khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Kinerja Algoritma Pengenalan Wajah untuk Sistem Penguncian Pintu Otomatis Menggunakan Raspberry-Pi

Raden Budiarto*
Sistem Informasi
STMIK Jakarta STI&K, Jakarta
*raden@jak-stik.ac.id

Abstrak-Pengenalan wajah mampu memberikan pengalaman interaksi yang paling alami, sebagaimana manusia mampu mengenali manusia lain melalui wajah. Implementasi pengenalan wajah memerlukan biaya yang rendah karena tidak memerlukan alat pengenal khusus selain kamera yang pada saat ini sudah tertanam di berbagai perangkat seperti laptop, smartphone, atau tablet. Pengenalan wajah bukan merupakan tugas mudah bagi komputer. Masalah bertambah ketika komputer diharuskan untuk melakukan klasifikasi wajah dengan berbagai situasi dan kondisi seperti pencahayaan yang gelap, dan tangkapan gambar latar belakang yang ada. Artikel ini mendeskripsikan hasil penelitian sistem pengenalan wajah menggunakan Raspberry Pi yang diterapkan untuk sebuah prototipe pengunci pintu. Metode yang digunakan adalah mengambil sampel dataset kemudian mengevaluasi dan membandingkan algoritme pembelajaran untuk dianalisis tingkat keakuratan dan kecepatan dalam mengenali wajah. Pengujian dilakukan untuk menganalisis metode training dataset yang paling baik untuk diimplementasikan berdasarkan kriteria sensitivitas, spesifisitas, dan *false rate*. Terdapat 4 buah algoritme yang diuji yakni *Eigenfaces/PCA* dan *K-Nearest Neighbor (K-NN), PCA-LDA* dan *K-NN, Eigenfaces/PCA* dan *Support Vector Machine (SVM), PCA-LDA,* dan *SVM*. Hasil penelitian menunjukkan algoritme *hybrid Eigen-Fisherfaces* (PCA-LDA) dan k-nearest neighbor adalah yang metode yang paling akurat untuk pengenalan wajah. Akurasi mendekati 100% dapat diperoleh dengan perhitungan machine learning dengan 4 fold.

Kata Kunci: Pengenalan wajah, *machine learning, raspberry, eigen fisherfaces, k-nearest neighbor*

1. Pendahuluan

Selama beberapa tahun terakhir, ada banyak sekali pilihan pada teknologi konvensional dan teknologi biometri untuk memenuhi kebutuhan keamanan untuk rumah tangga atau kantor. Beberapa sistem keamanan konvensional, contohnya menggunakan kunci, *passcode*, kartu ID, dan/atau kartu RFID, bisa jadi tidak dapat diandalkan apabila benda untuk akses tersebut dicuri atau hilang. Sistem keamanan seperti itu memiliki kekurangan ketika akses tersebut dicuri oleh orang yang tidak memiliki wewenang untuk mendapatkan akses [1]. Oleh karena itu, sistem keamanan biometri dianggap sebagai salah satu metode autentikasi yang paling aman, sehingga dapat memberikan tingkat keamanan yang lebih baik dibandingkan dengan sistem keamanan konvensional [2].

Fingerprint recognition atau pengenalan sidik jari menjadi metode paling populer dalam teknologi biometri. Metode populer lainnya adalah pengenalan suara, pengenalan tanda tangan, pemindaian iris, dan pemindaian retina [3]. Metode-metode tersebut memberikan tingkat nilai keamanan yang sama karena menggunakan identitas unik pada manusia. Akan tetapi, dalam implementasinya metode-metode tersebut tidak dapat mengalahkan metode pengenalan wajah dalam aspek kenyamanan. *Face recognition* atau pengenalan wajah mampu memberikan pengalaman interaksi yang paling alami, hal yang serupa ketika manusia mampu mengenali manusia lain melalui wajah. Hal positif lain dalam implementasi pengenalan wajah juga akan mengurangi biaya karena tidak memerlukan alat pengenal khusus selain kamera yang biasanya menjadi perangkat umum pada komputer. Akan tetapi, implementasi sistem dalam mengenali wajah harus dirancang dengan baik dan akurat agar sifat alami yang diberikan metode pengenalan wajah bisa dimanfaatkan dengan maksimal.

Berbeda dengan pengenalan wajah manusia yang mudah dilakukan setiap saat oleh manusia, bagi komputer pengenalan fitur wajah manusia adalah sebuah tugas yang sulit. Masalah ini timbul karena komputer diharuskan untuk melakukan klasifikasi wajah dengan benar dengan berbagai situasi dan kondisi seperti pencahayaan yang gelap, dan tangkapan gambar latar belakang yang ada [4]. Sampai ini algoritme pengenal wajah masih menjadi topik penelitian yang menarik dan terus berkembang. Pada umumnya cara komputer merepresentasikan wajah dibagi dalam dua metode pendekatan [5] sebagaimana dijabarkan dalam tabel 1.

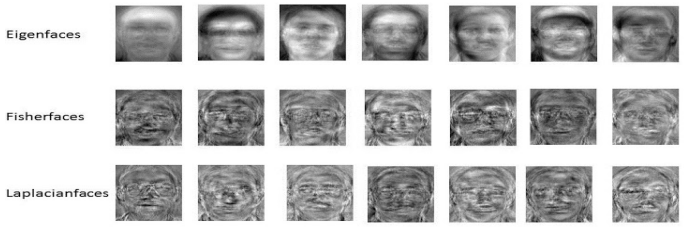
Perancangan sistem menggunakan Raspberry Pi 3 Model B untuk mekanisme kunci pintu dengan pengenalan wajah. Alasan utama menggunakan Raspberry Pi adalah portabilitasnya, selain dari faktor harga yang murah yang menjadi suatu kelebihan bagi pengguna sistem ini [6]. Perangkat Raspberry Pi populer untuk berbagai aplikasi sistem tertanam dan aplikasi *Internet of Things* (IoT).

