Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau

BAB II

LANDASAN TEORI

2.1 Sistem Pengenalan Wajah

Sistem pengenalan wajah dapat dibagi menjadi dua tahap, yang pertama tahap deteksi wajah (*pre-processing*) dan yang kedua adalah tahap pengenalan wajah (*face recognition*).

2.1.1 Sistem Deteksi Wajah

Pendeteksian wajah (*face detection*) adalah salah satu tahap awal (*preprocessing*) yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*). Bidang-bidang penelitian yang juga berkaitan dengan pemrosesan wajah adalah:

1. Pengenalan wajah (*face recognition*) yaitu membandingkan citra wajah masukan dengan suatu database wajah dan menemukan wajah yang paling cocok dengan citra masukan tersebut.
2. Autentikasi wajah (*face authentication*) yaitu menguji keaslian/kesamaan suatu wajah dengan data wajah yang telah diinputkan sebelumnya.
3. Lokalisasi wajah (*face localization*) yaitu pendeteksian wajah namun asumsi hanya ada satu wajah di dalam citra
4. Penjejakan wajah (*face tracking*) yaitu memperkirakan lokasi suatu wajah di dalam video secara real time.
5. Pengenalan ekspresi wajah (*facial expression recognition*) untuk mengenali kondisi emosi manusia.

2.1.2 Sistem Pengenalan Wajah

Sistem pengenalan wajah (face recognition) adalah suatu sistem yang bertujuan untuk mengidentifikasi atau memverifikasi wajah seseorang melalui citra digital maupun video digital. sistem pengenalan wajah telah menjadi salah satu aplikasi pengolahan citra yang cukup populer, khususnya di bidang sistem keamanan, verifikasi kartu kredit dan indentifikasi penjahat.

2.2 *Down syndrome*

Down syndrome adalah kelainan yang terjadi pada anak yang mengalami keterbelakangan mental yang disebabkan oleh adanya kelainan pada kromosom nomor 21 yang terdiri dari dua kromosom sebagaimana mestinya, melainkan tiga kromosom (*trisomy* 21), yang mengakibatkan anak mengalami penyimpangan fisik (ISDI, 2017).

Anak *down syndrome* biasanya lahir dengan keterbatasan kognitif, maka akan mempengaruhi akademik mereka. Anak *down syndrome*e biasanya mengalami kesulitan dengan hal-hal yang berhubungan dengan belajar karena kemampuan atensi, metacognition, memory, dan generalisasi yang lambat. Masalah ini dapat berasal dri kemampuan persepsi dan menilai yang lemah, hal ini disebabkan karena kemampuan dalam menggunakan ingatan jangka pendek yang lemah pada anak *down syndrome*. Tapi pada anak *down syndrome* memiliki kemampuan *visual processing skills* yang lebih baik sehingga diyakini gambaar merupakan metode yang bagus untuk mengajarkan anak *down syndrome* belajar, berbicara, dan berinteraksi.

Beberapa penelitian menunjukan bahwa anak dengan *intellectual disability* mengalami masalah dengan *working memory*, dan penelitian lain mengatakan anak dengan *intellectual disability* memiliki ciri ketidakmampuan yang signifikan dalam fungsi *memory* khususnya *working memory* (Rista P, Lenny K, Afra HN)

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.3 Pengolahan Citra

Pengolahan citra digital mulai dikembangkan pertama kali pada tahun 1964, pada saat itu pengolahan citra digital digunakan untuk memperbaiki kualitas citra bulan dari distorsi di laboratorium *Jet Propulsion.* Pengolahan citra digital berkembang hingga saat ini, teknologi ini memiliki spectrum aplikasi yang sangat luas, diantaranya dibidang biomedis, penginderaan jarak jauh, biometrika, fotografi, desain visual, identifikasi objek, bidang volumetric, dll.

Ada beberapa hal yang penting dalam pengolahan citra digital, antara lain teknik-teknik pengambilan citra, model citra digital, sampling dan kuatitasi, histogram, proses *filtering*.

2.4 Citra Digital

Citra digital adalah gambar dua dimensi (2D) yang bisa ditampilkan pada layar komputer sebagai himpunan nilai digital yang disebut pixel atau *picture elements*. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intentitas cahaya pada bidang dua dimensi (2D).

Citra digital merupakan fungsi intensitas cahaya f(x,y), dimana harga x dan harga y adalah koordinat spasial. Harga fungsi tersebut di setiap titik (x,y) merupakan tingkat kecemerlangan citra pada titik tersebut.

Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar / pixel / piksel / pels / *picture elements*) menyatakan tingkat keabuan pada titik tersebut.*grayscale*

2.4.1 *Grayscale image*

Grayscale image merupakan citra yang terdiri dari 256 warna keabuan atau sering sering disebut citra 8 bit. (Santi, 2011) menunjukkan salah satu cara untuk mengkonversi citra RGB menjadi *grayscale*, yaitu dengan rumus:

$$s = \frac{r+g+b}{3} \qquad [2.1]$$

Keterangan:

s = intensitas warna keabuan piksel.

r = intensitas warna piksel merah.

g = intensitas warna piksel hijau.

b = intensitas warna piksel biru.

(Tarun Kumar and Karun Verma, 2010) juga menunjukkan rumus yang lain dalam mengkonversi citra RGB menjadi *grayscale* dengan rumus:

$$Iy = 0.333Fr + 0.5\,Fg + 0.1666Fb \qquad [2.2]$$

Keterangan:

Iy = intensitas warna keabuan piksel

Fr = intensitas warna piksel merah

Fg = intensitas warna piksel hijau

Fb = intensitas warna piksel biru

Setelah proses perubahan citra ke dalam citra *grayscale*, maka selanjutnya akan dilakukan pengubahan citra *grayscale* menjadi *binary image.*

2.4.2 *Binary image*

binary image didapat dengan mengatur derajat keabuan yang ada pada sebuah citra (Santi, 2011). *Binary image* adalah citra yang terdiri dari hitam(0) dam putih(255). Salah satu cara untuk mengubah citra *grayscale* menjadi *binary image* adalah dengan *thresholding*.

2.4.3 *Thresholding*

thresholding adalah teknik pendekatan daerah yang bertujuan untuk melihat objek padat pada sebuah latar belakang (Castleman, 1996). *Thresholding* adalah metode paling sederhana dari segmentasi citra. Dari abu-abu, *thresholding* dapat digunakan untuk membuat citra biner (Shapiro, 2002). Pada proses *thresholding*, masing-masing piksel pada sebuah citra ditandai sebagai piksel milik objek jika

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak Cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

nilainya lebih besar dari nilai *threshold*. Dengan asumsi objek tersebut lebih terang dari latar belakangnya, hal ini disebut *thresholding above*. Sedangkan kebalikannya disebut *threshold below*. Jika piksel-piksel suatu objek berada diantara dua *threshold* disebut *threshold inside* dan kebalikannya disebut *threshold outside*. Biasanya piksel suatu objek dilabeli dengan nilai 1, sementara piksel-piksel latar belakangnya dilabeli dengan nilai 0. *Thresholding* dapat dibentuk dalam persamaan sebagai berikut:

$$g(x,y) = \begin{cases} 1, if & f\ (x,y) \geq T \\ 0, if & f\ (x,y) \leq T \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

Jika nilai T konstan, pendekatan ini disebut *global thresholding*.

2.4.4 *Connected Component Labeling*

Dalam permasalahan *connected components labeling* akan menghasilkan sebuah citra dengan label baru yang sudah ditetapkan yang sudah terkait satu sama lainnya antar sesama komponen (Luigi and Andrea, 2011).

Penandaan komponen terhubung dilakukan dengan memeriksa suatu citra, piksel per piksel dari kiri ke kanan dan atas ke bawah untuk mengidentifikasi area piksel terhubung, yaitu suatu area dari piksel berbatasan yang memiliki nilai intensitas yang sama. Pelabelan komponen terhubung dapat dilakukan pada citra biner maupun citra *grayscale*. Sebagai contoh terdapat pola deretan angka biner sebagai berikut :

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 2.1 Matriks gambar

Proses periksa citra dilakukan dengan bergerak sepanjang baris sampai menemukan piksel p yang merupakan *foreground*. *Foreground* direpresentasikan dengan piksel berwarna hitam atau bernilai 0, sedangkan *background* direpresentasikan dengan piksel berwarna putih atau bernilai 1 Bila sudah ditemukan piksel p maka periksa nilai piksel tetangga dari p, yaitu piksel di atas dan di kiri dari p, kemudian lakukan pemeriksaan berikut :

1. Bila kedua piksel tetangga bernilai 1 maka berilah label baru pada p.
2. Jika hanya satu saja dari piksel tetangga tersebut bernilai 0 maka berilah label dari piksel tetangga tersebut pada p.
3. Bila kedua piksel tetangga bernilai 0 dan memiliki label sama maka berilah label dari piksel tetangga tersebut pada p.
4. Bila kedua piksel tetangga bernilai 0 dan memiliki label berbeda maka berilah label terkecil dari kedua piksel tetangga tersebut pada p dan buat catatan bahwa kedua label yang berbeda tersebut adalah ekuivalen.

