

Pembuatan Sistem Multimedia Pada Smart Mirror dengan Google Assistant

Sean Marshall^{#1}, Hapnes Toba^{#2}

[#]Program Studi SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Surya Sumantri No.65, Bandung

¹seanmarrshal@gmail.com

²hapnestoba@it.maranatha.edu

Abstract — Although Raspberry Pi has a mini size, it can do quite complex computing jobs, such as: digital image processing to recognize faces. In this research, facial recognition techniques are applied in combination with eigenfaces algorithm. The algorithm is capable to deliver accurate facial recognition with minimal exposure by making black and-white face detection. Other aspects which are investigated in this research is focused on the settings of Raspberry Pi associated with the multimedia system which aim to allow users interacting with camera via Raspberry Pi. The programming language used during the development is Python. The results show that eigenfaces algorithms using Python language is capable to detect faces during login activity on a computer system by combining Raspberry Pi and its camera.

Keywords— Image Processing, Eigenfaces Algorithm, OpenCV, Python.

I. PENDAHULUAN

Pada saat ini telah banyak dijumpai komputer mini berukuran kecil. *Raspberry Pi* memiliki kemampuan yang hampir sama dengan komputer pada umumnya namun memiliki keunggulan yaitu fitur *General Purpose Input/Output (GPIO)*. *GPIO* adalah pin yang terdapat pada *Raspberry Pi* untuk berinteraksi dengan fitur lain [1]. *Raspberry Pi* telah terbukti juga dapat digunakan sebagai *Home Automation* atau suatu sistem otomatisasi untuk memudahkan pemilik rumah dalam mengendalikan peralatan elektronik.

Raspberry Pi adalah sebuah komputer yang dikembangkan di Inggris dengan maksud mengajarkan ilmu komputer di sekolah. *Raspberry Pi* dapat pula digunakan sebagai *media entertainment* yang dapat difungsikan sebagai sistem kontrol atau *remote*. Alat ini dirancang untuk memudahkan dalam menyalakan dan mematikan beberapa alat elektronik yang ada di rumah atau dapat mengembangkannya ke berbagai alat dengan fungsi yang sama sebagai otomatisasi [1]. Dalam studi ini dirancang sebuah sistem multimedia dan penjadwalan (*scheduler*) untuk membantu rutinitas keseharian secara efektif yang dikemas dengan praktis dibantu oleh Google Assistant dalam integrasi setiap akun pengguna, misalnya untuk memudahkan proses login ke dalam komputer.

II. KAJIAN LITERATUR

Algoritma *computer vision* pertama kali digagas pada tahun 1982 di Jepang untuk topologi *contour* antar gambar [2]. Kemudian dikembangkan oleh Intel Research Lab sebagai aplikasi dengan pemakaian CPU terbanyak untuk dasar infrastruktur, dan dikeluarkan versi alpha ke publik di IEEE Conference on *Computer vision and Pattern Recognition* pada tahun 2000. Pemakaian *Raspberry Pi* berbasis OpenCV sudah banyak dilakukan diantaranya adalah *Image Data Gathering*. *Image Data Gathering* memanfaatkan *Raspberry Pi* kamera untuk mengambil gambar dan menyimpan di *Raspberry Pi* untuk langsung diproses [3]. Proyek lain yang menggunakan OpenCV dan *Raspberry Pi* adalah *Motion Detection*, *Sign Language Projek* dan *Object Tracking* [4].

A. Sistem Operasi Linux

Linux adalah sistem operasi yang berevolusi dari sebuah kernel yang diciptakan oleh Linus Torvalds ketika beliau masih menjadi mahasiswa di Universitas Helsinki. Dulunya Linux merupakan proyek hobi yang diinspirasi dari Minix, yaitu sistem UNIX kecil yang dikembangkan oleh Andrew Tanenbaum. Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991. Kemudian pada tanggal 5 Oktober 1991 Linus mengumumkan versi resmi Linux yaitu versi 0.02 yang berhasil mengeksekusi *shell bash* [5].