Tabel 1. Pendekatan representasi gambar wajah

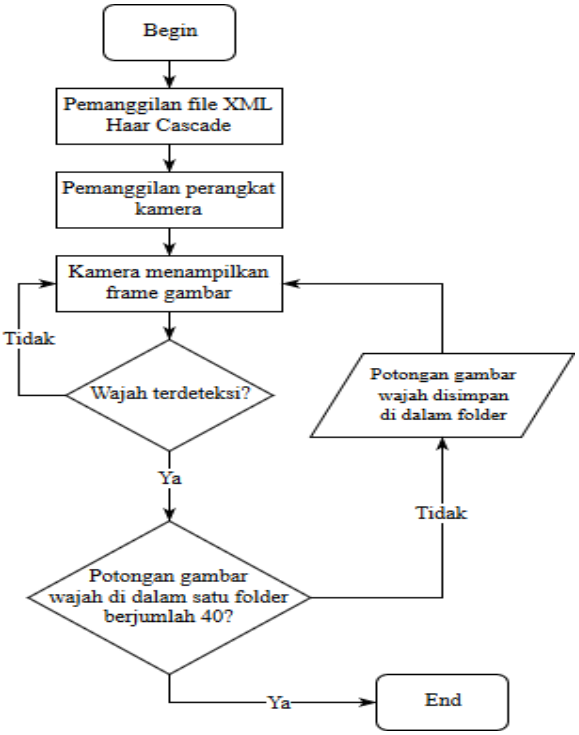
Metode	Pendekatan	Keterangan
Eigenfaces	*Appearance-based*	Menangkap variasi atau keragaman pada kumpulan gambar wajah dan menggunakan informasi yang ada untuk *encode* dan membandingkan gambar dari setiap wajah dengan karakter holistik. (Zhang & Turk, 2008).
Fisherfaces	*Feature-based*	Menggunakan representasi basis vektor pada fitur wajah (Martinez, 2011)

Tabel 2. Perbandingan antara berbagai teknologi biometrika

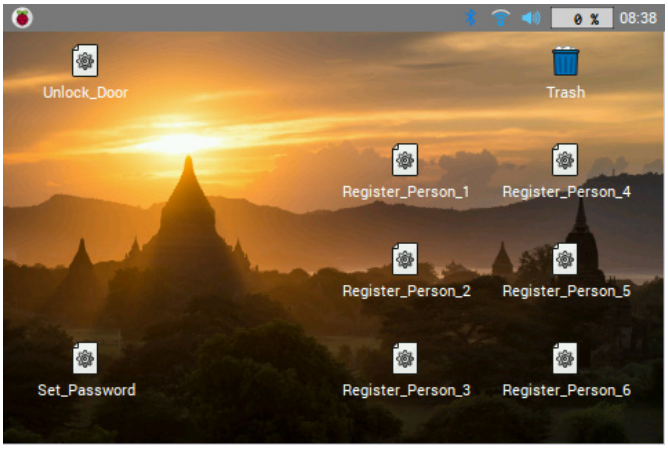
Metode	Akurasi	Biaya	Tingkat Penerimaan	Alat yang Dibutuhkan
Wajah	Sedang	Rendah	Sedang	Kamera
Tangan	Rendah-Sedang	Sedang	Sedang	*Scanner*
Sidik jari	Tinggi	Sedang	Sedang	*Scanner*
Retina	Tinggi	Tinggi	Rendah	Kamera
DNA	Tinggi	Tinggi	Rendah	*Scanner* DNA
Suara	Sedang	Sedang	Tinggi	Mikrofon
Tanda tangan	Rendah	Sedang	Tinggi	Pen optik



Gambar 1. Perbandingan rekonstruksi wajah metode *eigenfaces* dan *fisherfaces*



Gambar 2. *Flowchart* aplikasi pengambilan *dataset* wajah



Gambar 3. Tampilan antarmuka sistem penguncian pintu dengan metode pengenalan wajah

Raspberry Pi memiliki konsumsi daya CPU dan GPU yang rendah, yang sekaligus memiliki nilai kelebihan dan kekurangan.

Dengan spesifikasi tegangan dan arus input yang disarankan sebesar 5V/2.5A (Raspberry, n.d.), Raspberry Pi 3 Model B bisa dikatakan *low voltage*. Raspberry Pi 3 Model B juga dapat bekerja dengan baik pada tegangan dan arus input 5V/2A, sehingga dapat menggunakan *power bank* sebagai sumber tegangan eksternal. Kesimpulan awalnya, kebutuhan *power* yang rendah akan berdampak pada performa yang biasanya buruk, akan tetapi cukupan apabila mengingat harganya yang murah.

Berdasarkan uraian latar belakang masalah tersebut maka tujuan penelitian ini adalah untuk menguji berbagai kombinasi dari algoritme deteksi wajah berdasarkan kriteria sensitivitas, spesifisitas, dan *false rate*. Batasan penelitian yang ditetapkan adalah terkait metode kombinasi algoritme yang diuji yakni *Eigenfaces*/PCA dan K-*Nearest Neighbor* (K-NN), PCA-LDA dan K-NN, Eigenfaces/PCA dan *Support Vector Machine* (SVM), PCA-LDA dan SVM. Batasan juga ditetapkan pada implementasi pengujian hanya pada sistem Raspberry Pi 3. Luaran dari penelitian ini diharapkan dapat menghasilkan rancangan prototipe sistem keamanan kunci pintu dengan implementasi teknologi pengenalan wajah. Selain itu kontribusi teori yang diharapkan adalah untuk menemukan metode pengenalan wajah yang efektif dan efisien sesuai dengan performa Raspberry Pi 3.

