

ORIENTASI CITRA SECARA OTOMATIS BERDASARKAN KEBERADAAN WAJAH MENGGUNAKAN FITUR HAAR-LIKE

Ahmad Hifdhul Abror¹, Handayani Tjandrasa²

¹Pusat Teknologi Informasi dan Pangkalan Data, Universitas Islam Negeri Sunan Ampel
abror@uinsby.ac.id

²Jurusan Teknik Informatika, FTIf, Institut Teknologi Sepuluh Nopember
handatj@its.ac.id

Abstrak

Perkembangan dan penggunaan teknologi kamera digital saat ini sudah sangat meningkat, ditandai dengan banyaknya jenis kamera serta disematkannya kamera pada berbagai perangkat seperti laptop, ponsel, tab, jam tangan dan gadget lainnya, sehingga kegiatan fotografi menjadi semakin mudah. Namun sebagian besar perangkat-perangkat tersebut tidak memiliki sensor untuk menyimpan informasi mengenai orientasi foto yang diambil apakah itu portrait atau landscape. Karena itu, kebanyakan foto-foto yang mengalami rotasi tidak sebagaimana mestinya baru disadari ketika disajikan di depan layar komputer atau televisi. Pada penelitian ini diajukan sebuah metode perbaikan orientasi citra secara otomatis menggunakan fitur haar-like dengan klasifikasi cascade adaboost untuk mendeteksi objek wajah manusia yang ada didalamnya, kemudian dari objek wajah yang ditemukan akan dijadikan acuan untuk perbaikan orientasi.

Kata Kunci: cascade adaboost; grayscale; haar-like; orientasi citra.

Abstract

The development and use of digital camera technology is now greatly improved, marked with the number of cameras and camera pinned on a variety of devices such as laptops, mobile phones, tabs, watches and other gadgets, so the activity of photography becomes easier. However, most of these devices do not have sensors to store information about the orientation of the photos taken whether it is portrait or landscape. Therefore, most of the photos are experiencing rotation is not as it should be and it is known when presented in front of the television or computer screen. In this study proposed a method of automatic fixation of image orientation using Haar-like features with a cascade adaboost classification to detect human face objects inside, then the human face objects found will be used as a reference for fixation of orientation.

Keywords: cascade adaboost; grayscale; haar-like; image orientation.

I. PENDAHULUAN

Penggunaan teknologi kamera digital saat ini sudah sangat berkembang, bahkan pada setiap perangkat elektronik bisa dilengkapi dengan teknologi ini. Pertumbuhan ponsel pintar atau biasa disebut *smartphone* beberapa tahun belakangan ini dan sampai sekarang masih berkembang pesat juga mempengaruhi banyaknya pengguna kamera. Semakin hari kualitas kamera yang di sematkan pada *smartphone* semakin bagus sehingga aktifitas berfoto menjadi semakin menyenangkan. Sejalan dengan kondisi tersebut, hasil foto yang diambil dapat bervariasi dalam hal orientasi atau dengan kata lain ada foto terotasi tidak sebagaimana mestinya.

Seiring dengan maraknya media jejaring sosial seperti *Facebook*, *Twitter*, *Path*, *Instagram* dimana terdapat fitur menarik yaitu interaksi sosial yang bisa melampirkan foto. Terdapat sebuah penelitian dari *Pew Research Center* yang mengungkapkan bahwa ada sekitar 91% anak muda mengunggah foto mereka

sendiri (*selfie*, narsis). Dari beberapa alasan yang disebutkan diatas, maka foto dengan objek manusia terutama wajah berpeluang besar mengalami tren peningkatan.

Secara umum orientasi citra digital dibagi menjadi 4 macam, dengan perbedaan putaran 90^0 . Citra yang tidak terotasi akan memiliki orientasi yang benar (0^0), citra akan terlihat terbalik jika orientasinya 180^0 dan akan tampak miring ke kiri atau ke kanan jika terotasi 90^0 atau 270^0 .