Saat ini Linux telah berkembang pesat sebagai suatu sub-sistem UNIX yang sangat lengkap, dan bisa digunakan untuk jaringan, pengembangan *software* dan bahkan untuk pekerjaan sehari-hari. Linux sekarang ini telah menjadi alternatif sistem operasi yang jauh lebih murah jika dibandingkan dengan sistem operasi komersial (misalnya Windows 10). Linux dikembangkan oleh beragam kelompok orang.

B. Raspberry Pi

Raspberry Pi adalah computer papan tunggal (*Single Board Circuit/SBC*) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi adalah *modul micro* komputer yang juga mempunyai *input output digital port* seperti pada *board microcontroller*. Di antara kelebihan Raspberry Pi dibanding *board microcontroller* yang lain yaitu mempunyai *Port / koneksi* untuk *display* berupa TV atau monitor PC serta koneksi USB untuk *keyboard* serta *mouse* [2]. Raspberry Pi dibuat di Inggris oleh Raspberry Pi *Foundation*.

Adapun fungsi komponen dari Raspberry Pi, yaitu [3]-[4]:

1. *SD Card Slot*
Media penyimpanan *software* pada sebuah *raspberry* adalah *card*. dimana *card* tersebut sebagai media penyimpan OS yang akan digunakan *raspberry*. Jenis-jenis *card* yang bisa digunakan yaitu Micro SD/MMC/SDIO
2. *LAN Port*
Pada sebuah *raspberry* terdapat 1 *LAN port* yang digunakan untuk mengakses jaringan
3. *Memory*
Untuk Raspberry Model 3B+ memiliki 1GB. *Memory* ini tidak utuh digunakan sebagai RAM, tetapi penggunaan memorinya dibagi dengan penggunaan memori GPU
4. *USB*
Jumlah *USB port* pada tipe-tipe *raspberry* berbeda, untuk Rev. A memiliki 1 buah *USB port*, untuk Rev. B memiliki 2 buah *USB port* dan untuk Rev. B+ memiliki 4 buah *USB port*. *USB port* yang digunakan adalah USB 2.0. Tarikan awal arus maksimumnya adalah 100 mA untuk jangka pendek
5. *Audio Output*
Audio output dalam bentuk 3,5mm *stereo audio socket*. Seperti halnya *analog video output*, *audio output* juga tidak digunakan bersamaan dengan HDMI, ketika HDMI digunakan maka *output* suara pada *audio output* tidak akan keluar

C. Computer Vision

Computer vision merupakan salah satu cabang dalam ilmu komputer yang bertujuan untuk membuat suatu keputusan yang berguna mengenai objek fisik nyata dan keadaan berdasarkan sebuah gambar atau citra. *Computer vision* merupakan kombinasi antara lain [6]:

1. Pengolahan Citra
Pengolahan Citra (*Image Processing*) merupakan bidang yang berhubungan dengan proses perubahan pada citra agar mendapatkan kualitas citra yang lebih baik
2. Pengolahan Pola
Pengenalan Pola (*Pattern Recognition*) merupakan bidang yang berhubungan dengan proses identifikasi objek pada citra atau interpretasi citra, yang bertujuan untuk mengekstrak informasi yang disampaikan oleh citra.

D. Proses Citra Digital

Pengolahan citra digital (*digital image processing*) adalah manipulasi dan interpretasi dari citra dengan bantuan komputer. Pengolahan citra biasanya digunakan untuk memperbaiki kualitas citra, melakukan proses penarikan informasi atau deskripsi objek yang terkandung dalam citra, dan melakukan kompresi atau reduksi data.

Berdasarkan dari jenisnya suatu citra dapat diklasifikasikan menjadi dua bentuk, yaitu: citra kontinu dan diskrit. Citra kontinu diperoleh dari sistem optik yang menerima sinyal analog, seperti halnya pada mata manusia dan kamera analog, sedangkan citra diskrit (citra digital) dihasilkan melalui proses digitalisasi terhadap citra kontinu.

E. OpenCV

OpenCV adalah suatu *library* yang dikembangkan oleh developer Intel Corporation. *Library* ini terdiri dari fungsi- fungsi *computer vision* dan *image processing* berkapasitas tinggi. OpenCV sangat disarankan untuk *programmer* yang akan berkecukupan pada bidang *computer vision*, karena *library* ini mampu digunakan untuk menciptakan aplikasi yang handal, kuat di bidang *digital vision*. Karena *library* ini bersifat cuma- cuma dan sifatnya yang *open source*, maka dari itu OpenCV tidak dipesan khusus untuk pengguna arsitektur Intel, tetapi dapat dibangun pada hampir semua arsitektur.