2. Tinjauan Pustaka

Untuk melakukan penelitian mengenai bagaimana perancangan sistem keamanan rumah tangga menggunakan implementasi teknologi pengenalan wajah, saat ini terdapat berbagai penelitian serupa. Di antaranya penelitian yang dimuat di dalam makalah yang berjudul *Real Time Access Control Based on Face Recognition* [7] Penelitian ini dilakukan oleh Ylber Januzaj, Artan Luma, Ymer Januzaj, dan Vehbi Januzaj dari South East European University pada tahun 2015. Tujuan dari penelitian ini adalah menyediakan sistem keamanan akses yang mengontrol orang keluar dan masuk dari berbagai gedung menggunakan *magnetic lock*, sesuai dengan permintaan yang ada. Pada makalahnya, Januzaj membahas landasan teori tentang tiga metode yang dapat digunakan untuk membuka akses pintu, yaitu metode akses menggunakan kata kunci, akses menggunakan RFID, dan akses menggunakan teknologi biometri. Bagaimana celah pada penelitian-penelitian sebelum adalah kurangnya uji analisis perbandingan gabungan algoritme dan cenderung hanya menguji satu atau masing-masing algoritme.

Setiap manusia memiliki identitas bawaan yang sifatnya unik, oleh karena itu sistem keamanan yang berbasis biometri sangat dianjurkan untuk digunakan di dalam lingkungan yang membutuhkan tingkat keamanan yang tinggi. Teknologi biometri adalah teknologi yang paling aman digunakan karena fitur-fitur pada manusia seperti wajah, jari, dan suara tidak bisa dipinjamkan atau dicuri. [8]. Berdasarkan riset yang telah dilakukan oleh mereka, sejumlah empat faktor yang menjadi perbandingan pada beberapa sistem keamanan berbasis teknologi biometri ini yaitu akurasi, biaya, tingkat penerimaan, dan peralatan yang dibutuhkan, dijelaskan ke dalam tabel 2.

Tabel 2 menjelaskan perbandingan antara berbagai teknologi biometri, penggunaan metode pengenalan wajah memiliki tingkat akurasi yang sedang apabila dibandingkan dengan metode lainnya. Akan tetapi, tingkat akurasi yang sedang tersebut bisa ditingkatkan dengan cara memperbaiki kondisi lingkungan sekitar agar menjadi lebih kondusif, contohnya pencahayaan. Dibandingkan dengan beberapa metode lain, penggunaan metode pengenalan wajah juga murah dalam segi biaya. Biaya yang murah untuk metode pengenalan wajah disebabkan oleh kesederhanaan dari alat yang dibutuhkan, yaitu kamera. Saat penelitian ini ditulis, harga modul kamera untuk Raspberry Pi relatif murah dibandingkan alat lainnya, yaitu berkisar antara Rp300.000 sampai dengan Rp400.000. Faktor terakhir tentang perbandingan antara teknologi biometri adalah tingkat penerimaan, yang mana metode pengenalan wajah memiliki tingkat penerimaan sedang [9]. Untuk meningkatkan keamanan dari sistem, maka diperlukan penyetelan agar dapat mengurangi tingkat penerimaan tersebut.

Pengenalan wajah mengidentifikasi beberapa fitur wajah dengan mengekstrak fitur pada gambar yang menampilkan wajah subjek. Contohnya, sebuah algoritme yang menganalisis posisi, bentuk, atau ukuran dari mata, hidung, bibir, dan dagu. Hasil dari pengukurannya disimpan ke dalam *dataset* dan menghasilkan *facial metrics* [10]. Fitur ini yang akan digunakan untuk *template matching* pada teknik konvensional. Teknik konvensional seperti

template matching adalah satu dari beberapa sistem yang banyak diterapkan untuk metode pengenalan wajah (gambar 3).

Template matching pada teknik pengenalan wajah adalah proses pencarian lokasi dari fitur-fitur wajah yang penting dan menonjol dan urutan representasi dari wajah. *Template matching* menjadi salah satu teknik yang esensial digunakan dalam aplikasi analisis gambar [11]. Di dalam implementasi *template matching*, fitur wajah yang sudah diekstrak dibandingkan dengan data yang sudah disimpan untuk pengenalan wajah. Akan tetapi, penggunaan *template matching* untuk pengenalan wajah hanya cocok untuk kondisi lingkungan yang stabil. *Template matching* sederhana tidak akan mampu mengenali wajah pada tingkat perubahan cahaya yang signifikan. Oleh karena itu, demi meningkatkan akurasi sistem pada proses pengenalan wajah maka diperlukan metode lain yang dapat membaca garis-garis wajah yang dideteksi dengan lebih baik.

Metode *eigenfaces* dipengaruhi oleh teknik yang dinamakan *principal component analysis* (PCA) untuk merepresentasikan gambar wajah secara efisien. Pada kumpulan gambar wajah yang tersedia, PCA menghitung sistem koordinat terbaik untuk kompresi gambar, di mana setiap koordinat adalah gambar yang disebut *eigenpicture*. Sirovich dan Kirby [12] menyatakan bahwa di dalam prinsip ini setiap koleksi gambar wajah dapat direkonstruksi dengan cara menyimpan koleksi kecil dari berat nilai setiap gambar dan set kecil dari gambar standar (*eigenpicture*). Berat nilai yang menggambarkan setiap wajah ditemukan dengan memproyeksikan gambar wajah ke dalam setiap *eigenpicture*.

Berbeda dengan *eigenfaces* yang menggunakan teknik *principal component analysis* (PCA), metode *fisherfaces* menggunakan model proyeksi linier yang diskriminan atau biasa disebut *linear discriminant analysis* (LDA). Metode *fisherfaces* ini pertama kali diajukan oleh Belheumur *et al.* [13] yang telah mereka dibuktikan bahwa implementasi metode ini mampu mengurangi *error rate* yang dihasilkan oleh metode *eigenfaces* pada lingkungan eksperimen yang sama. LDA memaksimalkan rasio antar kelas dengan penyebaran di dalam kelas, oleh karena itu, *fisherfaces* bekerja lebih baik daripada PCA pada *eigenfaces* untuk tujuan diskriminasi. Implementasi *fisherfaces* sangat berguna saat gambar wajah memiliki variasi pencahayaan dan ekspresi wajah yang besar. Perbedaan hasil rekonstruksi citra wajah kedua algoritme ini ditunjukkan pada gambar 1.