Pengenalan orientasi citra secara otomatis merupakan masalah yang cukup rumit. Otak manusia menggunakan pengenalan objek dan informasi kontekstual untuk mengenali orientasi citra dengan benar. Beberapa pengenalan objek yang biasa digunakan otak manusia misalnya manusia, wajah, benda, langit, tumbuhan, hewan, air, gedung dan tanah, kemudian dari objek tersebut dapat ditentukan apakah suatu foto salah orientasi atau tidak. Memperbaiki foto yang salah orientasi sebenarnya sangat mudah, hanya

dilakukan dengan memutar ke arah (orientasi) yang tepat. Namun jika banyak foto yang memiliki orientasi salah, maka akan memerlukan beberapa waktu jika dilakukan perbaikan secara manual.

Pada beberapa penelitian sebelumnya untuk mendeteksi orientasi citra secara otomatis bisa dengan cara memanfaatkan *low-level content features* seperti *luminance* (pencahayaannya) and *chrominance* (warna), hal ini dilakukan oleh [1][2][3]. Fitur *low-level* ini digunakan untuk mendeteksi perbedaan kecerahan warna pada sub region foto. Fitur kecerahan warna dari citra bisa digunakan untuk menentukan posisi keberadaan langit dari foto yang diambil di luar ruangan dan memiliki kondisi yang cerah seperti pada siang hari. Karena langit pada umumnya memiliki kekontrasan yang berbeda dan lebih terang dengan objek lain seperti bangunan atau tanah. Namun akan timbul masalah ketika foto yang diambil berada di dalam ruangan atau pada saat kurang cahaya atau malam hari. Karena itu, diperlukan penggabungan deteksi objek untuk perbaikan orientasi.

Penelitian ini mengusulkan sebuah metode untuk mendeteksi orientasi citra secara otomatis dengan memperhatikan keberadaan objek wajah. Metode deteksi wajah seringkali digunakan secara *real time* yang artinya memiliki keunggulan dalam hal komputasi secara cepat.

II. DESAIN SISTEM

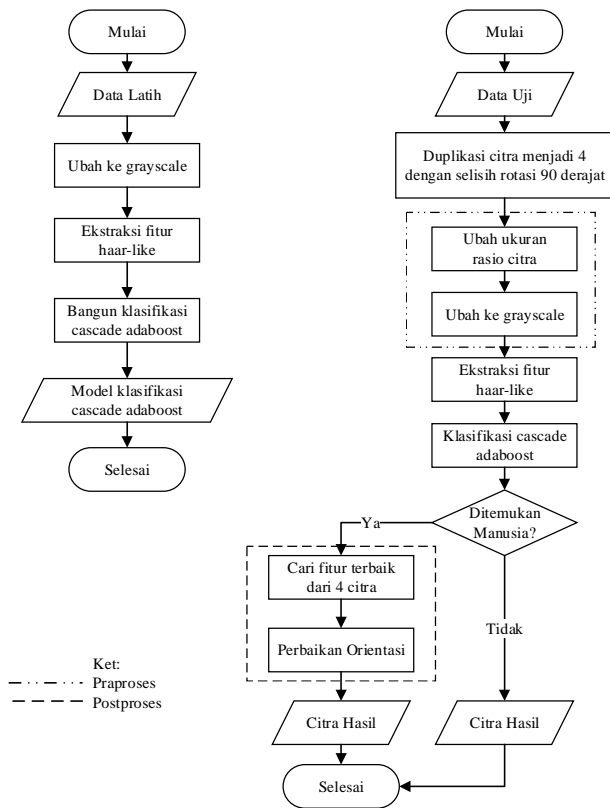
Diagram alir rancangan sistem yang diusulkan ditunjukkan pada Gambar 1. Pada penelitian ini fokus pada deteksi objek untuk perbaikan orientasi citra, karena itu rancangan sistem memiliki fungsi utama yaitu deteksi wajah. Pada akhir proses, hanya satu objek terbaik dari wajah yang akan diambil untuk acuan perbaikan orientasi citra dengan melakukan rotasi kearah yang benar.