Library OpenCV telah memiliki lebih dari 2500 algoritma yang sudah teroptimasi, yang terdiri dari sekumpulan algoritma klasik dan algoritma *computer vision*. Algoritma ini bisa digunakan untuk mendeteksi dan mengenal wajah,

mengidentifikasi objek, mengklasifikasi tindakan manusia dalam video, melacak pergerakan kamera, melacak pergerakan objek, mengekstraksi model 3D dari objek, memproduksi poin 3D dari *stereo kamera*, menggabungkan gambar untuk memproduksi gambar yang memiliki resolusi tinggi, menemukan gambar dalam sebuah *database*, menghilangkan mata merah dari gambar yang diambil menggunakan *flash*, mengikuti pergerakan mata, mengenal pemandangan, serta membangun penanda untuk *augmented reality*.

F. Face Recognition

Pengenalan wajah (*face recognition*) merupakan suatu pengembangan dari teknologi deteksi wajah (*face detection*) yang dapat mengenali wajah dari hasil tangkapan kamera dan akan melakukan deteksi persamaan wajah dengan data wajah yang telah disimpan di *database* pada komputer. Dengan demikian komputer dapat mengenali dan mengetahui identitas wajah seseorang tersebut.

Cara kerja pengenalan wajah (*face recognition*) secara umum yaitu dengan mengkonversikan foto, sketsa, dan gambar video menjadi serangkaian angka, yang disebut dengan *faceprint* yang kemudian akan membandingkan dengan rangkaian angka yang lain yang mewakili wajah-wajah yang sudah dikenal. Proses pengenalan citra wajah oleh sistem secara garis besar dapat dibagi menjadi lima tahap, yaitu: deteksi, pengenalan posisi, normalisasi, pengkodean, dan perbandingan.

G. Algoritma Eigenfaces

Eigenface adalah algoritma pengenalan wajah yang berdasarkan pada *Principle Component Analysis* (PCA). Dengan kata lain, *eigenfaces* menggunakan perhitungan PCA sebagai pendekatannya [6]. *Eigenfaces* dapat ditentukan dengan prinsip-prinsip analisis komponen terhadap sekumpulan citra sampel dengan wajah terpusat dan terutama dengan ukuran yang setara. Pola wajah di dalam suatu citra dapat ditentukan dengan memindahkan *frame* pada sub-citra yang menutupi atau membatasi keseluruhan pola pada citra ke lokasi lain yang berukuran sama.

Eigenfaces dibuat dengan matriks dalam dimensi $[m \times n]$, yang kemudian dikonversi menjadi $[(m \times n) \times 1]$. Kemudian mencari *average face vector* dengan menghitung rata-rata *image vector* terhadap total vektor. Setelah itu, dihitunglah hasil *average face vector* dikurangi dengan nilai vektor setiap gambar. *Stack image vector* dan *average vector* digunakan hingga membentuk matriks $A = [(m \times n) \times i]$ dimana i adalah jumlah *image*. Tahap terakhir adalah dengan menghitung *covariance*-nya dengan persamaan $C = A \times \text{transpose}A$. Hasil akhir adalah *eigenvector* dan *eigenvalue* dari matriks C tersebut, dengan memilih *eigenvalue* sebagai konstanta yang bernilai lebih kecil dari i [7].

H. Python

Python merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Python dapat digunakan dalam berbagai sistem operasi, seperti Unix, Linux, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada kebanyakan sistem operasi Linux, bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya. Tipe data Python berupa:

1. *Boolean*, contoh *True and False*
2. *Complex*, pasangan angka *real* dan imajiner, misalnya $1 + 5j$
3. *Date*, bilangan yang dapat dikonversi menjadi format tanggal