3. Metodologi

Jenis penelitian yang digunakan adalah penelitian kuantitatif yakni penelitian yang digunakan untuk meneliti pada populasi atau sampel tertentu, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat statistik, dengan tujuan untuk menguji hipotesis yang telah ditetapkan [14]. Penelitian ini tergolong jenis penelitian kuantitatif karena data penelitian berupa angka-angka dan analisis menggunakan statistik yang diuji terukur melalui percobaan empiris. Pengumpulan data dalam penelitian ini dilakukan dengan studi literatur berupa mengumpulkan pembahasan jurnal, atau makalah yang ada kaitannya dengan judul penelitian dan observasi dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

Metode pengukuran efisiensi dan efektivitas algoritme yang digunakan adalah teknik *cross-validation* yang merupakan metode statistik untuk mengevaluasi dan membandingkan algoritme pembelajaran dengan membagi data menjadi dua segmen: satu digunakan untuk belajar atau melatih model dan yang lainnya digunakan untuk memvalidasi model. [15]. *Cross-validation* digunakan untuk mengevaluasi atau membandingkan algoritme pembelajaran sebagai berikut: pada setiap iterasi, satu algoritme *learning* atau yang lain menggunakan data k – 1 untuk mempelajari satu model atau lebih, dan selanjutnya model yang dipelajari diminta untuk membuat prediksi tentang data di dalam *validation fold*. Performa setiap algoritme *learning* pada setiap *fold* dapat dilacak dengan menggunakan beberapa *performance metric* yang telah ditentukan seperti akurasi. Setelah selesai, sampel k dari *performance metric* akan tersedia untuk setiap algoritme.

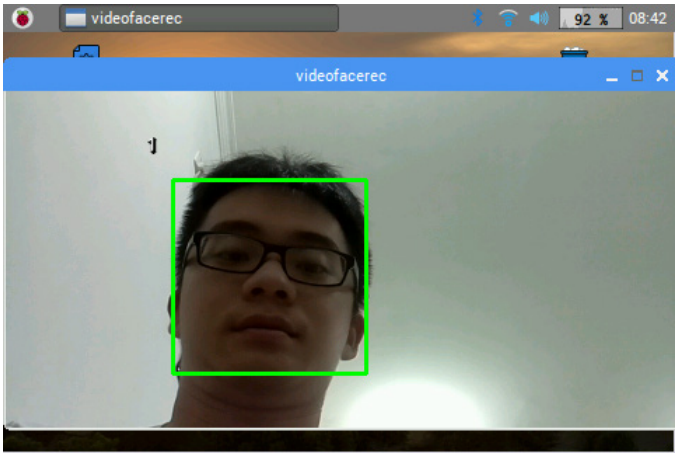
Dalam validasi silang k-fold, data dipartisi terlebih dahulu menjadi segmen atau lipatan berukuran sama (atau hampir sama). Selanjutnya iterasi k pada *training* dan validasi dilakukan sedemikian rupa sehingga dalam setiap iterasi data yang berbeda dilangsungkan untuk proses pengetesan, sedangkan fold sisanya k – 1 akan digunakan untuk proses *training*. Data biasanya berupa data bertingkat sebelum dipecah menjadi *k-fold*. Setelah itu dilakukan proses penyusunan ulang data untuk memastikan setiap *fold* merupakan hasil representasi yang baik dari keseluruhan. Sebagai contoh, pengaturan $k = 2$ akan menghasilkan 2-fold *cross-validation*. Pada 2-fold *cross-validation*, secara acak *dataset* dibagi menjadi dua yaitu d_0 and d_1, sehingga kedua set memiliki ukuran yang sama. Setelah itu, d_0 dan d_1 masing-masing dites, lalu diikuti dengan melakukan *training* pada d_1 dan testing pada d_0. Ketika $k=n$ (jumlah observasinya), *k-fold cross-validation* yang dipakai akan sama dengan *leave-one-out cross-validation*.

Untuk program pengambilan *dataset* wajah, ada tiga proses yang dijalankan. Yang pertama, program melakukan pemanggilan file XML Haar Cascade versi alternatif kedua yang dinamakan *haarcascade_frontalface_alt2.xml* secara *default*. Kedua, program akan melakukan pemanggilan perangkat kamera (RPi kamera pada Raspberry Pi). Ketiga, kamera menampilkan *frame* gambar dengan rate 30 *frame* per detik (fps). Ketika program mendeteksi ada wajah di dalam *frame* gambar, maka potongan wajah yang ditangkap oleh XML disimpan di dalam folder sebagai *dataset* wajah untuk *training*. Apabila jumlah potongan gambar wajah yang sudah tersimpan mencapai 40 buah, maka program diinstruksikan untuk berhenti mengambil dan menyimpan potongan gambar wajah. Proses ini dijelaskan dalam *flowchart* pada Gambar 2.

Ada tujuan di balik pembatasan jumlah potongan gambar wajah yang disimpan ke dalam *dataset* yang hanya sejumlah 40 buah per folder (atau per orang). *Dataset* wajah yang digunakan di dalam sistem penguncian pintu berbasis metode pengenalan *wajah* terdiri dari 40 folder untuk sampel wajah *true negative* yang diambil dari AT&T *Faces Database* di mana masing-masing sampel wajah terdiri dari 40 potongan gambar, dan 6 folder untuk sampel wajah *true positive*. Potongan gambar wajah untuk sampel wajah *true positive* yang disimpan melalui program berjumlah 40 buah akan mengurangi gejala *underfitting* atau *overfitting* karena memiliki bobot yang setara dan masuk akal dengan sampel wajah *true negative* yang digunakan. Gejala *underfitting* atau *overfitting* akibat perbedaan jumlah sampel akan memperburuk tingkat keyakinan sistem dalam melakukan pengenalan wajah.

	o0	o1	o2		o0	o1	o2		o0	o1	o2
c0	A	B	B	c0	B	A	B	c0	B	B	A
c1	A	B	B	c1	B	A	B	c1	B	B	A
c2	A	B	B	c2	B	A	B	c2	B	B	A

Gambar 4. Teknik matrix cross validation



Gambar 5. Tampilan program pengenalan wajah saat membuka kunci pintu

Pembuatan antarmuka sistem untuk masing-masing pilihan menu menggunakan *Bash shell* yang diletakkan pada *desktop* Raspbian Jessie. *Bash* sendiri merupakan nama bahasa pemrograman pada Linux yang fungsinya sebagai *shell*, yaitu memberikan urutan perintah kepada terminal dan perintah tersebut dieksekusi. Gambar tampilan antar muka program ditunjukkan dengan gambar 3.