Dari Gambar 1 terdapat dua diagram alir, sebelah kiri adalah diagram alir untuk proses pelatihan yang menghasilkan suatu model klasifikasi, sedangkan sebelah kanan adalah proses pengujian. Pada proses pengujian sebuah masukan berupa citra akan diduplikasi dan dirotasi menjadi 4 macam citra (dijelaskan lebih lanjut pada sub bab 3.1). kemudian dilakukan praproses terlebih dahulu sebelum dilakukan proses deteksi. Praproses pada deteksi wajah meliputi mengubah ukuran rasio citra dan ruang warna ke *grayscale*. Jika pada proses deteksi wajah ditemukan objek, maka dilakukan postproses yaitu dilakukan pencarian objek terbaik dari objek-objek yang sudah ditemukan pada 4 citra yang sudah diduplikasi dan dirotasi. Satu objek terbaik akan dijadikan acuan untuk rotasi perbaikan orientasi citra. Namun jika tidak ditemukan objek, maka citra masukan dibiarkan seperti aslinya tanpa ada perbaikan orientasi

III. METODOLOGI

Pada penelitian ini terdapat beberapa tahapan untuk melakukan proses deteksi objek dan perbaikan orientasi

citra, diantaranya adalah duplikasi dan rotasi citra, praproses yaitu mengubah ukuran citra dan ruang warna ke grayscale, ekstraksi fitur, klasifikasi dan postproses yang meliputi pencarian objek terbaik dan perbaikan

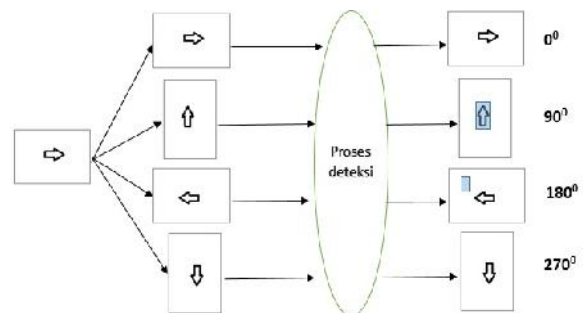


Gambar 1. Diagram alir sistem

orientasi. Tahapan tersebut akan dijelaskan pada sub bab berikut:

3.1. Duplikasi Citra

Salah satu tahapan awal sistem adalah duplikasi citra masukan menjadi 4 macam. Duplikasi citra ini berguna untuk memeriksa segala kemungkinan posisi orientasi yang benar. 4 macam citra hasil duplikasi tersebut akan dirotasi kedalam 4 putaran derajat yang berbeda dengan selisih 90⁰ yaitu 0⁰, 90⁰, 180⁰ dan 270⁰. Arah putaran yang dilakukan adalah berlawanan arah dengan putaran jarum jam. Ilustrasi duplikasi dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi duplikasi citra, kotak kecil berwarna biru adalah posisi objek yang terdeteksi

Semua duplikasi citra yang sudah diputar akan diproses pada deteksi wajah. Setiap objek yang ditemukan akan dicatat posisi rotasi citranya, kemudian objek-objek tersebut akan dirangking dan dicari objek terbaik dengan nilai fitur paling besar. Kemudian merotasi citra sesuai dengan rotasi citra hasil duplikasi dimana objek terbaik tersebut berada.

3.2. Citra Grayscale

Salah satu praproses pada deteksi wajah adalah mengubah ruang warna citra asli menjadi *grayscale*. hal ini dilakukan karena objek wajah pada citra *grayscale* pada umumnya memiliki pola yang konsisten seperti warna mata lebih gelap dari warna pipi atau dahi [4]. Untuk mengubah citra warna (RGB) ke *grayscale* dihitung dengan Rumus (1).

$$grayscale = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

3.3. Fitur Haar-Like

Haar-like merupakan fitur sederhana berbentuk kotak yang digunakan untuk membedakan tingkat intensitas warna (0-255) rata-rata antara region satu dengan lainnya. Dengan fitur ini berbagai macam objek yang memiliki kesamaan bentuk dan warna bisa diambil polanya. Fitur ini menjadi populer ketika digunakan untuk mendeteksi objek wajah [4].