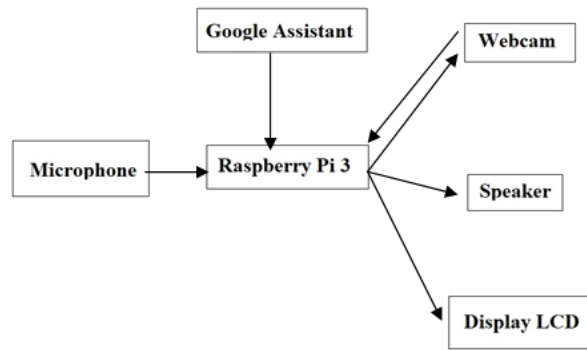
III. ANALISIS DAN RANCANGAN SISTEM

Analisis sistem merupakan penguraian suatu sistem ke dalam komponen untuk mempelajari bagaimana komponen-komponen tersebut bekerja dan berinteraksi.

A. Diagram Blok Rangkaian

Secara garis besar diagram blok rangkaian bertujuan untuk memudahkan proses analisis suatu rangkaian. Penjelasan diagram blok pada Gambar 1 untuk sistem yang dikembangkan dalam riset ini adalah sebagai berikut:

1. Microphone memberikan masukkan kata pada Raspberry Pi 3 seperti 'Ok, Google'
2. Raspberry Pi 3 menerima masukkan dari microphone kemudian menjalankan perangkat lunak kode javascript yang terdapat pada Raspberry Pi 3
3. Masukan kata dari microphone juga akan menjalankan *webcam* (pada Raspberry Pi) untuk mengambil gambar dan mengirimkan gambar ke Raspberry Pi 3
4. Raspberry Pi 3 akan menjalankan perintah yang telah tersedia di Google Assistant



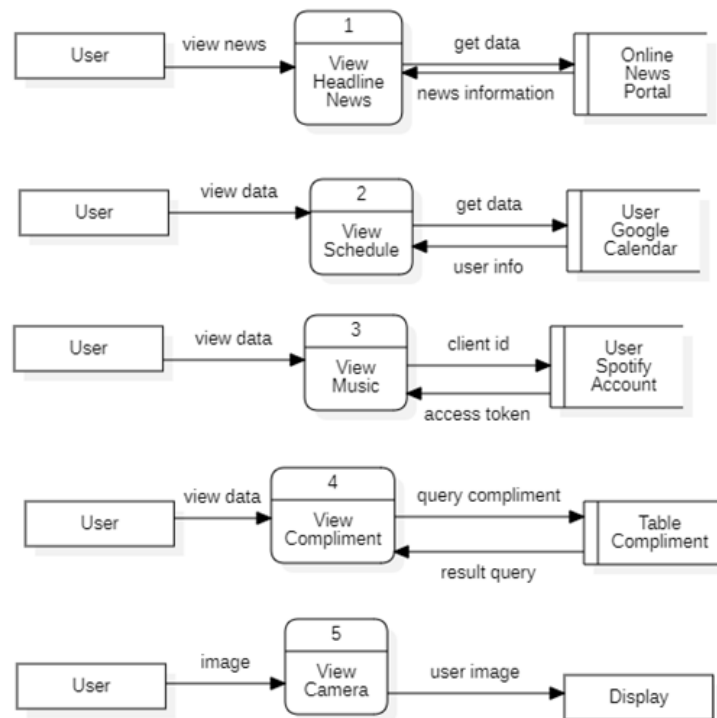
Gambar 1 Diagram Blok Rangkaian

B. Data Flow Diagram

Data Flow Diagram (DFD) adalah gambaran proses, aliran data, dan simpanan data secara grafis dalam suatu sistem. Pendekatan aliran data menekankan logika urutan proses yang mendasari sistem. Dengan menggunakan kombinasi dari empat simbol, penganalis sistem dapat menciptakan suatu gambaran proses yang bisa menampilkan dokumentasi sistem yang solid