Ada beberapa alasan yang kuat mengapa *Bash* diimplementasikan untuk antarmuka sistem [16]. Yang pertama adalah menampilkan pilihan menu secara instan ketika perangkat Raspberry Pi dinyalakan. Kedua, *Bash* tidak akan membebani prosesor sehingga tidak mempengaruhi kinerja program pengenalan wajah secara keseluruhan. Ketiga, *Bash* sangat mudah untuk dibuat. Cara untuk membuat file *Bash* adalah membuat file baru yang kosong, edit file tersebut menggunakan *text editor*, dan kemudian memasukkan *command* yang ingin dieksekusi apabila file *Bash* tersebut dijalankan. File kosong akan otomatis menjadi file *Bash* ketika di baris 1 (gambar 4).

Untuk program pengenalan wajah, ada tiga proses tugas *machine learning* yang harus dilakukan oleh sebuah perangkat keras atau komputer dalam implementasinya melakukan *training* pengenalan wajah.

Yang pertama adalah pengenalan terhadap fitur wajah yang diambil dari *dataset* menggunakan teknik PCA atau perpaduan antara teknik PCA dan LDA. PCA sendiri digunakan untuk memperkirakan fitur *dataset* wajah ke dalam fitur *vector* yang memiliki dimensional yang lebih rendah. Di dalam perpaduan implementasi antara PCA dan LDA, metode ini terbagi ke dalam dua langkah yaitu memproyeksikan gambar wajah dari ruang *vector* asli ke *subspace* dengan PCA, kemudian menggunakan LDA untuk mendapatkan linear *classifier* terbaik. Ide konsep untuk memadukan PCA dan LDA adalah untuk meningkatkan kemampuan generalisasi LDA apabila hanya tersedia sedikit sampel per satu *class*. Di sisi lain, LDA meningkatkan nilai diskriminan dari fitur PCA.

Yang kedua, melakukan pengklasifikasian *dataset* wajah menggunakan metode *classifier* yang dianggap cocok untuk sistem ini, yaitu k-*nearest neighbor* dan *support vector machine*. Keduanya memiliki representasi yang berbeda pada pendekatan dalam proses *learning*. *K-nearest neighbor* berusaha untuk memperkirakan distribusi data yang mendasarinya secara non-parametrik, sedangkan SVM mengasumsikan ada *hyperplane* yang memisahkan titik data, yang merupakan asumsi yang cukup ketat. Yang terakhir, proses verifikasi pengenalan wajah diakhiri dengan validasi data menggunakan k-fold *cross validation*. Cara kerja validasi data yang ditugaskan kepada metode k-fold *cross validation* adalah sebagai berikut:

Data yang sudah melewati tahap pengenalan fitur dan klasifikasi, k-*fold cross validation* membagi data tersebut menjadi *fold* atau lipatan sebanyak *k* yang besarnya sama dan tidak *overlapping* untuk proses *training* dan testing. Sebagai contoh, pada gambar 4 merupakan 3-fold *cross validation* untuk 9 data observasi and 3 class, sehingga setiap observasi diberikan *index* [c_i][o_i] seperti pada gambar 4.

Model dari hasil *training dataset* wajah disimpan ke dalam *file pickle*. *Pickle* sendiri merupakan sebuah modul Python untuk serialisasi data, yang mana akan mengubah objek *dataset* wajah menjadi sebuah *file byte stream*. Objek *dataset* wajah tersebut pada dasarnya adalah sebuah *classifier* dari *machine learning* yang harus dipanggil setiap kali menjalankan program. Fungsi *pickle* di dalam program pengenalan wajah ini adalah menyimpan objek *dataset* wajah dalam format ekstensi (*.pkl) sehingga sistem hanya memerlukan satu kali proses *training* dan setelahnya *file pickle* dapat dipanggil tanpa perlu mengulang proses *training* yang membutuhkan waktu relatif lama.

Proses *training* selesai setelah melalui tahap pengenalan fitur, klasifikasi, dan validasi data wajah. *File pickle* yang sudah disimpan dapat langsung digunakan untuk program pembuka pintu, yang mengakibatkan waktu yang dibutuhkan untuk peluncuran program menjadi instan. Program dianggap sudah berjalan ketika keberadaan perangkat RPi kamera berhasil dideteksi dan kamera mampu menampilkan *frame* gambar sebanyak 30 *frame* per detik (fps).

Tabel 3 Hasil pengukuran sensitivitas

Nilai k-fold	*PCA, kNN*	*PCA-LDA, kNN*	*PCA, SVM*	*PCA-LDA, SVM*
10-fold	40 (1)	40 (1)	40 (1)	40 (1)
8-fold	40 (1)	40 (1)	40 (1)	40 (1)
6-fold	40 (1)	40 (1)	40 (1)	39 (0.975)
5-fold	40 (1)	40 (1)	39 (0.975)	39 (0.975)
4-fold	39 (0.975)	39 (0.975)	39 (0.975)	39 (0.975)
3-fold	39 (0.975)	39 (0.975)	38 (0.95)	37 (0.925)
2-fold	36 (0.9)	36 (0.9)	36 (0.9)	36 (0.9)