Setelah melalui proses di atas, maka akan didapatkan label yang baru sebagai berikut:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	2	2	2	2	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	3	2	2	2	2	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	4	3	0	0	2	2	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	5	4	3	3	3	2	2	0
0	0	0	0	0	6	1	0	0	0	0	0	0	0	0	5	4	3	3	3	2	2	0
0	0	0	0	7	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0
0	0	0	8	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0
0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 2.2 Matriks gambar yang telah diberi label

Pada akhir proses, semua piksel bernilai yang merupakan *foreground* telah mendapat tanda tetapi beberapa label – label tersebut mungkin masih ada yang ekuivalen. Oleh karena itu proses berikutnya yang harus dilakukan adalah

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau

mengurutkan pasangan – pasangan tanda yang ekuivalen ke dalam kelas – kelas ekuivalen kemudian memberi tanda yang berbeda pada setiap kelas ekuivalen. Kemudian dilakukan proses periksa kedua untuk mengganti label dengan tanda pada kelas ekuivalen.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	2	2	2	2	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	2	2	2	2	2	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	2	2	0	0	2	2	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 2.3 Matriks gambar proses labeling kedua

Setelah citra yang baru mendapat label, maka akan dipanggil tiap label, dimana label tersebut berisi karakter-karakter, lalu label tersebut akan dilakukan similaritas menggunakan *Eigenface* terhadap data training.

2.5 Principal Component Analysis (PCA)

PCA atau dikenal juga dengan nama *Karhunen-Loeve Transformation* (KLT) merupakan salah metode ekstraksi fitur secbara statistik dan telah dikenal sejak 30 tahun dalam dunia pengenalan pola (Turk M. and Pentland A, 1991). PCA merupakan sebuah cara untuk mengidentifikasi pola dalam kumpulan data untuk mencari persamaan dan perbedaannya. PCA sangat cocok untuk menemukan pola dalam data berdimensi tinggi .

Dalam konteks pengenalan wajah, PCA mencari *eigenface* yang merupakan kumpulan dari *eigenvector*. *Eigenface* adalah *principal component* (ciri-ciri penting) dari distribusi citra wajah yang didapatkan dari *eigenvector* dengan *eigenvalue* terbesar. Untuk mendapatkan *eigenface*, PCA melakukan perhitungan matrik kovarian dari kumpulan citra wajah latih setelah dilakukan normalisasi.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Eigenface tersebut akan menjadi dasar perhitungan *face space* yang merepresentasikan nilai bobot individu yang mewakili satu atau lebih citra wajah.

2.6 *Eigenface*

Eigenface adalah salah satu algoritma pengenalan pola wajah yang berdasar pada *Principle Component Analysyis* (PCA). Sekumpulan citra digital dari wajah di normalisasikan dan di proses pada resolusi yang sama (missal mxn), kemudian citra tadi diperlakukan sebagai vektor dimensi mxn dimana komponennya diambil dari nilai piksel citra (Marti, 2010).

Principal Component Analysis (PCA), biasa disebut juga dengan *Transformasi Hotelling* atau *Transformasi Karhunen Loeve* adalah teknik untuk mengurangi dimensi sebuah ruang yang direpresentasikan oleh variable statistik (*x1, x2,…,xk)*, dimana variabel tersebut biasanya saling berkolerasi (Kholistianingsih, 2012).

Prinsip (PCA) adalah memproyeksikan citra ke dalam ruang eigennya dengan cara mencari *eigenvector* yang dimiliki setiap citra dan memproyeksikan ke dalam ruang *eigen* yang di dapat dan besarnya ruang *eigen* tergantung dari jumlah citra referensi yang dimiliki.

Analisis algoritma *eigenface* pada pengenalan wajah merupakan dasar bagi seseorang yang ingin mendalami ilmu biometrik karena identifikasi dilakukan dengan *pattern matching* sederhana tanpa menggunakan metode pembelajaran khusus. Metode *eigenface* ini banyak digunakan karena memiliki tingkat akurasi yang cukup baik dan sistem komputasi yang tidak terlalu rumit. Contoh algoritma *Eigenface* sebagai berikut:

1. Algoritma pengenalan wajah dimulai dengan membuat matriks kolom dari wajah yang diinput ke dalam database. Rata-rata *vector* citra (*mean*) dari matriks kolom dihitung dengan cara membaginya dengan jumlah banyaknya citra yang disimpan di dalam database.