1) *Data Flow Diagram Level 1*

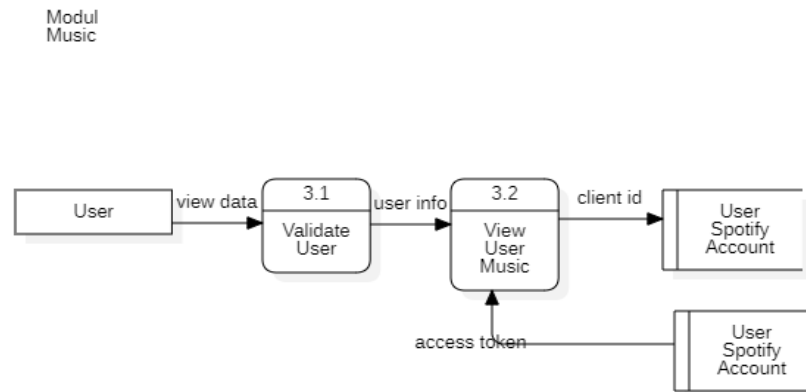
Pada Gambar 2, terlihat 5 modul utama yaitu Modul *Newsfeed*, Modul *Schedule*, Modul *Music*, Modul *Compliment* dan Modul Kamera. Untuk setiap modul, *user* akan mengirim data diri untuk diolah pada Raspberry Pi. Data tersebut akan diolah pada setiap *data store*, setelah melewati tahap tersebut kemudian dapat ditampilkan pada *display*.



Gambar 2 Dataflow diagram level 1

2) *Data Flow Diagram Level 2 Modul Music*

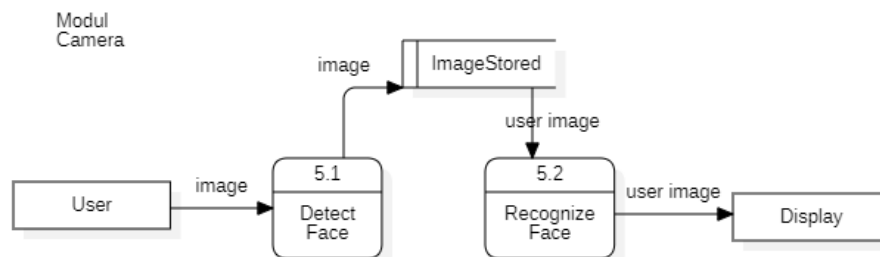
Gambar 3 merupakan diagram Modul *Music*. Dalam modul ini, hal yang dilakukan pertama oleh *user* adalah memvalidasi akun *Spotify User* dengan memberikan *client-id*. Saat *Spotify* mencocokkan dan sesuai dengan data maka akan memberikan akses untuk mengolah informasi milik *Spotify*.



Gambar 3 Dataflow Diagram level 2 Modul *Music*

3) Data Flow Diagram Level 2 Modul Kamera

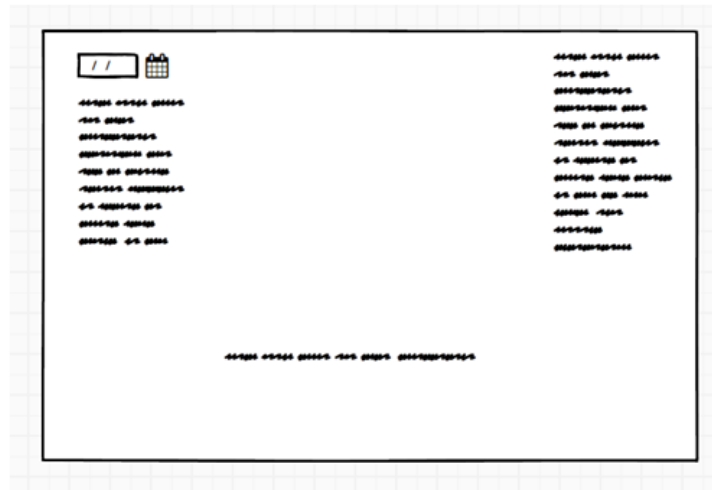
Gambar 4 adalah penjelasan rinci dari modul kamera. *User* akan mengirim gambar pada sistem untuk diproses selanjutnya di dalam *Detect Face*. Saat wajah dikenali kemudian dikirimkan pada gambar yang telah disimpan sebelumnya untuk dikenali. Setelah wajah dikenali maka akan ditampilkan oleh sistem, dan user diijinkan untuk masuk ke dalam sistem.