Tabel 4 Hasil uji spesifisitas

Nilai k-fold	*PCA, kNN*	*PCA-LDA, kNN*	*PCA, SVM*	*PCA-LDA, SVM*
10-fold	1600 (1)	1578 (0.986)	1600 (1)	1571 (0.982)
8-fold	1591 (0.994)	1567 (0.979)	1600 (1)	1569 (0.981)
6-fold	1600 (1)	1574 (0.984)	1583 (0.989)	1547 (0.967)
5-fold	1600 (1)	1567 (0.979)	1559 (0.974)	1538 (0.961)
4-fold	1586 (0.991)	1555 (0.972)	1553 (0.971)	1528 (0.955)
3-fold	1531 (0.957)	1539 (0.962)	1524 (0.953)	1456 (0.91)
2-fold	1424 (0.89)	1372 (0.9)	1404 (0.878)	1304 (0.815)

Tabel 5 Hasil pengukuran *false positive rate*

Nilai k-fold	*PCA, kNN*	*PCA-LDA, kNN*	*PCA, SVM*	*PCA-LDA, SVM*
10-fold	0 (0)	22 (0.014)	0 (0)	29 (0.018)
8-fold	9 (0.06)	33 (0.021)	0 (0)	31 (0.019)
6-fold	0 (0)	26 (0.016)	17 (0.011)	53 (0.033)
5-fold	0 (0)	33 (0.021)	41 (0.026)	62 (0.039)
4-fold	14 (0.009)	45 (0.028)	47 (0.029)	72 (0.045)
3-fold	69 (0.043)	61 (0.038)	76 (0.048)	144 (0.09)
2-fold	176 (0.11)	228 (0.143)	196 (0.123)	296 (0.185)

Hampir sama seperti program pengambilan *dataset* wajah, ketika *frame* gambar yang ditampilkan berhasil menemukan wajah manusia berdasarkan *file* XML Haar Cascade, maka program pengenalan wajah ini akan berusaha menemukan wajah. Perbedaannya, program juga sekaligus memiliki tugas tambahan yaitu menebak siapa orang yang berada di dalam *frame* gambar tersebut. Jika wajah yang ditangkap kamera dianggap cocok dengan sampel *true positive* yang berasal dari file *pickle* berisi informasi terenkripsi berupa *byte stream*, maka program pengenalan wajah ini akan menggerakkan solenoid kunci pintu melalui perintah GPIO dan membukakan akses pintu. Selain kondisi tersebut, program tidak akan memerintahkan GPIO untuk mengaktifkan rangkaian solenoid. Tampilan program aplikasi saat menangkap wajah dapat dilihat pada gambar 5.

4. Hasil dan Diskusi

Pengujian waktu pengenalan wajah dilakukan menganalisis metode pengenalan fitur, *classifier*, dan validasi *dataset* yang terbaik untuk diimplementasikan di dalam sistem penguncian pintu berbasis pengenalan wajah. Hasil-hasil pengujian di bawah ini adalah berdasarkan jumlah *dataset* yang digunakan yaitu berjumlah 1640 (40 gambar x 41 orang) dengan ukuran gambarnya masing-masing sekitar 75x100 piksel. Pengujian ini dilakukan sebanyak tiga kali setiap *fold* lalu ketiga hasil tersebut dirata-rata. Perangkat tersebut adalah Raspberry Pi 3 Model B. Pengujian ini menunjukkan bagaimana performa Raspberry Pi 3 Model B yang akan digunakan sebagai perangkat keras utama dalam sistem penguncian pintu ini bekerja dalam melakukan validasi *dataset* gambar wajah.

Pengujian pada dalam penelitian ini adalah pengujian yang dapat diukur bersifat statistika, antara lain *detection rate*, tingkat akurasi validasi, waktu validasi, dan waktu pengenalan wajah.

a. Detection Rate

Pengujian *detection rate* dilakukan dengan pencocokan gambar wajah dari *dataset* wajah. Hasil-hasil pengujian di bawah ini adalah berdasarkan jumlah *dataset* yang digunakan yaitu berjumlah 1640 (40 gambar x 1 sampel *true positive*, 40 gambar x 40 sampel *true negative*) dengan ukuran gambarnya masing-masing sekitar 75x100 piksel. Pengujian ini dilakukan sebanyak tiga kali setiap *fold* lalu ketiga hasil tersebut dirata-rata. Pengujian ini bertujuan untuk mengetahui hasil *system testing* yang diselesaikan oleh k-fold *cross validation* pada saat proses *training*. Berdasarkan jenisnya, pengujian *detection rate* ini dibagi menjadi 4, di antaranya:

1) Sensitivitas
2) Spesifisitas
3) *False possitive rate*
4) *False negative rate*

b. Sensitivitas

Untuk pengenalan wajah, sensitivitas menyatakan probabilitas sebuah wajah yang dideteksi bersifat cocok dengan gambaran wajah lain dari wajah yang sama. Semakin tinggi nilai sensitivitas maka semakin baik performa sistem dalam melakukan pencocokan wajah seseorang dengan wajah orang lain. Sebaliknya, semakin rendah nilai sensitivitas maka semakin buruk performa sistem dalam pencocokan wajah seseorang dengan wajah orang lain, sehingga memungkinkan terjadinya kesalahan dalam melakukan pengenalan wajah. Hasil pengukuran sensitivitas yang dibagi berdasarkan nilai k-fold untuk jumlah pengetesan sampel *true positive* sejumlah 40 gambar

dapat dilihat di dalam Tabel 3 di bawah ini.