2. Langkah pertama adalah menyusun seluruh *training image* menjadi 1 matrix tunggal. Misalnya *image* yang kita simpan berukuran H x W piksel dan

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

jumlahnya N buah, maka memiliki *flatvector* dengan dimensi N x (H x W). representasikan semua matriks *training* menjadi matriks dengan bentuk N×1 atau matriks linier seperti yang ditunjukkan berikut ini:

$$\begin{bmatrix} a & b & c \\ x & y & z \end{bmatrix} \text{-> } [a \quad b \quad c \quad x \quad y \quad z] \qquad [2.4]$$

Misalnya didalam *training image* terdapat tiga image ukuran 3x3 piksel maka kita akan mempunyai *eigenvector* ukuran 2x9. Contoh di bawah ini mengunakan empat wajah citra yang telah diubah menjadi matrix, lalu matrix tersebut diubah kedalam bentuk rataan *FlatVector*.

$$C_1 \text{ -> } \begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix} \text{-> } [10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10]$$

$$C_2 \text{ -> } \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \text{-> } [2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2]$$

$$C_3 \text{ -> } \begin{bmatrix} 11 & 11 & 11 \\ 11 & 11 & 11 \\ 11 & 11 & 11 \end{bmatrix} \text{-> } [11 \quad 11 \quad 11 \quad 11 \quad 11 \quad 11 \quad 11 \quad 11 \quad 11]$$

$$C_4 \text{ -> } \begin{bmatrix} 9 & 9 & 9 \\ 9 & 9 & 9 \\ 9 & 9 & 9 \end{bmatrix} \text{-> } [9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9]$$

3. Hitung Rataan FlatVector dari FlatVector yang diperoleh. Jumlahkan seluruh barisnya sehingga kita peroleh matrix berukuran 1 x (H x W).

$$C_1+C_2+C_3+C_4 = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \end{bmatrix}$$

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau

$$C_1+C_2+C_3+C_4 = [32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32]$$

Setelah itu bagi matriks tadi dibagi dengan jumlah image N untuk mendapatkan Rataan *FlatVector*.

$$\frac{[32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32 \quad 32]}{4} = [8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8]$$

Nilai *flatvector* citra akan digunakan untuk menghitung nilai *eigenface* citra wajah untuk *training image*.

4. Dengan memakai rataan *flatvector* citra di atas nilai *eigenface* untuk matriks *flatvector* yang sudah disusun tersebut dapat dihitung nilai *eigenface*nya. Caranya dengan mengurangi baris-baris pada matrix *flatvector* dengan rataan *flatvector*. Jika didapatkan nilai dibawah nol, maka nilainya diganti dengan nol.

$$C_1 = \begin{array}{ccccccccc} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ \hline 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{array} -$$

$$C_2 = \begin{array}{ccccccccc} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} -$$

$$C_3 = \begin{array}{ccccccccc} 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ \hline 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{array} -$$

$$C_4 = \begin{array}{ccccccccc} 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} -$$

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

5. Untuk mengenali citra tes (*testface*), langkah identifikasinya adalah hitung nilai *eigenface* untuk matriks *testface* dengan cara sebelumnya untuk penentuan nilai *eigenface* dan *flatvector* citranya.

$$C_t \text{ -> } \begin{bmatrix} 9 & 9 & 9 \\ 9 & 7 & 9 \\ 9 & 7 & 9 \end{bmatrix} \text{ -> } (9\ 9\ 9\ 9\ 7\ 9\ 9\ 7\ 9)$$

$$\text{->} \begin{array}{ccccccccc} 9 & 9 & 9 & 9 & 7 & 9 & 9 & 7 & 9 \\ \underline{8} & \underline{8} & \underline{8} & \underline{8} & \underline{8} & \underline{8} & \underline{8} & \underline{8} & \underline{8} \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{array} -$$

Nilai *Eigenface* dari *testface* = (1 1 1 1 0 1 1 0 1)

Nilai *eigen* (*eigenvalue*) dari *testface* digunakan untuk identifikasi dengan menentukan jarak terpendek dengan *eigenface* dari *eigenvector training* dengan cara menentukan nilai *absolut* dari pengurangan baris I pada matriks *eigenface training* citra dengan *eigenface* dari *testface* dan jumlahkan dengan elemen penyusun *vector* yang dihasilkan dari pengurangan dan didapat jarak d indeks I dan cari nilai d yang paling kecil.

Setelah nilai *eigenface* untuk *testface* diperoleh maka kita bisa melakukan identifikasi dengan menentukan jarak terpendek dengan *eigenface* dari *eigenvector training image*. Pertama tentukan nilai *absolut* dari pengurangan baris pada matrix *eigenface training image* dengan *eigenface* dari *testface*, kemudian jumlahkan elemen-elemen penyusun *vector* yang dihasilkan dari pengurangan tadi dan ditemukan jarak indeks. Lakukan untuk semua baris. Cari nilai yang paling kecil.