Fitur dasar haar-like memiliki tiga jenis pola untuk mendapatkan nilai-nilainya. Fitur ini dapat berisi dua, tiga atau empat kotak berwarna hitam dan putih yang saling bertetangga (Gambar 3) dan dilapiskan pada citra asli. Tiap fitur digunakan untuk menghitung selisih intensitas piksel antar region hitam dan putih pada citra asli yang ada dibawahnya. Untuk mempercepat perhitungan dapat digunakan teknik *integral image*. Fitur haar-like dapat memiliki berbagai ukuran mulai dari yang terkecil hingga terbesar tapi tidak melebihi ukuran citra, sehingga jumlah fitur ini bisa melebihi jumlah piksel dari citra yang akan dideteksi. Ilustrasi fitur Haar-like pada objek wajah ditunjukkan Gambar 7.



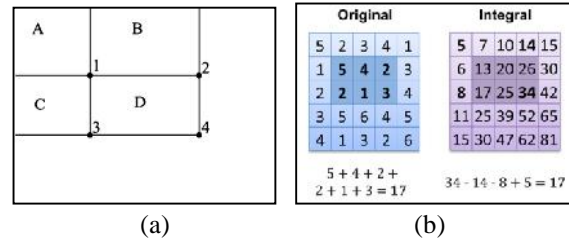
Gambar 3. Fitur haar-like

3.4. Integral Image

Untuk mempercepat perhitungan fitur haar-like dapat menggunakan teknik *integral image* [4]. *integral image* ii pada titik (x, y) berisi nilai penjumlahan semua piksel kiri atas dari titik (x, y) citra asli (*original image*) i , dapat dihitung dengan Rumus (2).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2)$$

Setelah *integral image* terbentuk maka jumlah intensitas piksel region kotak pada citra asli dapat dengan mudah dihitung dengan melihat empat titik referensi pada *integral image*. Misal ada empat region yang saling bertetangga A, B, C, D dengan masing-masing piksel ujung kanan bawah region tersebut adalah 1, 2, 3, 4 (Gambar 4), maka untuk mencari akumulasi intensitas piksel dari region D diperoleh

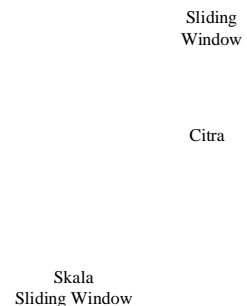


Gambar 4. *Integral image*. (a) Empat titik referensi pada *integral image*; (b) perhitungan *integral image*

dengan menghitung piksel kanan bawah pada *integral image* yaitu : $4 - 2 - 3 + 1$.

3.5. Sliding Window

Sliding window adalah sebuah kotak persegi yang bergerak menelusuri citra mulai dari kiri atas hingga kanan bawah dengan berbagai ukuran (mulai dari ukuran terbesar hingga terkecil) yang berfungsi untuk mengevaluasi apakah daerah (sub region citra) tersebut merupakan objek atau bukan. Perubahan ukuran dan seberapa jauh langkah pergerakan dari kotak satu ke kotak berikutnya merupakan sebuah parameter yang harus ditentukan. Semakin kecil perubahan ukuran dan



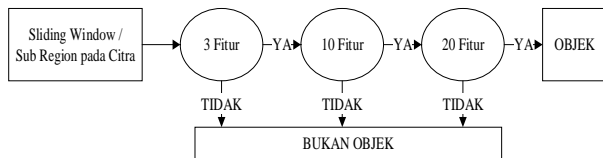
Gambar 5. Ilustrasi *sliding window*

pergerakan *sliding window* maka semakin baik dalam mengevaluasi tiap daerah pada citra, namun semakin besar pula komputasi yang dibutuhkan. Ilustrasi *sliding window* ditunjukkan pada Gambar 5

3.6. Klasifikasi Cascade Adaboost

Klasifikasi cascade adalah sebuah model klasifikasi yang memiliki beberapa tingkatan klasifikasi. Tiap tingkat klasifikasi dibentuk menggunakan algoritma adaboost [4]. Pada prinsipnya baik atau tidaknya kerangka model klasifikasi ini tergantung dari tiga hal,