Gambar 4 Dataflow Diagram level 2 Modul Kamera

C. Perancangan Interface

Pada Gambar 5 diberikan perancangan tampilan awal pada program. Sudut kiri atas terdiri dari modul *alarm* yang berisi hari, tanggal, tahun, jam. Kemudian jadwal libur tahunan Indonesia. Pada bagian bawah adalah *quote* yang diambil dari internet untuk motivasi. Pada sudut kanan atas adalah jadwal tersedia hari itu dan beberapa hari ke depan



Gambar 5 Rancangan Interface

IV. HASIL IMPLEMENTASI

A. Tampilan Utama

Pada Gambar 6 diberikan realisasi tampilan utama ketika program dijalankan. Tampilan utama memiliki 5 modul, yaitu: *Newsfeed, Music, Scheduler, Kamera, dan Compliment*.

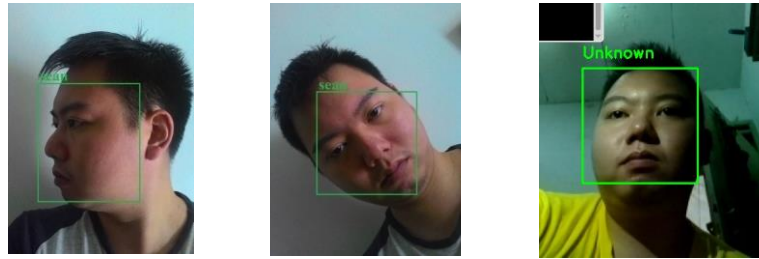


Gambar 6 Tampilan utama

B. Pengambilan Gambar Wajah

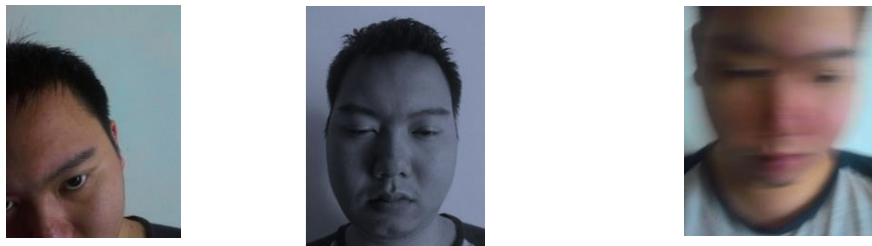
Saat mengambil gambar wajah pada modul kamera, ada beberapa faktor yang harus diperhatikan salah satunya adalah pencahayaan yang baik.

TABEL 1
TEKNIK PENGAMBILAN GAMBAR



Gambar yang baik untuk dapat dikenali oleh program perlu memiliki pencahayaan yang baik, serta dapat memperlihatkan bagian wajah yang jelas seperti mata, hidung, dan mulut. Contoh gambar-gambar yang baik dapat dilihat dalam Tabel 1.

TABEL 2
CONTOH GAMBAR TIDAK TERDETEKSI



Gambar pada Tabel 2 tidak dapat dideteksi dengan baik karena program tidak dapat mengenali mata, hidung atau mulut. Contoh lainnya saat gambar telah di *filter* sebelumnya, terakhir adalah gambar terlalu buram untuk dapat dideteksi.

V. PENGUJIAN

Sebelum program digunakan oleh pengguna, dilakukan proses pengujian agar program yang dibangun terbukti dapat berfungsi dengan baik. Proses pengujian ini juga dapat meminimalisir kesalahan yang menimbulkan ketidaknyamanan pengguna. Pengujian dilakukan pada perangkat keras Raspberry Pi secara *black box*, sesuai dengan perancangan yang disampaikan pada bagian III sebelumnya.

A. Pengujian Black Box

Hasil pengujian *black box* secara garis besar dapat dilihat pada Tabel 3. Fitur-fitur umum yang disampaikan dalam bagian III telah berhasil diimplementasikan dan hasilnya dapat berfungsi dengan baik. Adapun salah satu hal yang perlu diperhatikan dari hasil pengujian adalah keterbatasan Raspberry Pi yang hanya dapat efisien memproses sebanyak lima buah gambar, berukuran pada kisaran 100 dpi, di dalam *database*. Jika jumlah gambar ditambahkan waktu pemrosesan diperlukan lebih lama dan terkadang bahkan menyebabkan Raspberry Pi berhenti berfungsi secara tiba-tiba.