Hasil pengujian menunjukkan setiap algoritme yang diuji dengan nilai K-fold di atas 8 sudah memiliki sensitivitas pengenalan wajah yang sangat bak mencapai 100% sesuai dengan jumlah sampel yang diberikan. Ketika jumlah K-fold dikurangi terlihat bahwa kemampuan sensitivitas pengenalan wajah juga menurun.

c. Spesifisitas

Untuk pengenalan wajah, spesifisitas menyatakan probabilitas sebuah wajah yang dideteksi bersifat tidak cocok dengan gambaran wajah lain dari wajah yang berbeda. Semakin tinggi nilai spesifisitas maka semakin baik performa sistem dalam melakukan pencocokan wajah seseorang dengan wajah orang lain. Sebaliknya, semakin rendah nilai spesifisitas maka semakin buruk performa sistem dalam pencocokan wajah seseorang dengan wajah orang lain, sehingga memungkinkan terjadinya kesalahan dalam melakukan pengenalan wajah. Hasil pengukuran spesifisitas yang dibagi berdasarkan nilai k-fold untuk jumlah pengetesan sampel *true negative* sejumlah 1600 gambar dapat dilihat di pada Tabel 4. Hasil pengujian tersebut menunjukkan bahwa algoritme PCA dan SVM serta algoritme PCA dan K-NN mendapat hasil terbaik dibandingkan dengan algoritme lainnya.

d. *False Positive Rate*

Untuk pengenalan wajah, *false positive rate* menyatakan probabilitas sebuah wajah yang dideteksi bersifat cocok dengan gambaran wajah lain dari wajah yang berbeda. Semakin tinggi nilai *false positive rate* maka semakin buruk performa sistem dalam melakukan pencocokan wajah seseorang dengan wajah orang lain, sehingga memungkinkan terjadinya kesalahan dalam melakukan pengenalan wajah. Sebaliknya, semakin rendah nilai *false positive rate* maka semakin baik performa sistem dalam pencocokan wajah seseorang dengan wajah orang lain. Hasil pengukuran *false positive rate* yang dibagi berdasarkan nilai k-fold untuk jumlah pengetesan sampel *true negative* sejumlah 1600 gambar dapat dilihat di dalam Tabel 5. Hasil pengujian menunjukkan bahwa algoritme PCA & K-NN mencacat hasil *false positive rate* terendah dibandingkan dengan algoritme lainnya.

e. *False Negative Rate*

Untuk pengenalan wajah, *false negative rate* menyatakan probabilitas sebuah wajah yang dideteksi bersifat tidak cocok dengan gambaran wajah lain dari wajah yang sama. Semakin tinggi nilai *false negative rate* maka semakin buruk performa sistem dalam melakukan pencocokan wajah seseorang dengan wajah orang lain, sehingga memungkinkan terjadinya kesalahan dalam melakukan pengenalan wajah. Sebaliknya, semakin rendah nilai *false negative rate* maka semakin baik performa sistem dalam pencocokan wajah seseorang dengan wajah orang lain. Hasil pengukuran *false negative rate* yang dibagi berdasarkan nilai k-fold untuk jumlah pengetesan sampel *true positive* sejumlah 40 gambar dapat dilihat di dalam Tabel 6. Hasil pengujian menunjukkan bahwa nilai *false negative rate* terendah dimiliki algoritme PCA & k-NN dan PCA & LDA K-NN. Hasil pengujian menunjukkan kedua algoritme gabungan tersebut memiliki nilai identik yang sama persis.

Tabel 6 Hasil pengukuran *false negative rate*

Nilai k-fold	*PCA, kNN*	*PCA-LDA, kNN*	*PCA, SVM*	*PCA-LDA, SVM*
10-fold	0 (0)	0 (0)	0 (0)	0 (0)
8-fold	0 (0)	0 (0)	0 (0)	0 (0)
6-fold	0 (0)	0 (0)	0 (0)	1 (0.025)
5-fold	0 (0)	0 (0)	1 (0.025)	1 (0.025)
4-fold	1 (0.025)	1 (0.025)	1 (0.025)	1 (0.025)
3-fold	1 (0.025)	1 (0.025)	2 (0.05)	3 (0.075)
2-fold	4 (0.1)	4 (0.1)	4 (0.1)	4 (0.1)

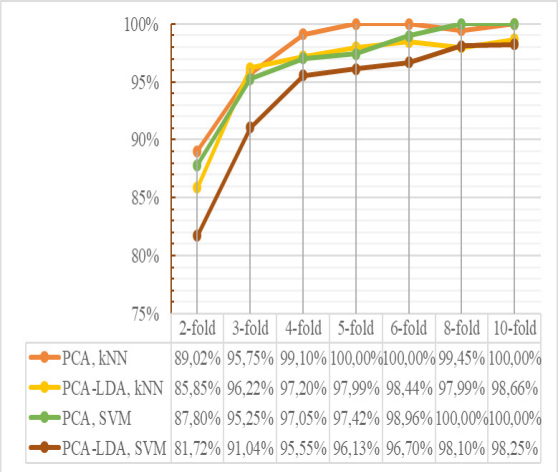
Tabel 7 Rata-rata waktu pengenalan wajah raspberry Pi 3

Nilai k-fold	PCA, kNN	PCA-LDA, kNN	PCA, SVM	PCA-LDA, SVM
10-fold	0.121	0.122	0.127	0.121
8-fold	0.126	0.129	0.125	0.122
6-fold	0.132	0.127	0.133	0.109
5-fold	0.121	0.120	0.122	0.121
4-fold	0.111	0.116	0.117	0.126
3-fold	0.125	0.128	0.122	0.128
2-fold	0.129	0.125	0.123	0.124

Melalui data yang disajikan di dalam Tabel 7, menunjukkan waktu yang dibutuhkan Raspberry Pi 3 Model B dalam melakukan proses pengenalan wajah pada masing-masing metode berkisar antara 0.109 sampai dengan 0.133 detik. Hasil ini sifatnya identik sehingga tidak dapat digunakan sebagai parameter untuk membandingkan masing-masing metode. Akan tetapi, ada yang dapat disimpulkan dari hasil tersebut, yaitu selama aplikasi pengenalan wajah berjalan, Raspberry Pi 3 Model B hanya mampu menghasilkan sekitar 8 hingga 9 *frame* per detik, dari target 30 *frame* per detik. Angka-angka yang dihasilkan di dalam Tabel 3 lebih dipengaruhi oleh performa komputasi Raspberry Pi 3 Model B yang cukup baik. Hasil pengukuran rata-rata waktu pengenalan wajah dapat dilihat di dalam Tabel 7.