yaitu jumlah tingkat klasifikasi, jumlah fitur pada tiap tingkat klasifikasi dan nilai batas ambang pada tiap tingkat klasifikasi, namun semua itu diperoleh berdasarkan parameter yang dimasukkan yaitu jumlah fitur, detection rate dan *false alarm rate*. Untuk menentukan parameter yang optimal merupakan suatu masalah tersendiri [4]. Isi dari model klasifikasi cascade adaboost adalah beberapa fitur haar-like dan nilai *threshold*. Sebuah sampel dikatakan positif objek jika lolos (nilai fitur lebih besar dari nilai *threshold*) pada semua tingkat klasifikasi, jika tidak maka sampel tersebut negatif (bukan objek). Metode ini bisa berjalan efektif dalam mereduksi komputasi ketika digunakan untuk deteksi objek pada citra. Untuk mendeteksi objek, dibentuk suatu *sliding window* yang memiliki rasio (perbandingan lebar dan tinggi) sesuai rasio objek yang diinginkan dengan berbagai ukuran mulai terkecil hingga terbesar dan bergerak menelusuri citra dari kiri atas ke kanan bawah. Bagian citra (*sliding window*) yang bukan objek bisa dengan mudah ditolak atau diabaikan pada tingkat awal klasifikasi (Gambar 6).



Gambar 6. Klasifikasi cascade untuk deteksi objek

IV. DATASET

Data yang digunakan untuk proses pelatihan dan pengujian diperoleh dari berbagai sumber. Saat proses pelatihan membutuhkan data positif (citra berisi objek) dan negatif (tidak berisi objek). Data latih wajah berasal dari *CMU Face Detection Project* (http://vasc.ri.cmu.edu/idb/html/face/frontal_images/) pada data latih ini berupa citra foto grayscale dan file teks yang berisi posisi pada bagian-bagian wajah. Dari file teks tersebut bagian wajah dipotong (*crop*) dijadikan sampel positif dan sisanya (*background*) dijadikan sampel negatif. Pada saat pengujian, data uji yang digunakan diperoleh dari <http://tamaraberg.com/faceDataset/>, yang berisi berbagai macam foto dengan objek wajah manusia yang memiliki gaya dan ekspresi bervariasi [5]. Beberapa foto terbaru untuk data uji juga diambil dari *flicker.com* dan *google images*.

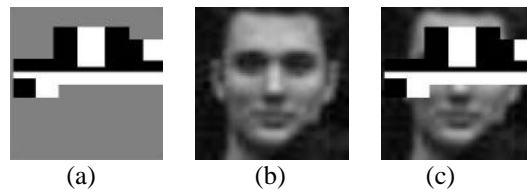
V. HASIL PELATIHAN

Proses pelatihan data menggunakan tools perangkat lunak OpenCV 2.1 dengan *library opencv_haartraining* pada sistem operasi linux debian 5.0. komposisi model klasifikasi cascade adaboost untuk deteksi wajah ditunjukkan pada Tabel 1.

Proses pelatihan deteksi wajah menggunakan 491 data positif dan 1000 data negatif dengan parameter *detection rate* = 0.99 dan *false alarm rate* = 0,5 serta ukuran objek wajah 24x24 piksel, menghasilkan 12

tingkat model klasifikasi cascade, 199 fitur dan nilai *threshold* pada tiap tingkat klasifikasi cascade.

Fitur-fitur teratas pada model klasifikasi wajah jika digambarkan dalam bentuk fitur haar-like ditunjukkan pada Gambar 7. Fitur tersebut akan menghitung perbedaan kecerahan warna antara daerah mata dengan daerah sekitarnya.



Gambar 7. Penggunaan fitur haar-like. (a) 4 fitur pertama dari model klasifikasi *cascade*; (b) citra asli; (c) fitur haar-like ketika dilapiskan pada citra asli

Tabel 1. Model Klasifikasi Cascade Adaboost Deteksi Wajah

Tingkat cascade	Jumlah fitur haar-like	Nilai threshold
1	4	-0,381
2	8	-1,369
3	9	-1,411
4	14	-0,739
5	9	-1,612
6	18	-0,977
7	19	-1,013
8	20	-1,146
9	20	-0,977
10	23	-0,946
11	30	-1,067
12	25	-0,870