TABEL 3
HASIL PENGUJIAN *BLACK BOX*

No	Test Case	Hasil Harapan	Hasil Keluaran	Hasil Uji
1	Pengujian menampilkan <i>schedule</i> baru melalui smartphone	Mengisi form <i>schedule</i> pada <i>Google Calendar</i> melalui smartphone	<i>schedule</i> baru muncul pada layar kaca	Berhasil
2	Pengujian menampilkan <i>music</i> melalui smartphone	Memutar lagu melalui jaringan yang sama dengan aplikasi <i>Spotify</i>	Detail lagu muncul pada layar kaca	Berhasil
3	Pengujian <i>face recognition</i> pada kamera	Memasukkan wajah pada <i>dataset</i>	Wajah berhasil dideteksi	Berhasil

B. Pengujian Eigenfaces

Langkah pertama dalam sistem adalah pengambilan data dengan menggunakan *pi-camera*. Pengambilan data yang digunakan oleh *user* berupa citra RGB, selanjutnya data diolah dan diproses oleh sistem yang bertujuan untuk menghasilkan citra data *training*.

TABEL 4
GAMBAR PADA DATA TRAINING



Hasil citra RGB kemudian diproses hingga menjadi citra *grayscale*.

TABEL 5
HASIL KONVERSI CITRA RGB KE CITRA *GRAYSCALE*



Setelah melewati tahap mengubah citra RGB menjadi citra *grayscale*, dilanjutkan dengan pengenalan wajah pada data training. Hasil pengenalan wajah ditampilkan pada Gambar 6 dan telah berhasil mengenali bagian citra untuk digunakan sebagai akses (*login*) ke dalam komputer.

TABEL 6
HASIL PENGENALAN WAJAH



VI. SIMPULAN DAN SARAN

Kesimpulan umum terhadap hasil tercapai dalam adalah sebagai berikut:

1. Algoritma *eigenfaces* cocok digunakan melalui bahasa pemrograman Python versi 3 dengan ketersediaan *library* yang cukup lengkap selama proses pengolahan citra.
2. Penggunaan algoritma *eigenfaces* pada OpenCV dapat digunakan pada Raspberry meskipun memerlukan pengaturan ekstra untuk jumlah foto lebih dari lima citra.

Berdasarkan hasil penelitian, saran yang direkomendasikan adalah perlu ditambahkan fitur *auto update*, terutama pada aplikasi *Google Calendar* dan *Spotify* sesuai periode yang ditetapkan. Penggunaan algoritma *eigenfaces* pada OpenCV dapat menjadi pertimbangan untuk digunakan karena tidak memakan sumber daya yang banyak dan hasil yang akurat. Perlu dipertimbangkan untuk mengkoneksikan Raspberry Pi dengan sumber daya dari komputer host, agar proses pengolahan citra bisa dilakukan dengan lancar untuk jumlah foto yang banyak.

DAFTAR PUSTAKA

- [1] Drogon, "WiringPi," WP Weaver, 9 May 2014. [Online]. Available: www.WiringPi.com. [Accessed 22 September 2019].
- [2] A. K. Gary Bradski, *Learning OpenCV: Computer Vision with the OpenCV Library*, America: O'Reilly Media, 2008.
- [3] P. A. Eko, "Mengenal Raspberry Pi," UGM, 30 August 2012. [Online]. Available: <http://agfi.staff.ugm.ac.id/blog/index.php/2012/08/mengenal-raspberry-pi/>. [Accessed 16 October 2019].
- [4] F. Wicaksono, *Mudah Belajar Raspberry Pi Disertai 19 Contoh Proyek sampai dengan Proyek IOT, Soal Latihan dan Pembahasan*, Bandung: Informatika, 2018.
- [5] I. Sofana, *Mudah Belajar Linux*, Yogyakarta: Informatika, 2010.
- [6] H. Al Fatta, *Rekayasa Sistem Pengenalan Wajah*, Yogyakarta: ANDI, 2009.
- [7] E. P. W. Qadriza Mutiara Detila, "Perbandingan Metode Eigenface, Fisherface, dan LBPH pada Sistem Pengenalan Wajah," *Jurnal Ilmiah KOMPUTASI*, vol. 18, no. 4, p. 318, 2019.