Nilai *Eigenface* C_1 = (2 2 2 2 2 2 2 2 2)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$C_1 = \begin{array}{ccccccccc} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 2 & 1 & 1 & 2 & 1 \end{array} -$$

C_1 = 1+1+1+1+2+1+1+2+1 = 11

Nilai *Eigenface* C_2 = (0 0 0 0 0 0 0 0 0)

$$C_2 = \begin{array}{ccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline -1 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & -1 \end{array} -$$

C_2 = 1+1+1+1+0+1+1+0+1 = 7

Nilai *Eigenface* C_3 = (3 3 3 3 3 3 3 3 3)

$$C_3 = \begin{array}{ccccccccc} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline 2 & 2 & 2 & 2 & 3 & 2 & 2 & 3 & 2 \end{array} -$$

C_3 = 2+2+2+2+3+2+2+3+2 = 20

Nilai *Eigenface* C_4 = (1 1 1 1 1 1 1 1 1)

$$C_4 = \begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} -$$

C_4 = 0+0+0+0+1+0+0+1+0 = 2

Dari hasil perhitungan, diperoleh jarak citra wajah empat memiliki nilai yang terkecil yaitu dua. Karena jarak *eigenface face* empat dengan *testface* yang paling kecil, maka hasil identifikasi menyimpulkan bahwa *testface* lebih mirip dengan *face* empat dari pada *face* satu, *face* dua, dan *face* tiga.

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

2.7 *Euclidean Distance*

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. Euclidean ini berkaitan dengan Teorema Phytagoras dan biasanya diterapkan pada 1,2, 3 dimensi dan juga dapat diterapakan pada dimensi yang lebih

$$d(\boldsymbol{p},\boldsymbol{q}) - d(\boldsymbol{q},\boldsymbol{p}) - \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

tinggi. *Euclidean space* merupakan ruang dengan dimensi terbatas yang bernilai riil, jarak Euclidean antara dua titik adalah panjang sisi miring dari sebuah segitiga siku-siku. Jika **p** = (p_1, p_2,..., p_n) dan **q** = (q_1, q_2,..., q_n) adalah dua poin di *euclidean space*, maka jarak (d) dari **p** ke **q** atau dari **q** ke **p** sebagai berikut,

$$-\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Nilai vektor pada jarak Euclidean dikatakan mirip jika jarak A mendekati 0, dan jarak B > 0 bergantung batas *threshold*nya .

2.8 Tingkat Akurasi Pengujian

Pada pengenalan wajah, kesalahan saat pencocokan bisa saja terjadi. Hal ini disebabkan ada beberapa faktor yang membuat dua matriks dari dua orang yang sama tidak selalu mutlak sama terutama pada kasus pengenalan wajah dengan usia yang berbeda. *False Match Rate* (FMR) menyatakan probabilitas sampel dari pengguna cocok dengan acuan yang diambil secara acak milik pengguna yang berbeda. Sedangkan *False Non Match Rate* (FNMR) menyatakan probabilitas sampel dari pengguna tidak cocok dengan acuan lain yang diberikan pengguna yang sama. Rasio kesalahan kecocokan dan ketidakcocokan dihitung dengan rumus berikut :

$$Presentasi\ kesalahan\ kecocokan = \frac{\Sigma\ Data\ yang\ cocock}{\Sigma\ jumlah\ data\ input}\ x\ 100 \qquad (2.5)$$

$$Presentasi\ kesalahan\ ketidakcocokan = \frac{\Sigma\ Data\ yang\ tidak\ cocock}{\Sigma\ jumlah\ data\ input} x\ 100 \quad (2.6)$$

Kecocokan data didapatkan berdasarkan perhitungan ekstraksi ciri citra menggunakan *eigenface* dan dikenali oleh *Canberra Distance*. Dengan menghitung kesalahan kecocokan, maka akan didapatkan tingkat akurasi dari metode ini.

2.9 Bahasa Pemrograman Java

Java adalah bahasa pemrograman dan platform komputasi pertama kali dirilis oleh Sun Microsystems pada tahun 1995 (www.java.com/en/about/) . Java berjalan pada berbagai sistem seperti windows, Mac OS, berbagai macam sistem dari UNIX.

Java digunakan bermacam jenis aplikasi, seperti aplikasi keuangan, desktop, simulasi, pemrosesan citra, game dan masih banyak lagi.

Karakteristik java antara lain:

1. Sederhana, sintaksnya mirip C++, tapi beberapa sintaks telah di perbaharui termasuk menghilangkan pointer yang rumit.
2. Berorientasi objek, Program java dapat dibuat secara modular dan dapat dipergunakan kembali.
3. Dapat di distribusikan dengan mudah
4. Interpreter, Program Java dijalankan menggunakan Interpreter yaitu JVM yang menyebabkan kompilasi source code Java dapat dijalankan pada *platform* (platform adalah sebuah sistem komputer yang mendasari program aplikasi dapat berjalan di sebuah komputer) yang berbeda.
5. Robust, Reliabilitas Java tinggi, compilernya lebih teliti mendeteksi error daripada bahasa pemrograman lain, dan Java memiliki runtime Exception handling untuk membantu mengatasi error pemrograman.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

6. Aman (*Secure*), aplikasi java memiliki mekanisme keamanan untuk menjaga aplikasi tidak untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

7. *Architecture Neutral*, program Java merupakan patform independent. Program cukup membunyai satu versi yang dapat dijalankan pada platform yang berbeda dengan JVM.