VI. HASIL PENGUJIAN

Pengujian dilakukan menggunakan perangkat lunak MATLAB R2014a dengan sistem operasi Windows 8 64 bit dijalankan pada perangkat keras Intel Core i7 2.2 GHz dan RAM 8 GB. Pengujian ini menggunakan 200 citra data uji yang dibagi menjadi 4 kelompok citra dengan masing-masing kelompok berisi 50 citra. Tiap kelompok memiliki rotasi citra yang berbeda yaitu kelompok dengan rotasi 0°, 90°, 180° dan 270° (arah putaran rotasi adalah berlawanan jarum jam). Semua citra data uji yang digunakan memiliki objek wajah. Selama pemrosesan ukuran rasio citra diubah untuk keperluan efisiensi komputasi hingga panjang atau lebar citra menjadi 384 piksel (mana yang lebih besar).

Pencarian objek pada klasifikasi cascade diterapkan seperti pada umumnya dilakukan oleh [4][6] yaitu dengan menelusuri setiap bagian citra menggunakan *sliding window* dengan berbagai ukuran mulai dari yang

terbesar sesuai ukuran citra masukan hingga terkecil sesuai ukuran citra data latih (24x24 piksel). Setiap bagian citra yang masuk pada *sliding window* akan dihitung nilai fiturnya, jika nilai fiturnya lebih besar dari nilai *threshold* awal klasifikasi, maka bagian citra pada *sliding window* akan diproses lebih lanjut ke tingkat klasifikasi berikutnya dengan jumlah fitur yang lebih kompleks. Jika semua nilai fitur pada tiap tingkat cascade adaboost lebih besar dari semua nilai *threshold* klasifikasi cascade, maka *sliding window* dianggap objek, seperti pada Gambar 8.

Untuk melakukan perbaikan orientasi, dibutuhkan satu objek terbaik yang berasal dari 4 citra hasil duplikasi dan rotasi, kemudian objek terbaik tersebut digunakan sebagai acuan perbaikan. Dari semua objek yang ditemukan, dicari nilai fitur terbesar F dari *sliding window x yang terpenuhi sebagai objek dengan nilai fitur f_i pada tingkat ke i klasifikasi cascade adaboost, dapat dinyatakan dengan Rumus (3).*

pada tingkat ke i klasifikasi cascade :
dinyatakan dengan rumus (3).

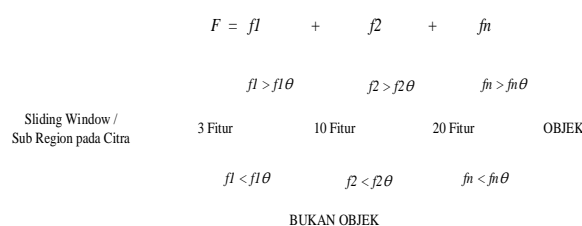
$$F = \arg \max \left\{ \sum_{i=1}^n f_i(x) \right\} \quad (3)$$

Setelah mendapatkan objek F , maka citra asli akan dirotasi sesuai dengan citra dimana objek F tersebut berada. Contoh penerapan pencarian objek terbaik dapat dilihat pada Gambar 9. Dari citra masukan, objek wajah terdeteksi pada putaran 0° dan 180° dengan objek terbaik terletak pada putaran 180° sehingga perbaikan orientasi citra dilakukan dengan memutar citra asli sebesar 180° .

Hasil uji coba ditunjukkan oleh Tabel 2. Pada uji coba ini, dari 200 citra uji terdapat 7 citra yang salah rotasi, 32 citra yang tidak ditemukan objek wajah sehingga dibiarkan tanpa ada perubahan rotasi dan 161 citra yang dirotasi dengan benar atau memiliki akurasi $161/200 = 80,5\%$.

Untuk memproses 200 citra membutuhkan waktu sekitar 167,81 detik, sehingga rata-rata untuk memproses satu citra adalah $161,81/200 = 0,81$ detik. Beberapa hasil dari metode ini ditunjukkan oleh Gambar 10.

Kesalahan rotasi terjadi bisa disebabkan karena kesalahan proses deteksi dalam mendeteksi objek wajah (sebenarnya bukan wajah tapi terdeteksi wajah) hal ini biasa disebut *false positive*. *False positive* ini



Gambar 8. Ilustrasi pencarian nilai total fitur F

sering terjadi ketika ada bagian dari citra yang bukan objek wajah namun memiliki pola yang hampir mirip dengan objek wajah, seperti terdapat dua titik yang mirip mata, dimana titik tersebut lebih gelap dibanding daerah sekitar (Gambar 10.d).