Pengujian akurasi validasi membantu menganalisis metode *training dataset* yang paling baik untuk diimplementasikan di dalam sistem penguncian pintu berbasis pengenalan wajah. Hasil-hasil pengujian di bawah ini adalah berdasarkan jumlah *dataset* yang digunakan yaitu berjumlah 1640 (40 gambar x 41 orang) dengan ukuran gambarnya masing-masing sekitar 75x100 piksel. Pengujian ini dilakukan sebanyak tiga kali setiap *fold* lalu ketiga hasil tersebut dirata-rata. Berdasarkan jenisnya, pengujian akurasi validasi ini dibagi menjadi empat, di antaranya:

1) Metode Eigenfaces/PCA dan K-Nearest Neighbor (K-NN)
2) Metode hybrid PCA-LDA dan K-NN
3) Metode Eigenfaces/PCA dan Support Vector Machine (SVM)
4) Metode hybrid PCA-LDA dan SVM



Gambar 6. Hasil Uji perbandingan metode yang digunakan

Berdasarkan hasil dari grafik perbandingan pada Gambar 6, terlihat bahwa jika dilihat dari hasil akurasi validasinya secara keseluruhan, metode *Eigenfaces* atau PCA menghasilkan nilai rata-rata yang lebih baik dibandingkan metode *hybrid Eigen-Fisherfaces* atau PCA-LDA. Untuk penggunaan *classifiernya*, hasil akurasi validasi yang lebih baik dihasilkan oleh metode klasifikasi k-NN dibandingkan dengan metode klasifikasi SVM. Ciri khas dari metode klasifikasi SVM yang dirasa menjadi hambatan adalah implementasi *hyperplane* [17]. Objek yang digunakan di dalam sistem ini memiliki jumlah minimal 41 buah, sehingga dengan konsep *hyperplane* yang dirancang untuk melakukan klasifikasi sebanyak 2 *class* tersebut SVM tidak mampu meraih nilai akurasi sebaik metode *k-nearest neighbor*.

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dibahas dapat diambil kesimpulan bahwa metode *machine learning* yang paling baik untuk implementasi sistem penguncian pintu ini adalah menggunakan metode pengenalan fitur *hybrid Eigen-Fisherfaces* (PCA-LDA) dan metode *classifier k-nearest neighbor*. Kesimpulan ini diambil dengan melihat rasio performa berdasarkan durasi waktu *training* dan akurasi validasi dengan jumlah *fold* paling sedikit. Akurasi mendekati 90% dicapai dengan 2 *fold*, dan 4 fold dapat dicapai akurasi mendekati 100%.

Metode *machine learning* dan jumlah *fold* yang digunakan pada proses validasi tidak mempengaruhi performa berdasarkan durasi waktu pengenalan wajah. Selain itu nilai k yang lebih besar dalam melakukan validasi menggunakan *k-fold cross validation* akan menghasilkan akurasi yang lebih baik, disebabkan karena semakin besar nilai k dalam jumlah *dataset* yang sama maka jumlah data yang dapat digunakan untuk *training* akan semakin banyak.

Hasil penelitian ini juga menunjukkan bahwa perangkat Mikrokontroler Raspberry Pi 3 Model B yang menjadi perangkat keras utama dalam sistem ini memiliki performa yang cukup baik dalam hal pengenalan wajah. Hal ini dibukti melalui serangkai hasil uji coba yang telah dilakukan perangkat tersebut berhasil mengenali wajah dengan baik dengan rasio sukses 81%-100%

6. Daftar Pustaka

[1] K. Cherry, "Biometrics: An In Depth Examination," dalam *SANS Institute InfoSec Reading Room*, Chicago, Sans Institute, p. 11, 2014.

[2] K. Delac, M. Grgic dan S. Grgic, "Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set," *Wiley Periodicals,* vol. 15, pp. 252-260, 2015.

[3] K. Ignivov dan V. Rans, "Biometrics: Personal Identification in Networked Society," *International Journal of biometrics,* pp. 150-158, 2016.

[4] A. K. Jain, A. Ross dan S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 14, no. 1, pp. 4-20, 2014.

[5] S. Zhang dan M. Turk, "Eigenfaces," *Scholarpedia,* vol. 3, no. 9, p. 4244, 2008.

[6] A. Martinez, "Fisherfaces," *Scholarpedia,* vol. 6, no. 2, p. 4282, 2011.

[7] Raspberry, "Raspberry Pi FAQs - Frequently Asked Questions," Raspberry, [Online]. Available: https://www.raspberrypi.org/ help/faqs/#powerReqs. [Diakses March 2017].

[8] Y. Januzaj, A. Luma, Y. Januzaj dan V. Januzaj, "Real Time Access Control Based on Face Recognition," *International Conference on Network Security & Computer Science,* pp. 7-12, 2015.

[9] L.-H. Chan, S.-H. Salleh dan C.-M. Ting, "PCA, LDA and Neural Network for Face," *ICIEA - IEEE,* pp. 1256-1259, 2016.

[10] A. R. S. Siswanto, A. S. Nugroho dan M. Galinium, *"Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System,"* 2014.

[11] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*, Wiley, 2009.

[12] M. Sirovich dan A. Kirb, "Reviewing Eigenfaces for Recognition," *Journal of Neuroscience,* vol. III, no. 1, pp. 71-86, 2011.

[13] P. N. Belhumeur, J. P. Hespanha dan D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class-Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, No. 7,* pp. 711-720, 18 April 1997.

[14] Sugiyono, *Metode Penelitian Kuantitatif Kualitatif dan R&D*, Bandung: Penerbit Alfabeta, 2015.

[15] P. Refaeilzadeh, L. Tang dan H. Liu, "Cross-Validation," 6 11 2008. [Online]. Available: http://leitang.net/papers/ency-cross-validation.pdf. [Diakses 17 4 2017].

[16] M. Faundez-Zanuy, "Biometric Security Technology," *IEEE A&E Systems Magazine Vol. 21, No. 6,* pp. 15-26, 2014.

[17] R. Budiarto, "Manajemen Risiko Keamanan Sistem Informasi Menggunakan Metode FMEA dan ISO 27001 pada Organisasi XYZ," *Journal Of Computer Engineering System And Science,* vol. II, no. 2, pp. 105-115, 2017.