8. *Portable*, Tanpa dikompulasi ulang program java dapat digunakan ke platform yang berbeda-beda.

9. *Performance*, Performancenya sering dikatakan kurang tinggi namun dapat ditingkatkan menggunakan kompilasi java lain seperti buatan Inprise, Microsoft, Symantec yang menggunakan Just In Time Compilers (JIT).

10. *Multithreaded*, Java memiliki kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. Dinamis, Perubahan dalam suatu Class dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program class tersebut.

2.10 Android

Android adalah sebuah sistem operasi *mobile* yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi dan digunakan oleh bermacam perangkat *mobile*. Berikut adalah beberapa versi android yang berkembang sampai saat ini:

1. Versi Beta : Dirilis 5 November 2007
2. Android 1.0 (API level 1) : Dirilis 23 September 2008
3. Android 1.1 (API level 2) : Dirilis 9 Februari 2009
4. Android 1.5 Cupcake (API level 3) : Dirilis 30 April 2009
5. Android 1.6 Donut (API level 4) : Dirilis 15 September 2009
6. Android 2.0 Eclair (API level 5) : Dirilis 26 Oktober 2009
7. Android 2.0.1 Eclair (API level 6) Dirilis 3 Desember 2009.

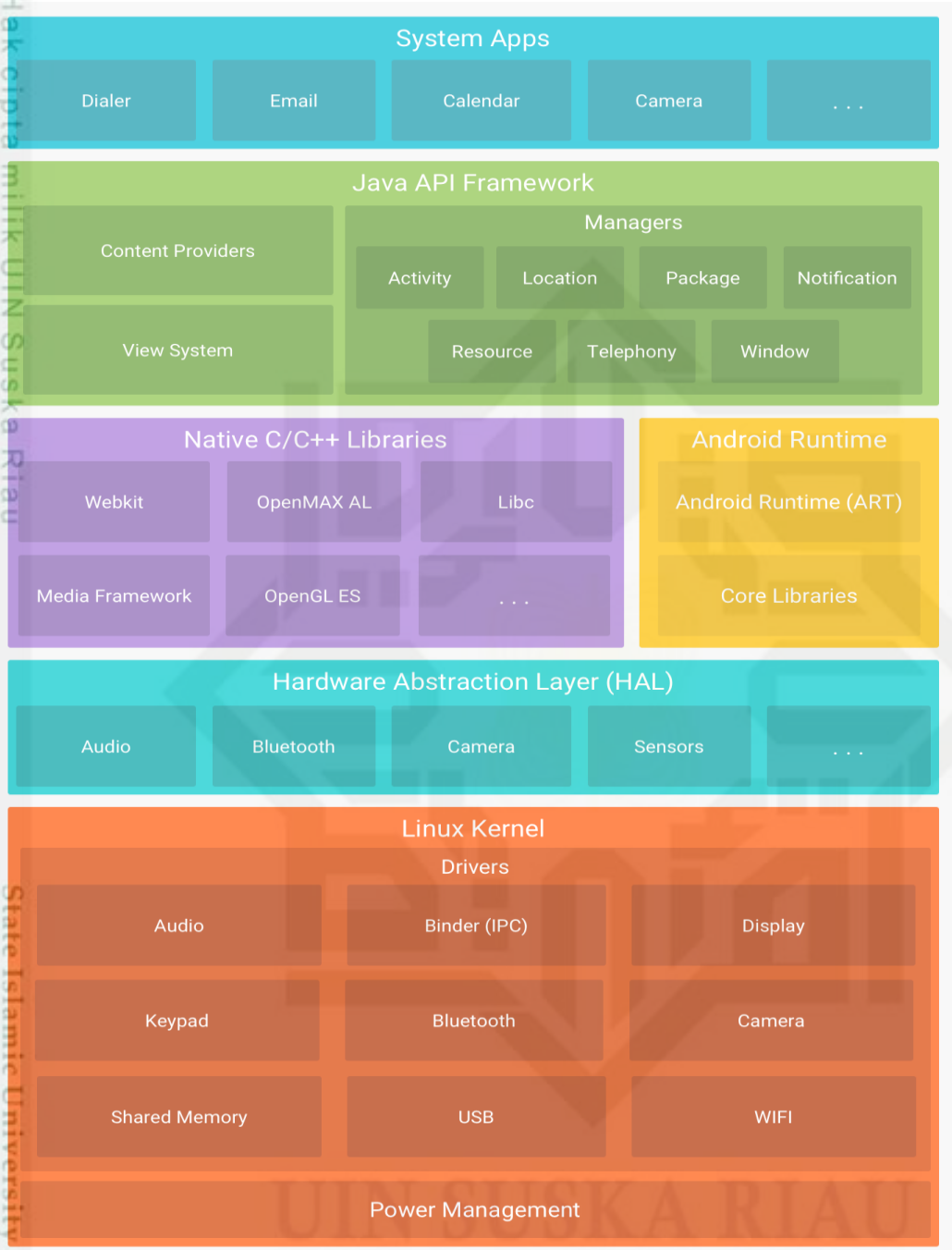
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