Gambar 9. Pencarian objek terbaik. Kotak merah adalah objek wajah yang terdeteksi dan Nilai F (warna putih diatas kotak merah). (a) citra masukan; (b) satu objek terdeteksi pada putaran 0° dengan nilai $F = -3,2305$; (c) empat objek terdeteksi pada putaran 180° dengan nilai F terbesar 32,4773; (d) citra hasil

Tabel 2. Hasil Uji Coba dari 200 Citra

Hasil	Jumlah
Benar	161
Salah	7
Tidak berubah (tidak ditemukan wajah)	32
Total	200

Citra yang tidak dilakukan perbaikan rotasi disebabkan karena metode deteksi wajah yang diterapkan gagal menemukan objek, meskipun semua data uji yang digunakan memiliki objek wajah. Kegagalan proses deteksi dalam menemukan wajah bisa disebabkan berbagai macam hal, diantaranya karena objek wajah

yang terhalang oleh benda lain seperti topi atau kacamata (Gambar 10.e), selain itu bisa juga disebabkan karena objek wajah yang menoleh atau miring (Gambar 10.f).



Gambar 10. Contoh hasil uji coba, 6 pasang citra masukan dan hasil dengan objek wajah yang ditemukan ditandai dengan kotak warna kuning. (a)(b)(c) citra yang terotasi dengan benar; (d) citra yang terotasi dengan salah; (e)(f) citra tidak dilakukan rotasi karena tidak ditemukan objek wajah

VII. PENUTUP

7.1. Kesimpulan

Pada penelitian ini diusulkan sebuah metode untuk melakukan perbaikan orientasi secara otomatis pada foto atau citra yang memiliki objek wajah manusia didalamnya. Objek wajah diperoleh dengan cara menghitung fitur haar-like pada citra *grayscale* dengan klasifikasi cascade adaboost. Satu objek wajah dengan nilai fitur terbesar akan dijadikan acuan untuk perbaikan orientasi. Dari 200 citra data uji, metode yang diusulkan mampu melakukan perbaikan orientasi dengan benar hingga 80,5% dengan kecepatan pemrosesan 0,81 detik per citra.

7.2. Saran

Saran untuk peneliti berikutnya adalah mengkombinasikan metode deteksi wajah dengan deteksi kepala atau tubuh bagian atas (*upper body*)

untuk meningkatkan akurasi. Karena dari hasil uji coba banyak objek wajah yang tidak terdeteksi disebabkan karena perubahan ekspresi, posisi wajah yang miring atau terhalang dengan suatu benda.

DAFTAR PUSTAKA

- [1] M. Borawski dan D. Frejlichowski. 2012. *An Algorithm for the Automatic Estimation of Image Orientation*. The Scientific World Journal Machine Learning and Data Mining in Pattern Recognition. Springer Berlin Heidelberg, vol. 7376, hal. 336-344.
- [2] Y. Wang dan H. Zhang. 2001. *Content-based Image Orientation Detection with Support Vector Machines*. Content-Based Access of Image and Video Libraries, IEEE Workshop, hal. 17-23.
- [3] M. Datar dan X. Qi. 2006. *Automatic Image Orientation Detection using The Supervised Self-Organizing Map*. Imaging Technologies. Signal and Image Processing. Proc. of the 8th IASTED International Conference.
- [4] P. Viola dan M. Jones. 2004. *Robust Real-Time Face Detection*. International journal of computer vision, vol. 57(2), hal. 137-154.
- [5] T. Berg, A. Berg, J. Edwards dan D. Forsyth. 2004. *Who's in the Picture*. Neural Information Processing Systems (NIPS).
- [6] Z. Wang, S. Yoon, S.J. Xie, Y. Lu dan D.S. Park. 2014. *A High Accuracy Pedestrian Detection System Combining a Cascade AdaBoost Detector and Random Vector Functional-Link Net*. The Scientific World Journal.