition, 65*, 273-284

Reid, A.J. (1968). *Anopheline mosquitoes of Ma-laya and Borneo*. Institute forMedical Research. Malaysia.