8. Android 2.1 Eclair (API level 7) Dirilis 12 Januari 2010
9. Android 2.2–2.2.3 Froyo (API level 8) Dirilis 20 Mei 2010, 18 Januari 2011, 22 Januari 2011, dan 21 November 2011
10. Android 2.3–2.3.2 Gingerbread (API level 9) Dirilis 6 Desember 2010, Desember 2010, dan Januari 2011
11. Android 2.3.3–2.3.7 Gingerbread (API level 10) Dirilis 9 Februari 2011, 28 April 2011, 25 Juli 2011, 2 September 2011, dan 21 September 2011
12. Android 3.0 Honeycomb (API level 11) Dirilis 22 Februari 2011
13. Android 3.1 Honeycomb (API level 12) Dirilis 10 Mei 2011
14. Android 3.2-3.2.6 Honeycomb (API level 13) Dirilis 15 Juli 2011, 20 September 2011, 30 Agustus 2011, December 2011, Januari 2012, Februari 2012
15. Android 4.0–4.0.2 Ice Cream Sandwich (API level 14) Dirilis 19 Oktober 2011, 21 Oktober 2011, 28 November 2011
16. Android 4.0.3–4.0.4 Ice Cream Sandwich (API level 15) Dirilis 16 Desember 2011, 29 Maret 2012
17. Android 4.1-4.1.2 Jelly Bean (API level 16) Dirilis 9 Juli 2012, 23 Juli 2012, 9 Oktober 2012
18. Android 4.2-4.2.2 Jelly Bean (API level 17) Dirilis 13 November 2012, 27 November 2012, 11 Februari 2013
19. Android 4.3 Jelly Bean (API level 18) Dirilis 24 Juli 2013
20. Android 4.4 KitKat (API level 19) Dirilis 31 Oktober 2013
21. Android 5.0 Lollipop (API level 21) Dirilis 03 November 2014
22. Android 5.1 Lollipop (API level 22) Maret 2015
23. Android 6.0 Marshmallow (API level 23)
24. Android 7.0 Nougat (API level 24)
25. Android 7.1.1 Nougat (API level 25)

2.10.1 Arsitektur Android

Android adalah perangkat berbasis Linux. Diagram berikut menunjukan komponen besar dari platform android (android, 2017).

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU



Gambar 2.1 Arsistektur Android

1. Linux Kernel

Fondasi platform android adalah Kernel linux. Sebagai contoh, *Android Runtime* (ART) bergantung pada kernel linux untuk fungsionalitas dasar seperti *threading* dan manajemen memori tingkat rendah

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Menggunakan kernel linux memungkinkan android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk membangun driver perangkat keras.

2. *Hardware Abstraction Layer* (HAL)

Hardware Abstraction Layer (HAL) menyediakan antar muka standar yang mengekspos kemampuan perangkat keras di perangkat kerangka kerja java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponenperangkat keras tertentu, seperti modul kamera atau Bluetooth.

3. Android Runtime

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap *Android Runtime* (ART). ART menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX.

Berikut beberapa fitur utama ART :

a. kompilasi mendahului waktu (AOT) dan tepat waktu (JIT)
b. pengumpulan sampah (GC) yang dioptimalkan
c. dukungan debug yang lebih baik, mencakup profiler sampling terpisah, pengecualian diagnostic mendetail dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke android versi 5.0 (API level 21), Dalvik adalah waktu proses android. Jika aplikasi berjalan baik pada ART, semestinya berfungsi baik juga pada dalvik.

4. Pustaka C/C++

Banyak komponen dan layanan sistem android inti seperti ART dan HAL dibuat dari kode asli yang asli memerlukan pustaka asli yang tertulis dalam C dan C++. Platform android memungkinkan kerangka kerja java API mengekspos

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak ipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

fungsionalitas beberapa pustaka asli pada aplikasi. Seperti mengakses OpengGL ES melalui kerangka kerja java openGL API android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi.

5. Kerangka Kerja Java API

Keseluruhan rangkaian fitur pada android OS tersedia melalui API yang ditulis dalam bahasa java. API ini membentuk elemen dasar yang diperlukan untuk membuat aplikasi android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular.

a. Tampilan Sistem

b. Pengelola Sumber Daya

c. Pengelola Notifikasi

d. Pengelola Aktivitas

e. Penyedia Materi

6. Aplikasi Sistem

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri.

2.11 Penelitian Terkait

Adapun beberapa penelitian sebelumnya yang terkait mengenai pendeteksian dan pengenalan wajah dengan metode *eigenface* adalah sebagai berikut.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

1. Rancang Bangun Sistem Pegenalan Wajah Dengan Multi Ekspresi Menggunakan Metode *Eigenface*

Pada penelitian yang di tulis oleh Mella Rahmadani (rahmadani, 2013) yang berjudul "**Rancang Bangun Sistem Pengenalan Wajah Dengan Multi Ekspresi Menggunakan Metode *Eigenface***" menjelaskan mengenai pengenalan wajah dengan berbagai ekspresi.

Yang diteliti pada penelitian ini adalah, apakah citra wajah dengan multi ekspresi dapat dikenali dan memiliki tingkat akurasi yang tinggi atau tidak, aplikasi yang dibuat menggunakan MATLAB dengan algoritma *eigenface*. Dengan membatasi beberapa permasalahan seperti citra wajah yang digunakan adalah citra wajah senyum, tertawa, marah, menutup mata dan tanpa ekspresi, serta citra wajah hadap depan dan hadap serong.

Hasil yang di dapat dari penelitian ini adalah, Sistem pengenalan wajah yang dibuat dengan metode *eigenface* dapat mengenali wajah hadap depan maupun hadap serong, untuk sampel citra wajah kembar sistem ini dapat membedakan citra wajah kembar tersebut, Perbedaan *cropping* pada citra wajah sangat mempengaruhi hasil pengenalan wajah yang diujikan, disebabkan citra wajah yang telah di *cropping* akan menghasilkan luas citra wajah yang berbeda pula, dan sistem ini dapat mengenali citra wajah multi ekspresi hadap depan dengan tingkat akurasi 72 %, hadap serong dengan tingkat akurasi 68 %.

2. Analisis Algoritma *Eigenface* (Pengenalan Wajah) Pada Aplikasi Kehadiran Pengajaran Dosen

Pada penelitian "**Analisis Algoritma *Eigenface* (Pengenalan Wajah) Pada Aplikasi Kehadiran Pengajaran Dosen**" yang ditulis oleh Fika Tiara Puri (Puri, 2011) menganalisa mengenai algoritma *eigenface* dan merancang aplikasi kehadiran dosen dalam pengajaran, dengan menggunakan bahasa pemrograman visual basic 6.0.

Adapun kesimpulan dari penelitian ini adalah, Algoritma *eigenface* adalah dasar bagi seseorang yang ingin mempelajari mengenai pengenalan wajah, karena identifikasi dilakukan dengan *pattern matching* sederhana.

Kekurangan yang terdapat dalam penelitian ini adalah, jarak kamera dan wajah yang di*capture* dibuat *constant* lebih kurang 30 cm, lebih dari itu maka wajah tidak akan dikenali dan Ketergantungan database harus sama dengan nama file suatu image, jika tidak sama maka tidak mampu untuk melakukan validasi.

3. Penerapan Face Recognition Dengan Metode *Eigenface* Dalam Intelligent Home Security

Pada penelitian yang ditulis oleh Setya bayu (Bayu, 2009) dengan judul "**Penerapan Face Recognition Dengan Metode *Eigenface* Dalam Intelligent Home Security"** menjelaskan bahwa metode *eigenface* ini mempunyai komputasi yang sederhana dan cepat dibandingkan dengan penggunaan metode yang memerlukan banyak pembelajaran seperti jaringan syaraf tiruan.

Proses dari aplikasi ini melalui tahap pengcapturan wajah, kemudian didapatkan sebuah nilai RGB. Kemudian dilakukan resize, RGB ke gray, dan histogram equalisasi untuk perataan cahaya, kemudian dilakukan proses PCA untuk mengutip bagian terpenting dengan metode *eigenface* sehingga di dapatkan *eigenvector* dan *eigenvalue* dari gambar tersebut. proses penyimpanan kedalam data wajah untuk setiap citra wajah yang telah digunakan dalam proses PCA dalam bentuk *.xml, semakin kompleks dan sering maka proses pengenalan wajah akan semakin baik. Data yang telah disimpan nantinya digunakan sebagai nilai pembanding pada proses penghitungan jarak untuk pengenalan wajah. Terakhir Proses pengenalan wajah dengan menghitung jarak antara fitur wajah yang ada dalam data dan fitur wajah yang baru. Jarak yang didapat di cari yang terkecil untuk identifikasi

Pada penelitian ini didapatkan akurasi pengenalan yang sangat baik, pemilik 1 dikenali dengan dengan akurasi 86,5%, pemilik 2 dikenali dengan akurasi 87,5%,

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

wajah random dengan akurasi 12,66% dengan artian wajah random merupakan bukan pemilik rumah.

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.